University of Pisa

Department of Computer Science
Ph.D. in Computer Science

Personal Data Analytics

Capturing Human Behavior to Improve Self-Awareness and Personal Services through Individual and Collective Knowledge

Ph.D. Thesis

Ph.D. Candidate
Riccardo Guidotti

Supervisors
Prof. Dino Pedreschi
Dott.ssa Fosca Giannotti

21 April 2017

Università degli Studi di Pisa

Dipartimento di Informatica
Dottorato di Ricerca in Informatica

Ph.D. Thesis: XXIX Cycle

Personal Data Analytics

Capturing Human Behavior to Improve Self-Awareness and Personal Services through Individual and Collective Knowledge

Riccardo Guidotti

Supervisor
Prof. Dino Pedreschi
Dott.ssa Fosca Giannotti

Referee
Prof. Srinivasan Parthasarathy
Dr. Ravi Kumar

Chair
Prof. Pierpaolo Degano

21 April 2017

a Camilla

Abstract

In the era of Big Data, every single user of our hyper-connected world leaves behind a myriad of *digital breadcrumbs* while performing her daily activities. It is sufficient to think of a simple smartphone that enables each one of us to browse the Web, listen to music on online musical services, post messages on social networks, perform online shopping sessions, acquire images and videos and record our geographical locations. This enormous amount of personal data could be exploited to improve the lifestyle of each individual by extracting, analyzing and exploiting user's behavioral patterns like the items frequently purchased, the routinary movements, the favorite sequence of songs listened, etc. However, even though some user-centric models for data management named Personal Data Store are emerging, currently there is still a significant lack in terms of algorithms and models specifically designed to extract and capture knowledge from personal data.

This thesis proposes an extension to the idea of Personal Data Store through *Personal Data Analytics*. In practice, we describe parameter-free algorithms that do not need to be tuned by experts and are able to automatically extract the patterns from the user's data. We define personal data models to characterize the user profile which are able to capture and collect the users' behavioral patterns. In addition, we propose individual and collective services exploiting the knowledge extracted with Personal Data Analytics algorithm and models. The services are provided for the users which are organized in a Personal Data Ecosystem in form of a peer distributed network, and are available to share part of their own patterns as a return of the service providing. We show how the sharing with the collectivity enables or improves, the services analyzed. The sharing enhances the level of the service for individuals, for example by providing to the user an invaluable opportunity for having a better perception of her self-awareness. Moreover, at the same time, knowledge sharing can lead to forms of collective gain, like the reduction of the number of circulating cars. To prove the feasibility of Personal Data Analytics in terms of algorithms, models and services proposed we report an extensive experimentation on real world data.

Acknowledgments

Rather than a job, or a course of study, a Ph.D. is a three year long, complex and not linear journey you make with your friends, colleagues, random encounters, loves, failures and successes. Only a Ph.D. can really understand what is a Ph.D. First of all, I want to thank Camilla, my only love, because even though she is not a Ph.D., she understood me better than everyone else. I want to thank her because she patiently accompanied me along this journey, giving me advice, listening to me, celebrating my successes and encouraging me after a defeat or in front of what looked like insuperable problems.

I am very grateful to my advisors Dino Pedreschi and Fosca Giannotti, who first convinced me to do a Ph.D., then believed in me and supported me. They guided me and shared with me exciting moments of research and thanks to their enthusiasm I learned to love the world of research and my work. Many thanks to the senior researchers Anna, Mirco, Roberto and Salvo that helped me to better understand my results and findings. A special thank goes to Anna who helped me to correct this thesis.

Another big thank goes to my KDDLab colleagues (past and present) for sharing ideas and reciprocal help: Giulio, Michele B., Michele C., Luca, Diego, Francesca, Letizia, Valerio, Daniele, Michela, Paolo, Lorenzo, Barbara, Vittorio, Chiara, Ioanna, Farzad, Laura, Viola, and Roberto. I want to thank also all my Ph.D. colleagues of the XXIX cycle. A particular thank goes to Jacopo, my roommate in the Department of Computer Science, he endured me, my Skype calls and my frequent questions about administrative things.

Many thanks to my friends and flatmates here in Pisa: Riccardo, Luca, Dario, Leonardo, Nicola and Marco. They made me laugh every day about my Ph.D., making fun of me and my "work" by saying that I was not really working but I was passing my days searching for gold like a mine searcher and stealing money from the state. Thanks to my best friends from Pitigliano for loving me regardless the geographical distance and my occasional appearances: Giovanni, Cristiano, Lucia, Laura, Andrea, Luca and Emilio.

The final, enormous "thank you" goes to my family: my mom, my dad, my brother and my grandparents. They always supported me, and constantly make me a better person giving me the example with their lives. Even if I'm not good at showing you my gratitude day by day, I'll always consider me as the luckiest in the world for having so special family.

Thank you to all those who have been close to me during these years.

Contents

1 Introduction **19**

I Setting the Stage **25**

2 Data Mining and User Profiling **27**
2.1 What is Data Mining? 27
2.2 User Profiling through Data Mining 27
2.3 Clustering: A Data Mining Technique for User Profiling 33

3 User Profiling for Individual and Collective Services **37**
3.1 Predicting Human Behavior 37
3.2 Recommendation Systems 42

4 Personal Data Store: A User-Centric Model **47**
4.1 Towards a User-Centric Model 47
4.2 What is a Personal Data Store? 48

II Personal Data Analytics **53**

5 Personal Data Analytics **55**
5.1 We All Need To Own and Use Our Own Data 55
5.2 Making Sense of Own Personal Big Data 56
5.3 The Personal Data Ecosystem 58
5.4 Potentialities and Socio-Economic Impact 61

6 Proxies of Human Behavior **63**
6.1 Human Mobility Data 63
6.2 Retail Market Data 65
6.3 Social-Network Data 65
6.4 Music Listening Data 66

III Algorithms and Models for Personal Data Analytics 67

7 Autofocus Algorithms for Personal Data Mining 69

7.1 Clustering Algorithm for Personal Location Detection 69

7.2 Transactional Clustering for Personal Data Mining 82

8 Personal Data Models 97

8.1 Personal Behavioral Entropy and Profitability in Retail 97

8.2 A Personal Data Model for Musical Preferences 108

8.3 Towards a Personal Automatic Spatio-Temporal Agenda 115

IV Personal Data Analytics for Individual and Collective Services 121

9 Improving Personal Mobility 123

9.1 Trajectory Prediction through Mobility Profiling 123

9.2 Towards an Experienced Route Planner 138

10 Social and Proactive Carpooling 145

10.1 Boosting Carpooling with Network Analysis 146

10.2 Enjoyable Carpooling from Crowdsourced Data 166

11 Socio-Economical Analysis of Well-Being 181

11.1 Discovering Temporal Shopping Regularities 181

11.2 Using Retail Market Data to Nowcast Well-Being 195

12 Deployment of Personal Data Analytics 203

12.1 LivLab: The Adoption of the Personal Data Store 203

12.2 Integration of Private and Public Means of Transport 205

13 Conclusion 209

Bibliography 211

List of Figures

2.1 The process of Knowledge Discovery in Databases 28
2.2 Mobility profiles: hierarchical view [113] 29
2.3 Mobility profiles: locations covering path [26] 29
2.4 Mobility profiles: extraction steps [285] 30

4.1 Personal Data Store: openPDS architecture [91] 49

5.1 Personal Data Analytics: making sense of own personal data 56
5.2 Personal Data Analytics: the Personal Data Ecosystem 58

6.1 Dataset: Octo and Coop dataset area 64
6.2 Dataset: Twitter dataset area for Rome and San Francisco 66

7.1 Tosca: location detection problem 71
7.2 Tosca: clustering steps 73
7.3 Tosca: experiments evaluation on null model 78
7.4 Tosca: experiments evaluation on mobility-like model 79
7.5 Tosca: experiments evaluation real data 80
7.6 Tosca: best competitors comparison and cut distance distribution 81
7.7 Tosca: location detection example on real data 81
7.8 Txmeans: structure of synthetic datasets 90
7.9 Txmeans: experiments evaluation of noisy synthetic datasets 91
7.10 Txmeans: experiments evaluation on scalability 92
7.11 Txmeans: experiments evaluation on sample size 92
7.12 Txmeans: sample size evaluation on real-world dataset 93
7.13 Txmeans: mass clustering evaluation 94
7.14 Txmeans: case study recommendation performances 96

8.1 Customer unpredictability: dataset distributions 99
8.2 Customer unpredictability: minimum support effect 103
8.3 Customer unpredictability: unpredictability distributions 103
8.4 Customer unpredictability: k-means SSE trend 103
8.5 Customer unpredictability: relationship with expenditure 105
8.6 Customer unpredictability: validation of systematic customers 105
8.7 Customer unpredictability: heatmaps of unpredictability and expenditure 106
8.8 Listening data model: listening event data 109
8.9 Listening data model: personal patterns extraction 109
8.10 Listening data model: data model details 110

8.11 Listening data model: indicators distributions 111
8.12 Listening data model: entropy distributions 111
8.13 Listening data model: indicator and entropy correlations 112
8.14 Listening data model: radar charts representing cluster centroids 113
8.15 Listening data model: frequency analysis for genre and artist 113
8.16 Listening data model: data storage analysis 115
8.17 Personal agenda: example of personal agenda 119

9.1 MyWay: interpolated route distance 125
9.2 MyWay: routine - trajectory match example 128
9.3 MyWay: prediction strategies schema 129
9.4 MyWay: prediction validation example 130
9.5 MyWay: accuracy rate evaluation for individual strategy 131
9.6 MyWay: prediction rate and accuracy evaluation for individual strategy . 131
9.7 MyWay: users predictability vs prediction rate 132
9.8 MyWay: spatial error evaluation for individual strategy 132
9.9 MyWay: accuracy rate evaluation for collective strategy 132
9.10 MyWay: prediction and accuracy rate evaluation for collective strategy . . 132
9.11 MyWay: users' predictability vs prediction rate for collective strategy . . . 133
9.12 MyWay: accuracy rate evaluation for hybrid strategy 133
9.13 MyWay: spatial error evaluation for hybrid strategy 133
9.14 MyWay: accuracy rate evaluation for global competitor 133
9.15 MyWay: prediction rate, collective and global coverage 134
9.16 MyWay: accuracy rate evaluation for hybrid-global competitor 134
9.17 MyWay: individual competitor performances 136
9.18 MyWay: collective competitor performances 136
9.19 MyWay: performance comparison 137
9.20 MyWay: performance evaluation on participation analysis 137
9.21 Route planner: mobility model construction steps 140
9.22 Route planner: deviation distributions for space, time and occurrences . . 141
9.23 Route planner: deviation nodes 142
9.24 Route planner: heatmap of deviation cells 142
9.25 Route planner: deviation cells distributions 143
9.26 Route planner: deviation flows 144

10.1 Proactive carpooling: routine containment example 148
10.2 Proactive carpooling: carpooling networks 149
10.3 Proactive carpooling: carpooling temporal network 152
10.4 Proactive carpooling: distribution of routines per user 156
10.5 Proactive carpooling: network construction parameters 156
10.6 Proactive carpooling: carpoolers classification 157
10.7 Proactive carpooling: routines distributions 157
10.8 Proactive carpooling: network degree and ranking scores distributions . . 158
10.9 Proactive carpooling: carpooling communities 159
10.10 Proactive carpooling: carpooling ranking scores box-plots 160
10.11 Proactive carpooling: autonomous and not autonomous communities . . . 160
10.12 Proactive carpooling: assignment performances evaluation 161

10.13 Proactive carpooling: assignment performances comparisons 162
10.14 Proactive carpooling: single occupancy vehicle distributions 162
10.15 Proactive carpooling: spatial distributions of pick-up and drop-off points . 165
10.16 Social carpooling: web page of the survey 174
10.17 Social carpooling: social measures distributions 176
10.18 Social carpooling: performance evaluation 178
10.19 Social carpooling: performance comparisons 178
10.20 Social carpooling: social performance evaluation 179

11.1 Temporal regularity: data unit 183
11.2 Temporal regularity: purchases data model 183
11.3 Temporal regularity: purchases data sequence 184
11.4 Temporal regularity: dataset distributions 188
11.5 Temporal regularity: individual distributions 190
11.6 Temporal regularity: collective distributions 191
11.7 Temporal regularity: collective temporal patterns 192
11.8 Temporal regularity: SSE and cluster size distribution for temporal patterns192
11.9 Temporal regularity: regular sub-sequences 193
11.10 Temporal regularity: SSE and cluster size distribution for sub-sequences . 193
11.11 Nowcasting well-being: bipartite network example 197
11.12 Nowcasting well-being: customers sophistication distribution 199
11.13 Nowcasting well-being: GDP and customers expenditure comparison . . . 200
11.14 Nowcasting well-being: GDP and customers sophistication comparison . . 201
11.15 Nowcasting well-being: correlation shifts 202

12.1 LivLab: personal home page 204
12.2 LivLab: visualization of personal statistics 204
12.3 LivLab: visualization of individual mobility network 204
12.4 LivLab: purchases and comparison with the other analysis 204
12.5 Multi-modal carpooling: PETRA architecture 206
12.6 Multi-modal carpooling: impact of routines on travel time 207
12.7 Multi-modal carpooling: PETRA journey planner application 207

List of Tables

6.1 Dataset: description of the datasets 63

7.1 Tosca: evaluation measures 75
7.2 Tosca: experiments on null model and mobility-like model 78
7.3 Tosca: experiments evaluation internal measures 80
7.4 Txmeans: dataset symbols and descriptions 90
7.5 Txmeans: experiments evaluation on simple synthetic datasets 91
7.6 Txmeans: experiments evaluation on real-world datasets 93
7.7 Txmeans: case study dataset statistics 95

8.1 Customer unpredictability: products of the systematic customers 103
8.2 Customer unpredictability: clusters statistics 104
8.3 Customer unpredictability: cluster centroid with unpredictability level . . . 104
8.4 Customer unpredictability: effect in predicting the total expenditure 107
8.5 Listening data model: cluster centroids 112
8.6 Listening data model: most listened artist and genre 114

9.1 MyWay: symbols and descriptions 131
9.2 Route planner: deviation analysis statistics 141

10.1 Proactive carpooling: datasets and performance summary 164
10.2 Proactive carpooling: estimates of potential saving 164
10.3 Social carpooling: user recommendations statistics 177
10.4 Social carpooling: estimated savings 179
10.5 Social carpooling: recommendation comparisons 180

11.1 Temporal regularity: symbols and descriptions 185
11.2 Nowcasting well-being: most and least sophisticated products 199
11.3 Nowcasting well-being: symbols and descriptions 200
11.4 Nowcasting well-being: measures correlations 201

List of Algorithms

1 Tosca: *main* function 72
2 Tosca: *select-cut* function 72
3 Txmeans: *main* function 83
4 Txmeans: *getRepr* function 84
5 Txmeans: *bisectBaskets* function 86
6 Txmeans: *txmeans-sampling* function 88

7 Customer unpredictability: *BRE* function 101
8 Customer unpredictability: *getReprBasketsCount* function 101

9 MyWay: *IRD* function 126

10 Proactive carpooling: *calculateGeedyAssignment* function 150
11 Proactive carpooling: *NeverDriveAlone* function 153
12 Proactive carpooling: *NeverDriveAloneCommunities* function 154
13 Social carpooling: *GRAAL* function 172

14 Temporal regularity: *extractIndividualProfile* function 184
15 Temporal regularity: *extractCollectiveFootprints* function 186
16 Temporal regularity: *provideCollectivePerspective* function 186

Chapter 1

Introduction

> “ ... and you’re a detail into the general framework,
> that they wish but can’t ignore ... "
>
> — L. Ligabue

Every day we create 2.5 quintillion bytes of data. So much that 90% of the data in the world today has been created in the last two years. These data come from everywhere: climatic data from sensors, posts to social media sites, pictures and videos, purchase transactional records, and cell phone GPS signals to name a few. These data are *big data* [98]. Most of these data are generated from the mobile phones that we use in our everyday life. In 2000 mobile phone users accounted for 12% of the world’s population. By the end of 2014, this figure reached 97%, i.e., 6.8 billion people. The number of mobile phones is 128% of the inhabitants in developed countries and 90% in developing countries. These numbers provide a better idea of the impact that mobile phones are having in our life.

In addition to that, there has been an explosive increase in the number of ways our smartphones can acquire and produce data through their built-in sensors, capable of recording locations (GPS data), acceleration, acquiring images and videos, interacting with other devices, connecting to the Internet and, obviously making phone calls (GSM data). The connection to the Internet enables the individuals to use an enormous set of different applications and services ranging from online social networks and shopping websites to search engines and online musical services. Hence, in turn, the usage of these applications produces others tons of personal data like the web pages visited in a browsing session (e.g. Google), the messages posted on a social network (e.g. Facebook and Twitter), or the songs listened on an online musical service (e.g. Spotify and LastFM). Furthermore, the widespread use of fidelity cards in retail market chains and in other services empowers the personal tracking of data like the purchased items in a shopping basket or the clinical events in a patient’s history. When it comes to producing data, we are really prolific: it has been estimated that, at individual level, each person generates an avalanche of information, i.e., more than 5 gigabytes of *personal data* per year, without considering images and videos. Thus, every one of us is an individual producer of big data.

The *digital breadcrumbs* that we leave behind us are probably the most unexpected and disruptive effect of the emergence of always-connected mankind. As consequence of this large data production, the world's technological per-capita capacity to store information has roughly doubled every 40 months since the 1980s [145]. This storage availability is a prerogative of service providers, i.e., institutions and companies. They collect our personal data for creating a mosaic of human behaviors used to extract valuable knowledge for marketing purposes: our personal data is the new gold. On the other hand, individuals do not have the tools and capabilities to extract useful knowledge from their personal data.

The knowledge hidden in personal data consists in the patterns people regularly follow in their routinary life. Personal patterns can be worth for many services and applications because human behavior is predictable in principle: people are systematic in their everyday choices and do not change their behavior randomly. In the book "Bursts" [23], Barabasi presents the theory that human behavior is bursty, i.e., humans have long inactivity intervals separated by moments of rapid activity. Recently, this theory has been repeatedly tested [22, 292]. Humans are predictable at the *individual* level. Bursty patterns of activities have been observed and can be predicted, for instance in writing emails. Also individual mobility is predictable: most people commute every working day between the same two locations, and can be predicted to do so with very high accuracy [297]. Moreover, humans are predictable also at *collective* level: groups of humans flock together in predictable patterns. For instance, people are more mobile early in the morning and late in the afternoon.

Therefore, up to now, the highly valuable personal patterns able to predict human behavior can only be extracted by big companies, which employ this information mainly to improve marketing strategies. This *organization-centric model* does not empower to take full advantage of the possibility of knowledge extraction offered by personal data [158], mainly because each company has only a limited view on individuals that is restricted to the type of data for which the company provides services. Moreover, users have a very limited capability to control and exploit their personal data. To overcome such problems, in line with the World Economic Forum [161, 239, 250], it has been proposed a change of perspective towards a *user-centric model* named *Personal Data Store* for personal data management [143]. In this model the user acquires a central role in the control of the lifecycle of her own personal data [119]. However, despite some novel user-centric model are being defined, in the current state-of-the-art there is yet a deep lack of algorithms and models specifically designed to extract knowledge from personal data, that is, automatic methods which do not need to be managed by an expert, or which require data which are not own by the user thyself. Data mining applied to personal data, i.e., *Personal Data Mining*, is the key for extracting personal patterns and, at the same time, an invaluable opportunity for individuals to improve their self-awareness and their lifestyle.

Starting from these observations, in this thesis we define how to extend the idea of Personal Data Store by articulating a *Personal Data Analytics* approach that seeks to analyze the digital breadcrumbs an individual leaves behind, and we demonstrate that the defined approach and the resulting analyses can lead to increased individual and collective benefits. Indeed, a key element of Personal Data Analytics is the analytical reinforcement resulting from the synergy of the widespread knowledge in the *Personal Data Ecosystem*.

In particular, in order to realize the *Personal Data Analytics* approach, we design parameter-free algorithms for Personal Data Mining able to automatically extract from the user's Personal Data the user's behavioral patterns. In addition, we define the user profile through Personal Data Models which are able to capture and collect the behavioral patterns containing the user hidden knowledge. Some examples can be the set of items frequently purchased together, the set of locations frequently visited, or a measure that expresses the level of repetitiveness with which a user listens to a musical genre, etc. Moreover, we describe the *Personal Data Ecosystem* as a peer distributed network of Personal Data Stores where the users can decide to share the knowledge extracted from their data. This enables the development of *individual* and *collective* services for the users. Thanks to the knowledge sharing with the collectivity, each user can gather an additional level of knowledge improving in this way the available services. Personal models can be exploited to compare the user's behavior with that of the others and improve the user's self-awareness, or to provide personalized services like shopping recommendations or trajectory prediction. Note how in such a distributed ecosystem there is not need of a central node providing a service, the user's collaboration and participation realize it [143, 158].

As concrete example of Personal Data Analytics we can refer to the vision proposed in [137] with respect to mobility data. In [137] it is proposed a Personal Data Store where all our movements are stored and Personal Data Analytics provides the algorithms and models to automatically extract the patterns describing our behavior like the locations we frequently visit and the time we spend in there, the routes we daily follow for our commuting, some indicators expressing our level of predictability and how far we move on average from our home. All these patterns are collected by a Personal Mobility Data Model defining the user profile. Then, the Personal Data Ecosystem could enable a proactive carpooling system which automatically links the users that can daily share the car, and at the same time considers that the suggestions provided are also the best ones for the collectivity, since they help in minimizing the number of circulating cars.

In this thesis, we demonstrate empirically that Personal Data Analytics is feasible in a real scenario. We realize several case studies by employing Personal Data Analytics on various real datasets. In particular, we analyze four different types of data. *Mobility data*, that is car movements represented as sequences of GPS points. *Transactional data*, i.e., market retail purchases extracted from fidelity cards containing the items composing the baskets, the shop where the transaction happened and the time and the day of the shopping session. *Music listening data*, that is web listening session relative to the track, artist, genre and time of the listening. *Social network data* from Twitter containing opinions, time and positioning of the tweet besides the social networks of the users analyzed. A characteristic of these datasets is that, in each dataset, independently of one another, the events (GPS signal, purchase, listening, tweet) in the dataset can be referenced to a user. This allows us to recognize the personal data of each user and to extract the user profile.

Thanks to these data, human activities at personal level can be minutely observed and, therefore, measured, quantified and, ultimately, predicted. It is not surprising how many aspects of our daily behavior, like our *whereabouts* and *purchases*, become predictable, given the regularity of our routines. By adopting Personal Data Models we estimate the levels of predictability in listening to a musical genre, in moving to occasional locations and in purchasing infrequent items in not habitual days. Moreover, the Personal Data Models summarize the shopping habits of a user in terms of typical basket composition and time and day of the shopping session, and condense the mobility habits in the set of route frequently followed and habitual locations visited. By exploiting these models at collective level we realized carpooling services minimizing the number of cars and considering at the same time also the social opinions of the users. The mobility models are used also for developing a trajectory prediction system, and the prototype of a route planner making use of the collective awareness. Furthermore, we have been able to provide a reliable nowcasting of the level of well-being of the user in the Personal Data Ecosystem analyzed.

With Personal Data Analytics we are just at the beginning of a data revolution that by putting the user at the center of the system will profoundly impact all aspects of the society: government, business, science and entertainment [12, 119, 181]. The personal big data we daily produce are the lens of a social *microscope* able to understand how we behave individually and collectively and also to predict these behaviors. Thanks to Personal Data Analytics we can activate this microscope and make it usable to everyone. It will help us to predict the consequences of our decisions both at collective and individual levels. Therefore, we will be in a position to make better choices, be more aware, understand and, perhaps, manage the complexity of the pluralistic and interconnected society we live in. It will improve our well-being. An entirely "new deal" for personal data is necessary to fully unleash the power of Personal Data Analytics in a safe ecosystem.

At the same time, we cannot give up the right of managing our personal information and communications freely and safely, sharing what we want with whom we choose and like. The Personal Data Ecosystem can be considered "safe" when the users are enabled with a secure management for their own Personal Data Store. Ethical issues, as well as privacy and trust must be properly handled. In our proposal, the autofocus algorithms and methods of Personal Data Analytics enable individuals to derive from their own data personal models summarizing user's behavior while revealing fewer details with respect to the entire raw data. Thus, the opportunity of participating in a collective service by sharing only the personal models helps in reaching an acceptable trade-off between benefits and risk in information sharing. In addition, the interplay between privacy and information exchange becomes even more satisfactory if the models shared are built using *privacy-by-design* methodologies [58]. However, ethics, trust, and privacy issues are not the central point of this thesis, and these analyses are left as future work.

Summing up, in this thesis we define a *Personal Data Analytics* approach through the design of Personal Data Mining methods and models to enrich the Personal Data Store with personal patterns, and through the development of services for the user which capitalize both on the individual knowledge, and also on the collective knowledge emerging from the cooperation of the users participating to the Personal Data Ecosystem.

The thesis is organized in four parts. In the first part, *Setting the Stage*, we introduce the preliminary notions needed to become familiar with the fields of research covered by this thesis. In Chapter 2 we recall the basic concepts related to data mining, user profiling, and clustering, that is one of the techniques most largely used in data mining to extract user profiles. Chapter 3 reports the current state-of-the-art with respect to personal models and individual and collective services with a focus on predictions and recommendations services in the fields of mobility and shopping data. After that, Chapter 4 illustrates the user-centric model with the current state of Personal Data Stores, which are the available implementations and what they offer to the customer.

In the second part, *Personal Data Analytics*, we present the main contribution of this thesis: the Personal Data Analytics approach and the Personal Data Ecosystem as a distributed network of users. Chapter 5 designs our vision of the Personal Data Store highlighting the differences with the state-of-the-art and by enhancing how Personal Data Mining is fundamental to extract Personal Data Models. Moreover, we show the Personal Data Ecosystem, i.e., the distributed network of users interconnected through their Personal Data Store and which are the potentialities of this approach and the socio-economic impact. Furthermore, in Chapter 6 we describe the real world datasets used in the analyses and case studies reported in the subsequent parts of this thesis.

The third part, *Algorithms and Models for Personal Data Analytics*, presents algorithms and models for Personal Data Mining. According to Personal Data Analytics, the clustering algorithms presented in Chapter 7 are auto-adaptive methods that do not require parameter tuning and that are able to automatically extract the behavioral patterns of each user on different type of data. In Chapter 8 we formalize some Personal Data Models to represent, measure and evaluate the systematic behavior of the users with respect to transactional data, mobility data, and listening data.

In the fourth part, *Personal Data Analytics for Individual and Collective Services*, we describe individual and collective services developed for the Personal Data Store and considering the Personal Data Ecosystem which exploits the models and methods presented in the previous parts. In Chapter 9 we develop two services for improving personal mobility: a personalized trajectory prediction system, and a route planner which exploits the wisdom of the crowd in suggesting the best route to be followed. In Chapter 10 we focus on two different carpooling services with the purpose of promoting carpooling as an everyday means of transport. The first carpooling service exploits habitual paths and complex network analysis to reduce the number of travelers driving alone. The second carpooling service we propose considers both mobility and social data in order to reduce the reticence to share the car with strangers together with the number of circulating cars. Chapter 11 shows how a collective analysis of shopping transactions either considering the temporal dimension or the basket dimension, can lead to a new level of knowledge useful to estimate the well-being level that can not be reached when only the data of a single individual is considered. In Chapter 12 we describe the deployment of the Personal Data Analytics approach in two scenarios for real applications.

Finally, Chapter 13 concludes this thesis by summarizing the main findings, and by presenting possible future research directions for Personal Data Analytics.

The last three parts of this thesis are based on peer-reviewed papers published in international conferences and journals. The concepts in Part II relative to Personal Data Analytics are partly gathered from [137]. In Part III, the algorithm described in Section 7.1 is inherited from [136], the measures of shopping profitability of Section 8.1 comes from [129], and the personal model for musical listening of Section 8.2 is extracted from [134]. Likewise, in Part IV the trajectory prediction system presented in Section 9.1 is gathered from [284], the route planner discussed in Section 9.2 is inherited from [128]. Moreover, Chapter 10 is gathered from the papers published about carpooling [127, 133, 135]. Finally, the nowcasting of well-being in Section 11.2 is extracted from [130] and the integration of private and public mobility in Chapter 12 comes from [31, 42]. The papers from which are extracted Sections 7.2, 8.3, part of Sections 10.2, 11.1, 12.1, and Chapter 5 are currently under review for international conferences or journals.

The algorithms, models, services and analytical results reported as contributions of this thesis are the results of the work done in the following European projects.

The *ICON* project[1] aims at developing a new approach in which gathered data is systematically analyzed to dynamically revise and adapt constraints and optimization criteria through a novel Inductive Constraint Programming paradigm that bridges the gap between the areas of data mining on one hand, and constraint programming on the other hand. The proactive carpooling system proposed in Section 10.1 extracted from [133], and papers [132, 173] (not discussed in this thesis) were realized in the ICON project.

The *PETRA* project[2] aims at developing a service platform that connects the providers and controllers of transport in cities with the travelers in a way that information flows are optimized, while respecting and supporting the individual freedom safety and security of the traveler. Cities will get an integrated platform to enable the provision of citizen-centric, demand-adaptive city-wide transportation services. Travelers will get mobile applications that facilitate them with personalized travel priorities and choices for route and modality. In PETRA were realized the following works [128, 284] relative to Chapter 9, [127, 133, 135] relative to Chapter 10, and [31, 42] relative to Chapter 12. Also [131] (not discussed in this thesis) was realized in PETRA.

The personal measures of shopping predictability [129] presented in Section 8.1 are a contribution both to HII and to CIMPLEX. The *HII*[3] project is an ambitious approach looking for a European solution to store digital data and contents, so that consumers and businesses in Europe do not have to worry on where and by whom their valuable digital age assets are handled, i.e., a prototype of Personal Data Store. The *CIMPLEX*[4] project proposes a visionary research to develop modeling, computational, and ICT tools needed to predict and influence disease spread and other contagion phenomena in complex social systems. Also [132, 252, 253] (not discussed in this thesis) were realized in CIMPLEX.

Finally, the *SoBigData*[5] project aims at creating the Social Mining & Big Data Ecosystem, i.e., a research infrastructure providing an integrated ecosystem for ethic-sensitive scientific discoveries and advanced applications of social data mining on the various dimensions of social life. We contributed to the SoBigData projects with the personal model for musical listening [134] illustrated in Section 8.2, and with the nowcast of the well-being level [130] reported in Section 11.2. Moreover, works under review are part of SoBigData.

[1] http://www.icon-fet.eu/

[2] http://www.petraproject.eu/

[3] http://www.eitictlabs.eu/innovation-entrepreneurship/future-cloud/

[4] http://www.cimplex-project.eu

[5] http://www.sobigdata.eu

Part I

Setting the Stage

Chapter 2

Data Mining and User Profiling

In this Chapter we recall the basic notion of *data mining* and we present an overview of *user profiling* models and methods. In particular, we analyze the current state-of-the-art with respect to *personal data models* defined to characterize mobility habits and shopping behaviors. Finally, since *clustering* is a technique widely used to extract user profiles through pattern detection, we accurately describe this task and its literature.

2.1 What is Data Mining?

Data Mining is one of the most important parts of the process of *Knowledge Discovery in Databases*, the so-called *KDD* process. The goal of this process is to transform raw data into useful information. The KDD process, as shown in Fig. 2.1, involves several steps: *(a)* data selection, *(b)* data preprocessing, *(c) data mining* to extract patterns, *(d)* and interpretation of the discovered structures through data analytics.

Data Mining is a technology that blends data analysis methods with sophisticated algorithms for processing large amounts of data. Data mining tasks can be divided into two main categories: *descriptive* and *predictive* tasks. Each category of tasks has different objectives of analysis and describes different types of possible data mining activities. Descriptive tasks have the goal of presenting the main features of the data: they essentially derive models that summarize the relationship in data, permitting in this way to study the most important aspects of the data. Predictive tasks have the specific objective of predicting the value of some target attribute of an object on the basis of observed values of other attributes of the object. Extracting useful information from personal raw data can be hard and challenging because of data collected by different and automated collection tools, and of the non-traditional nature of the data that becomes more and more complex. Indeed, *user profiling* is accomplished through the application of both descriptive and predictive data mining technologies. Among various data mining approaches, *clustering* is one of the most common and important, especially for user profiling.

2.2 User Profiling through Data Mining

User profiling refers to the process of construction and extraction of a personal data model representing the user behavior generated by computerized data analysis. A personal data model contains the systematic behaviors expressing the repetition of habitual actions, i.e.,

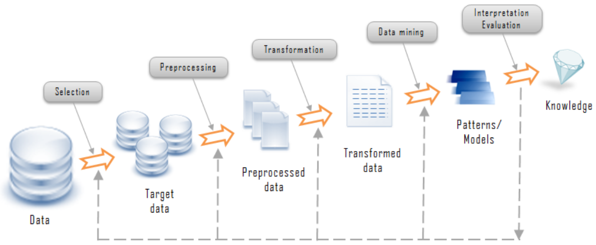


Figure 2.1: The process of Knowledge Discovery in Databases.

personal patterns. These patterns can be expressed as simple or complex indexes, behavioral rules, typical actions, etc. Users' profiles are employed on one hand to analyze and understand human behaviors and interactions. On the other hand, profiles are exploited by real services to make predictions, give suggestions, and to group similar users. Profiles can be classified as *individual* or *collective* according to the subject they refer to [146].

An *individual* or *personal* profile is a data model built considering the data of a single person. This kind of profiling is used to discover the particular characteristics of a certain individual, to enable unique identification for the provision of personalized services. A strong point of individual profiling is that the computation is generally not too expensive in terms of time and space because the data of a single user are limited. Conversely, a lack of knowledge can affect an individual profile: it could not consider a valuable pattern recognized by other users because it is not enough systematic for the individual. For example, if a user u goes occasionally to a lake on Sunday morning this movement can not be personally considered as a routine or a pattern if compared with the Home-Work-Home movements. However, if many people move from the same city of u to the same lake on Sunday morning, then this generates a pattern collectively recognized.

Traditionally, services are based on *global* profiling, i.e., each person is categorized or segmented within a certain class, relying on the fact that her behavior outlines with a data model formed by global patterns constructed on the basis of a massive amount of data related to many people. A weakness of global profiles is that they do not consider personal patterns because only the general patterns recognized by all the users emerges from the mass. Furthermore, it can be computationally hard to extract global profiles because a large amount of data must be considered all at the same time. Finally, global profiling requires every user to share all her raw data at the most detailed level.

We can talk about *collective* data models when personal models generated by individual profiling are aggregated without distinguishing the individuals. The difference with global models is that the collective profiles consider the personal patterns extracted from the individuals as a unique model, while in the global ones the patterns extracted are those related to the data of all the individuals, representing the behaviors of the mass.

Finally, we can refer to *combined* or *hybrid* models when two or more of the previous ones are merged in some way. Different models are generally combined to overtake weak points and to exploit strong points. An example of combined approach is the *hierarchical* one. It uses the individual profile as long as it is useful to solve the problem of a certain person, then it switches to collective patterns. Another example in stochastic applications is to mix the two different profiles according to a certain parameter or probability.

It is worth to notice that, the traditional approach for profiling a set of users through data mining techniques consists of the selection of a global parameters setting among all

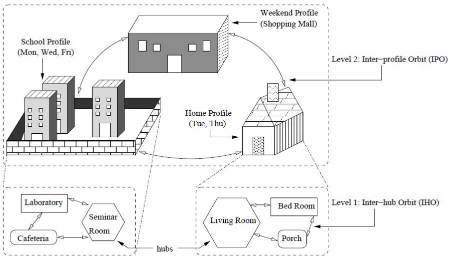


Figure 2.2: A hierarchical view of the list of places visited by a student.

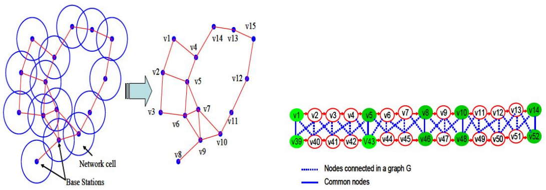


Figure 2.3: Covering graph (left) and two similar paths (right).

the users considered for the extraction of the data model. User profiling and personal data model extraction have been studied in various fields such as economy, politics, web, mobility and so on. In the following is deepened the description of users' profiles defined to capture behavioral patterns on mobility and shopping data.

2.2.1 Mobility Profiles

Despite user profiling has been deeply studied in fields like economy or in the World Wide Web, is still quite an emerging field of research with respect to mobility. Mobility user profiling was not recently investigated, while it was studied several years ago mainly to provide improvements in mobile phone services. Nowadays, thanks to the great availability of GPS and GSM data, mobility profiling is an open field of research.

Several works about profiling are related with wireless locations. In [113] it is extracted a movement pattern profile for each user. A *hub list* is the actual list of places visited by a student in a campus on the same day organized in a hierarchical way (see Fig. 2.2). Even if such hub lists may vary from one day to another, that variation is only marginal. Therefore, in most cases, a number of hub lists visited by the student over a period of days may be clustered together and represented by a single *weighted hub list*, where the weight associated with each hub denotes the probability of the student visiting that hub within that period. Such a weighted hub list then becomes the *student's individual mobility profile*. In practice, the authors exploit the fact that over a period of time a user repeatedly follows a mixture of mobility traces with a certain probability. Also in [7] a *user individual mobility profile* is a probabilistic model given by the combination of historical records and patterns of mobile terminals coming from a mobile wireless network with a cellular infrastructure. They are developed for estimating service patterns and tracking mobile users through the characterization of stochastic behaviors.

In [26], considering Long Term Evolution networks, it is presented the notion of *individual mobile user profile* which corresponds to frequent similar movements of a user. Such a profile is defined in the neighborhood covering graph of a cellular network (Fig. 2.3-left), as a set of similar sequences of crossed cells from one source cell to one destination cell. The objective of a user profile is to catch recurrent behaviors like going to the office or leaving it every day of the week at the same hours and using, more likely, the same paths. A major difficulty is that a profile does not necessarily translate into a unique path in the neighborhood graph of cells, repeating regularly the whole user behavior, but as a set of paths sufficiently close. The changes may come not only from the small variations in the

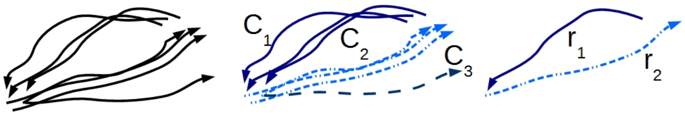


Figure 2.4: The *individual history* (black lines), the clusters identified by the grouping function (C_1, C_2, C_3) and the extracted *individual routines* (r_1, r_2) forming her *individual mobility profile.*

periodic movements of the user, but also from the variations in the propagation and identification of the crossed cells. As the behavior of the mobile user is not unique, each mobile user can be characterized by several paths that she generally follows when moving in the network coverage. In practice, the profile is defined as a set of similar paths which are pairwise closed and such that *(i)* they start at the same source node and end at the same destination node, *(ii)* they have at least a minimal number of identical nodes which appear in the same order in both paths, and *(iii)* if both paths have two different nodes, between two consecutive identical nodes, the two different nodes, then they should be connected in the covering graph (Fig. 2.3-right). The same authors in [27] introduce the notion of *global mobile user profile.* The local profiles are associated with a mobile user and correspond to its frequent and similar movements, whereas the global profile matches with the frequent and similar movements of the majority of users in the covered area. A global profile basically is a set of paths frequently followed by a certain percentage of mobile users in the network. The detection and the construction of the global profile follow the same procedure as local profiles. However, local profiles are not used for the global profile detection because a path could belong to a global profile without belonging to any local profile.

In [285] the authors present a methodology for extracting individual systematic movements from raw digital GPS traces. We adopt this methodology also in this thesis. Each movement of a user is described by a sequence of spatio-temporal points called *trajectory.* The set of all the trajectories traveled by a user makes her *individual history.* By defining a notion of spatio-temporal similarity between two trajectories they group the trajectories using a clustering algorithm (i.e., density-based Optics [15]) equipped with a certain trajectory *distance function*, and they obtain a partitioning of the original dataset from which they filter out the *clusters* with few trajectories and the one containing noise. Finally, they extract a *representative trajectory* from each remained cluster. These representative trajectories are called *routines* and the set of routines is called *individual mobility profile.* The mobility profile describes an abstraction in space and time of the systematic movements: the user's real movements are represented by a set of trajectories describing the generic path followed. Exceptional movements are completely ignored due to the fact they are not part of the profile. Fig. 2.4 depicts an example of mobility profile extraction.

In [153] user mobility profiles are built by using probabilistic suffix tree after transforming GPS trajectories in sequences of frequent regions. An *individual mobility profile* is a data structure which can organize the trajectory patterns of a user. They adopt a *probabilistic suffix tree* to manage the trajectories for the profile, and, in light of the tree structure, they propose a distance function to measure the distance between two probabilistic suffix trees. Explicitly, their design consists of *(i)* constructing trajectory profiles, *(ii)* formulating distance measurements among the trajectories of the profiles, and *(iii)* clustering similar trajectory into groups. Given the trajectories of a user, the frequent regions are first derived using a density-based approach, then is constructed a probabilistic suffix tree representing each trajectory into a sequence of frequent regions.

In the literature, we can find also works providing a dual vision. In [51] it is proposed a model based on both *individual and collective* behaviors. The model is based on the person's past trajectory and the geographical features of the area where the collectivity moves. The authors model *(i)* the propensity to change location, and *(ii)* the type of geographical areas that are of interest for the collectivity at a given time, both in terms of land use, points of interest and distance of trips. These features are assumed to be affecting the mobility choices as a proxy for activities. The idea of using collective behavior is not new, however, no information about geography has been combined so far. Trajectory patterns are instead considered to see whether different cars are moving in the same direction. They model the *individual behavior* as the probability of a cell j to be the next destination of a user in cell j equal to the frequency of visiting cell j starting from cell i during all the previous k periods considered. On the other hand, the *collective behavior* takes into account the collective behavior in two elements: distances being traveled, and types of places being visited. Since from the mobility traces they are not able to directly infer the activities that people make, they use information about an area's resources as a proxy for it. In other terms, they design the probability to choose a given destination to be a function of the distance of the destination, the presence of points of interests similar to the ones the collectivity has visited, and the type of land use the collectivity has been in. The authors define the user mobility profile as a probabilistic model obtained from the combination of the individual and collective models simply through a parameter $\alpha \in [0, 1]$ that weights the importance of the individual behavior rather than the collective one.

Trajectory pattern mining is introduced in [118] to extract the mobility behaviors. The authors propose a global model as concise descriptions of frequent behaviors, in terms of both space (i.e., the regions of space visited during movements) and time (i.e., the duration of movements). The new pattern, called *trajectory pattern*, represents a set of individual trajectories not necessarily belonging to the same user that share the property of visiting the same sequence of places with similar travel times. Therefore, two notions are central: *(i)* the regions of interest in the given space, and *(ii)* the typical travel time of moving objects from region to region. In fact, a trajectory pattern is a sequence of spatial regions that, on the basis of the source trajectory data considering all the users, emerges as frequently visited in the order specified by the sequence. In addition, the transition between two consecutive regions in such a sequence is annotated with a typical travel time that, again, emerges from the input trajectories. A trajectory pattern does not specify any particular route among two consecutive regions: instead, a typical travel time is specified, which approximates the travel time of each individual trajectory represented by the pattern. Moreover, the individual trajectories aggregated in a pattern are not necessarily simultaneous: it is only required that such trajectories visit the same sequence of places with similar transition times, even if they start at different absolute times. In this thesis, we employed this technique to extract personal mobility patterns.

2.2.2 Shopping Profiles

Customer profiling is a process widely used in economy since long time ago, before the coming of data mining. It can be used for direct marketing, site selection, and *customer relationship management*. This last one, enables the measurement from customers' purchases data to provide a 360° view of the client. Consequently, customer profiling can play today a very important role, from recommender systems to personalized prices.

Nowadays the market is characterized by being global, products and services are almost identical and there is an abundance of suppliers, and because of the size and complexity of the markets, mass marketing is expensive and the returns on investment are frequently questioned. Instead of targeting all the customers equally, a company can select only those customers who meet certain profitability criteria based on their individual needs and buying patterns [11]. To achieve this goal, the customers must be described by characteristics valuable for the business, like the demographic ones, the lifestyle, and the *shopping habits*. The aforementioned targets can be reached through *customer profiling*.

In customer relationship management it is worth to distinguish between two different branches: customer segmentation and customer profiling. *Customer segmentation* is a term used to describe the process of dividing customers into homogeneous groups on the basis of shared or common attributes (habits, tastes etc.) *Customer profiling* describes customers by their attributes, such as age, income, lifestyles and, in particular, by their shopping behavior. This last part of a customer profile can be identified as a *shopping profile*. Depending on data available, they can be used to prospect new customers or to "drop" out existing bad customers. The goal is to predict future purchases based on the information we have on each customer [34]. From a classical point of view, profiling is generally performed after segmentation. Segmentation offers to a company a way to know about loyalty and profitability of their customers. On the other hand, by knowing the profile of each customer, a company can treat a customer according to her individual needs in order to increase the lifetime value of the customer [11]. Furthermore, customer profiling is a key element which impacts into the decisions in product life cycle cost [99].

Customer Segmentation. Segmentation enables more targeted communication with the customers and describes the characteristics of groups of customers, called segments or clusters. Segmenting means partitioning the population into segments according to their affinity or similar characteristics. Customer segmentation can be a preparation step for classifying each customer according to predefined customer groups. Segmentation is essential to cope with today's dynamically fragmenting consumer marketplace. Through segmentation, companies are more effective in channeling resources and discovering opportunities [301]. The data mining methods used for customer segmentation belong to the category of *clustering* or *nearest-neighbors* algorithms.

Customer Profiling. Customer profiling provides the basis for companies to "communicate" individually with their customers in order to offer them improved personalized services and to retain them. Customer profiling is also used to prospect new customers using external sources, such as demographic data. These data are used to break the database into clusters of customers with shared purchasing traits [4]. Depending on the goal, a company must select what is relevant. Besides shopping data including also shopping frequency, preferences, lifestyle, attitudes, etc. needed to build the shopping profile, the features that can be used for a customer profile are geographical, cultural, ethnic, income, degree of satisfaction, age, beliefs, level of knowledge, media used, etc. [101].

In the following, we summarize some works on customer profiling from the literature which propose different personal data models based on associative rules and classification rules. In [4] it is described a system constructing personal profiles based on transactional histories. The system uses data mining techniques to discover a set of rules describing customers' behavior and supports human experts in validating rules. The *individual profile model* proposed has two parts: *factual* and *behavioral*. The factual part contains information, such as name, gender, and date of birth. The factual profile can also contain information derived from the transactional data, such as the last amount spent or favorite brand tastes. A shopping profile models the customer's purchases and is generally derived from transactional data. Examples of behavioral patterns in shopping are "When purchasing cereal, Bob usually buys milk". They model customer shopping with various types of rules, including association and classification rules. The use of rules as data model is an intuitive and descriptive way to represent shopping patterns, a rule is a well-studied concept used extensively in data mining, logic programming, and many other areas.

Another profile made by the item purchased by a client is proposed in [71]. This model is exploited to find segment of similar customers using a neighborhood algorithm. Then are observed the changes through time of the purchases within the segment and rules as in the case reported above are retrieved. The main point of this work is that it poses attention to shopping seasonality repeating the procedure for different seasons.

In [302] the authors attempt to analyze customers' purchasing behaviors based both on product profiles and customer profiles. The product profile is characterized by a set of features describing the product. The customer profile is basically an index expressing the level of interest in product features calculated using the product profiles. They use a two-stage clustering technique to find the group of customers that have similar interests and then extract rules from each cluster. We must notice that these kinds of profiling approaches are not limited to any specific representation of data mining rules or discovery method. However, because data mining methods discover rules for each customer individually, these methods work well for applications containing many transactions for each customer, such as credit card, grocery shopping, online browsing, and stock trading applications. In applications such as a car purchase or vacation planning, individual rules tend to be statistically less reliable because they are generated from relatively few transactions.

Furthermore, in customer analysis, there are several indexes such as *RFM* and *LTV* that can be regarded as the user profile. RFM analyzes customer value: it is commonly used in direct marketing and has received particular attention in retail and professional services industries. RFM stands for: *Recency*, how recently did the customer purchase, *Frequency*, how often do they purchase, *Monetary Value*, how much do they spend. User Life Time Value (LTV), is a prediction of the net profit attributed to the future relationship with a customer. The prediction model can have varying levels of accuracy, ranging from a crude heuristic to the use of complex predictive analytic techniques.

2.3 Clustering: A Data Mining Technique for User Profiling

Clustering is an unsupervised data mining technique. The task consists of grouping a set of objects or data event $X = \{x_1, \ldots, x_n\}$ in such a way that objects in the same group, called *cluster* $C_i \subset X$, are more similar to each other than to those in other groups. Thus a *clustering* $\mathcal{C} = \{C_1, \ldots, C_k\}$ is essentially a set of such clusters, usually containing all objects in the dataset [275]. In general, the notion of a *cluster* cannot be precisely defined, it depends on the kind of data analyzed and on the distance function used. This is one of the reasons why there is a great abundance of clustering algorithms in the literature. Typical clustering models can be classified in centroid models, distribution models, density models, graph-based models etc., [275]. Some of the most famous clustering algorithms are: K-Means [141, 196, 275], hierarchical clustering [159], DBSCAN [102], Optics [15] and Expectation-Maximization [92].

Clustering is a powerful tool, and, in particular, it is very useful for user profiling. It can be employed to extract personal data models because it is able to summarize and generalize users' behaviors. Given a collection of data representing observations of the user behavior, though clustering it is possible to group similar observations and summarize them. However, different types of data and different services require appropriate clustering models.

In the following, we report the state-of-the-art for applications of clustering algorithms with respect to personal data. The first application we analyze refers to the employment of clustering algorithms for detecting personal location, while the second one traits the extraction of clusters and representative transactions from transactional data. Both these applications, from a personal perspective, face the *parameter tuning* problem. Indeed, as enhanced in the following, generic clustering algorithms require parameters to be specified in order to return the best clustering. However, these parameters are data-dependent and, in addition, when large databases of users are analyzed they can not be manually tuned, nor estimation techniques can be used since they are generally time consuming. On the contrary, the clustering algorithms we propose in this thesis, follow the *personal data mining* approach: they are *autofocus* methods capable to automatically estimate the best parameter setting for each individual dataset while performing the clustering.

2.3.1 Clustering Personal Mobility Data

A common problem in mobility data mining, especially when mobility profiles must be built, is the clustering of stop points to *detect personal locations.* This task is generally addressed using "generic" clustering algorithms. However, in most cases, the clusterings produced are conditioned by some assumptions the algorithm makes about the data. Hence, it often works well on some datasets, yet behaves poorly on several others.

The work in [17] describes a predicting model built on locations automatically discovered and pushed into a Markov model. The authors use a variation of *K-Means* to detect the locations. K-Means [275] minimizes the distances between k cluster centroids and the points which are assigned to them. The main issue is the selection of the number of clusters k and of the initial centroids. This can be overtaken by making several runs with growing values of k. The best k can be selected by using the "rule of thumb" [138] or the k in the knee of the Sum of Squared Error or Silhouette curve [275]. Another limitation is that K-Means expects the clusters to be of similar size and isotropic shape. In [17] the authors select k as the first knee of the cluster radius curve. They find the knee by looking at the point in the curve that exceeds the average number of clusters w.r.t. some threshold. Thus, the parameter tuning problem is moved from k to the threshold. This threshold is not personalized for each user but it is fixed once for all.

The authors of [54] apply *Mean Shift* to detect the locations with the object of building a system to suggest touristic destinations on a large-scale geotagged web photo collection. Mean Shift [79] is a non-parametric clustering technique for locating the maxima of a density function. It is a hill climbing algorithm which involves the shifting of a kernel to reach a high density region. It often fails to find outliers or those points located between natural clusters. In an inner step, still in [54], the authors apply *Affinity Propagation* [104] based on the concept of communications between observations which locally decide in which cluster they belong to. Unlike clustering algorithms such as K-Means, it does not require the number of clusters to be determined or estimated before running the algorithm.

While the aforementioned methods do not need parameters estimation or it can be performed with established techniques, the algorithms employed in the following papers require at least a parameter related to the distance among the observations or with the distance for dividing the space. Obviously, the usage of such algorithms includes a tuning phase, which is usually time consuming and needs an analyst level expertise.

In [326] the *Grid* method is used to discover locations for recommendations. Grid [147] differs from clustering algorithms because it simply divides the space into cells of size ε and aggregates them considering their density. It has the obvious drawback of not considering the observations. The authors set the cell size at 300 meters to cluster the dataset without any test or tuning phase. This, in general, can greatly affect the results, since points belonging to different locations may have been put in the same cluster and vice-versa.

In [327] a variant of *DBSCAN* is employed, which considers also the time besides latitude and longitude to discover individual gazetteers. DBSCAN [102] is based on points classification with respect to the density around them. It takes the radius ε and the minimum number of points $MinPts$ as parameters. It can find arbitrarily shaped clusters and it is robust to noise. However, it cannot cluster well datasets with large differences in densities since the combination of ε and $MinPts$ in general cannot be chosen appropriately for all clusters. Also in this work, the authors ran the algorithm with a unique parameter setting for all the users ($MinPts = 10$ and $\varepsilon = 10$) and without any tuning.

In [131] it is employed *OPTICS* to retrieve the significant points of interest in daily life from GPS systematic movement data. OPTICS [15] is a variation of DBSCAN able to deal with clusters with different densities. The points are ordered with respect to the distance function, and a value for each point is derived, which represents the density needed to be accepted in a cluster for each point. The clusters, then, correspond to valleys in the plot of this distance with respect to the ordering. We had to perform an extensive tuning phase to obtain reliable locations capturing correctly human mobility.

Finally, in [226] the authors need to find home locations of the users analyzed, to investigate how known mobility models apply to car travels. They used *Bisecting K-Means* to detect the locations and then took the most visited one as *home*. Bisecting K-Means [272] repetitively applies K-Means with K=2 to subsequent partitions of the dataset. The clusters correspond to observations which are in the same partition of the lines (or planes) bisecting the dataset. Instead of fixing the final number of clusters to obtain, the method uses alternative stop criteria for the bisecting process, like the number of iterations or the maximum distance ε between observations in the same partition. In particular, the authors fixed ε=250 meters for all users, without considering individual behaviors.

We underline that the methods discussed above, and those proposed in this thesis, do not take into consideration geographical contextual information, such as existing points of interest, road network, etc. In literature, such information is typically used to assign single stops to a point of interest (e.g. [151]), mainly to enrich trajectories with activity information, rather than identifying recurrent locations. How to integrate that within a clustering-based location extraction process is an open problem that we leave as future work.

2.3.2 Clustering Personal Transactional Data

Transactional data is a special kind of categorical data in the form of sets of event data. A large amount of personal data generated by each individual consists of *transactions* like the items purchased in a shopping session, the web pages visited during a browsing session, the songs listened in a time period, etc. Clustering transactional data become essential when a user profile summarizing the user behavior must be built on these data.

In the literature, there is a great variety of papers proposing approaches to address the problem of clustering transactional data. Most of the existing algorithms require the setting of parameters which often may be difficult to be tuned. The first algorithm proposed for clustering transactional data is *large item* [298]. It requires a support threshold indicating the minimum number of occurrences for an item to be considered “large”, i.e. to be representative for a cluster. Through a scanning, it exploits the *large* items in a global cost function to evaluate the destination cluster or the creation of a new one. Several other algorithms like *rock* and *clope* were proposed with the same scanning strategy but different cost functions requiring parameters difficult to be interpreted [125, 311, 312].

Also in transactional clustering the most common parameter is the number of clusters [152, 304, 319]. *Tkmeans* [115] is one of the first attempts to use a different tranasctional clustering strategy by following the *K-Means* [275] approach. Another algorithm for extracting centroids from categorical data is *k-modes* [53]: it employs the mode instead of the mean. Finally, *purtreeclust* [66] is a recent method working on categorical data for clustering customers through their purchase trees which are built on the customers’ transactions.

A further notion used in cost functions is entropy [14, 24, 62, 189]. In [24] the authors propose *coolcat* that iteratively chooses the suitable cluster for each transaction such that at each step the entropy of the resulting clustering is minimized. A similar procedure is followed by the algorithm *limbo* in [14]. It uses the notion of entropy to identify the similarity between data objects and the clustering process minimizes the information loss. A dual approach is proposed in [189] where starting from a single cluster a Monte Carlo process selects a transaction and to assigns it to another cluster to decrease the entropy.

Besides parameter tuning, when dealing with transactional data another problem is high dimensionality. It is often necessary to transform the original dataset into a boolean dataset. Typically it makes algorithms inefficient in terms of execution time and clustering quality. To not suffer from high dimensionality, *subspace* clustering algorithms like [110, 320] have been proposed with the goal to find clusters embedded in subspaces of the original data space with their own associated dimensions. In the literature, some algorithms like [35, 93] use bipartite graph theory to cluster datasets. They generate co-clustering results where columns and rows are simultaneously partitioned. In transactional data, this means an unnatural split of the clusters that overlap over a few frequent items. However, they are often memory and time consuming, and inappropriate for clustering large datasets.

Some proposals were made to overcome manual parameter tuning and to automatically select the number of clusters. In [62] it is proposed an entropy-based clustering working in a bottom-up manner. It evaluates the similarity with incremental entropy, and finally, generates a clustering tree containing clusterings with different number of clusters. The authors of [308] propose to run their algorithm with a different number of clusters and choose the result that optimizes a specific index of quality. However, in terms of execution time these methods are clearly inefficient. The first parameter-free transactional clustering algorithm is *atdc* [60]. It adopts a top-down strategy resembling a decision tree learning algorithm. In [44] it is proposed the *practical* parameter-free method that through scanning automatically identifies clusters even in presence of rare items. Finally, also the *dhcc* algorithm presented in [305] is a parameter-free procedure based on a divisive hierarchical clustering approach. However, *dhcc* is especially designed for working on classical categorical data rather than on transactional data.

Chapter 3

User Profiling for Individual and Collective Services

In this Chapter we show how personal data models can be exploited by personal services. A classic usage of a user profile is the *prediction* of future actions. By accounting on the systematic repetitions of a user, valuable predictions of what a user is going to do in the future are enabled by recognizing that the user is acting by following one of her habitual behaviors. Besides prediction, user profiles are typically adopted in *recommendation systems*: personal suggestions are provided to the user according to her data model. In both cases there are two distinct phases: *extraction time* when the personal data model is built or updated, and *query time* when the profile is used for prediction or recommendation. The former can require a while but is repeated not frequently because a profile is assumed to be valid during predefined time intervals. On the other hand, the second one needs to be fast in order to return a prediction or a recommendation as soon as possible.

3.1 Predicting Human Behavior

A *prediction* or *forecast* is a statement about the way things will happen in the future, often but not always based on experience or knowledge. Although guaranteed information about the future is in many cases impossible, prediction is necessary to allow plans to be made about possible developments. Indeed, human predictability can be used to plan events and infrastructures, both for the public good and for private gains. Predictability is a vast research field, tackled with a number of approaches and for a number of different reasons. In the following we present a literature review about prediction methods for services with respect to the fields of mobility data and shopping transactional data. They all make use of different notions of user profiles and personal data models.

3.1.1 Prediction in Mobility Services

The approaches proposed in the literature for location and trajectory prediction can be classified on the basis of the prediction strategy used. In the literature, a lot of works addressing the location prediction problem propose methods that base the prediction only on the movement history of the object itself [10, 59, 122, 157, 167, 193, 221, 257, 283, 300, 310]. We say that these approaches use the *individual strategy* for the prediction of user future positions. Some approaches of this category adopt time series analyses [59, 257] to forecast user behavior in different locations. Time series analyses enable estimations as the time of the future visits and expected residence time in those locations [257]. In this kind of works, it is necessary to define the set of interesting locations to be considered in the analysis. In [59] these locations are areas statically defined, while [257] provides a method for extracting significant locations among which users move frequently.

Others prediction approaches are based on Markovian processes [221] and on machine learning techniques such as classification [10, 283]. In particular, in these two last works the location prediction problem is treated as a *classification problem*: in [10] the location information considered for classification refers to the history of user movements, that is represented by a vector of h time-ordered locations crossed by a user; while in [283] the classification tree is built based on simple, intuitive features extracted from the user visit sequence data with associated a semantic meaning. In [193], in order to capture aspects of the individual's mobility behaviors, the authors propose a modified Brownian Bridge model that incorporates linear extrapolation. Other works such as [157, 167] provide methods for the prediction of the movement ahead of a moving object whose movement is constrained to a road network. In [167] the authors assume that the objects' destinations are known. Considering the road network most of the works in this category transform the trajectory into a path on the graph representing the road network. This leads to a form of spatial generalization that we do not apply in our methods. Finally, some works combine historical spatial and temporal data about the user with contextual data such as accelerometer, bluetooth and call/sms log [122] or with social relationships with friends [310]. These approaches are different from those proposed in this thesis because we employ only spatio-temporal data from user's movements without any other additional information.

The main problem of approaches implementing the individual strategy is that they fail in predicting future locations of non-systematic users. In these cases applying a *collective strategy* could improve the prediction. Prediction approaches belonging to this category first extract mobility behavior for each user considering only the user's movement history, like in the individual strategy, and then they merge all the individual models for the construction of the predictor [89, 175, 315]. The main difference with our work is that we do not apply any spatial generalization on movements data while these works typically use a grid for obtaining cells instead of points like in [175], or extract semantic places from raw data by grouping different spatial coordinates that identify a stop [315].

Other approaches address the location prediction problem by using a *global strategy*, i.e., they extract movement behaviors from the movement history of all the users in the database and use this global knowledge to forecast the next location visited by a specific moving object. The basic assumption, in this case, is that people often follow the crowd, i.e., individuals tend to follow common paths. This strategy was followed in many papers; most of them extract frequent patterns and association rules from data [63, 156, 162, 183, 188, 197, 211, 215, 216, 314] using methods based on *Apriori*, *PrefixSpan* and *FPGrowth* techniques. Some recent works instead use probabilistic models and in particular Markovian models [48, 120, 243, 306]. Some of these approaches are suitable for predicting the next location by using GSM data [162, 188, 197, 314]; while others work well with GPS data [63, 120, 156, 183, 211, 215, 216, 243, 306]. Solutions based on GPS data typically apply a spatial discretization to make easier finding frequent or interesting locations. Two main types of discretization are applied: the first one extracts interesting places applying density based clustering techniques [156, 183, 188]; while the second one simply uses a grid on the space, determining for each trajectory the sequence of intersected cells [63, 120, 183, 211, 215, 216]. We highlight that the methods we will present in this thesis differ from these works because we do not apply apriori spatial discretization allowing us to predict the *exact* positions visited. Moreover, contrary to most of the above approaches, we use the temporal information both during the data model extraction and during the prediction. Some exceptions are [156, 183] which allow choosing the prediction time specifying the temporal information. Others work such as [211] base their approach on trajectory patterns which are intrinsically equipped with temporal information.

Another interesting way to exploit user mobility information for predicting the next user location is based on the idea to combine the *global* and *individual* strategies in order to obtain more accurate predictions. In particular, the idea is to have a global predictor constructed using all users' mobility data and for each user also producing a predictive model based only on her individual movements. Therefore, during the prediction the idea is to use one of these two predictors: when using the individual predictor is not possible to provide a valid and accurate prediction then the global predictor is used [17, 27, 64, 324]. Their basic idea is similar to our *hybrid strategy* where we provide the possibility to combine individual prediction with either the collective one or the global one. However, our methods differ from [17, 27, 324] also for the spatial precision of the predictions. Indeed, [17] it is based on GPS data but applies a discretization based on clustering; while the others are based on GSM data. In [96] it is used a global model to improve the personalized model: the prediction score that is a combination of the global score and individual one.

All these methods are tested on synthetic data or on small real dataset. We differ from them in testing our approaches in a big data context using large real-world datasets.

3.1.2 Prediction in Shopping Services

As for mobility data, there are various works trying to predict customer shopping behavior. Data mining [6, 165] is the best choice because it can be difficult to create a comprehensive model of overall customer behavior, as each single individual acts according to a very nuanced and personal utility function. Multiplex approaches are then used [67, 234]. Moreover, recent research showed that it is possible to describe the retail market as a complex system [235]. These works focus on the detection of regularities in what customers buy.

Interesting studies analyze purchasing transactional data to predict changing in the customer behavior [65, 270], and if a customer will switch from one brand to another [139]. In particular, in [65] there is an attempt to integrate typical customer behavioral variables such as recency and frequency with transactional data to establish a method for predicting changes in shopping behavior. In [139] it is developed a method for extracting useful knowledge from individual customers' purchase histories by combining information fusion techniques with data mining to predict whether a customer switches from one brand to another, or becomes loyal to a brand. In [261] the author show how signals of RFID can be exploited to detect and record how customers browse stores, which items of clothes they pay attention to, and which items of clothes they usually match with.

One of the most challenging goals with respect to shopping services is the prediction of the customer shopping list. The shopping list prediction can be developed to provide individual and personalized interactions with customers as they "navigate" through the retail store. In [87] it is described a prototype that predicts the shopping lists for customers in a retail store. Instead of using traditional approaches such as clustering or segmentation, they exploit the massive amounts of data captured, and the relative high shopping frequency of a grocery store, to build separate classifiers for every customer. This allows them to make very fine-grained and accurate predictions about which items a particular individual customer will buy on a given shopping trip. Thus, in this case, the customer profile corresponds to the personal classifier built. On the contrary, in [165, 289] are introduced two methods using supervised learning but that can be categorized as collective since they do not process each customer individually. In [165] the authors propose a prediction method for customer purchase behavior that combines in parallel several classifiers using genetic algorithms. In [289] it is applied an approach using a mix of classification, decision rules and probabilities to extract customer behavioral patterns. Since the prediction of the shopping list is more commonly considered as recommendation task, it is treated more deeply in Section 3.2.

A promising line of research investigates where customers go to buy what, i.e., how much the shops they visit are predictable [176] and how much they are willing to travel to satisfy their needs [94, 236]. In both cases, customers are shown to be rather predictable in their movements. The mobility dimension is very important when analyzing personal shopping for two reasons. First, it resulted to be highly predictable [222, 269]. Second, it is intimately linked with the social dimension. According to the previous Section, it has been proved that it is possible to predict the places an individual will visit because we know that their friends visit them and that social ties are more easily created among people who travel to the same places [69, 89, 280, 297]. The predictability of the creation of new social ties by an individual is a classic problem in social network analysis [190, 258].

Another type of data relative to shopping sessions and economy and strongly related with prediction is time series. Support vector machines (SVM), is a technique widely used especially to forecast financial time series [55, 166]. In [166] the authors try to predict the stock price index while in [55] it is studied the feasibility of SVM in financial forecasting with respect to neural networks methods. In these approaches, all the data are considered because is estimated the general situation of the market. However, in the same way, individual indexes could be evaluated by analyzing as input the shopping of each customer. This is done in [49, 293] where the authors try to predict the customer potential value in order to estimate the probability of churn, i.e. the expected degree of abandonment of a certain product or of a certain company from their habitual usage.

Moving from time series, we want to consider the temporal dimension of shopping. In the literature, there are various works which try to predict the shopping behavior. However, as reported above, they generally take into consideration aspects related to the items bought and not only the time. The temporal component of customer purchases is especially analyzed in [194, 203, 307]. In [307] the authors propose to represent the customer purchasing behavior using a directed graph retaining temporal information in a purchase sequence: they apply a graph mining technique to analyze the frequent occurring patterns. The authors of [203] examine the role of personal characteristics in time spent shopping. In particular, is analyzed the roles of time perceptions, brand and store loyalty, social, physical well-being, and demographic variables in predicting reported shopping time, including the hours spent at search and purchase. [194] studies changes in cluster characteristics of supermarket customers over a 24 week period by performing a temporal analysis that tries to detect the migrations of the customers from one group to another group. The temporal analysis presented is based on conventional and modified self-organizing maps. The personal models and methods we propose in this thesis, not only help in understanding the changing in customers' shopping behaviors and their cyclic succession, but also in unveiling and measuring the regularities of these changing if detected.

Finally, there are works aimed at understanding the behavior of customers in online shopping [140, 192, 224, 288]. Note that in the works reported above the customer behavior is generalized at global level, while the data model we propose in this thesis to treat the user shopping behavior is a personal description of the customer habits and preferences.

Nowcasting. Nowcasting is a novel promising field of research. It has been successfully combined with the analysis of large datasets of human activities. Nowcasting is generally a collective or global approach since it requires a data model which capture the behavior of a mass of people. Two famous examples are Google Flu trends [303] and the prediction of automobile sales [74]. Social media data has been used to nowcast employment status and shocks [195, 281]. Such studies are not exempt from criticisms: [180] proved that nowcasting with Google queries alone is not enough and the data must be integrated with other models. Nowcasting has been applied also to GDP [114]. However, the model uses a statistical approach that is intractable for a high number of variables, thus affecting the quality of results. Other examples can be found on the Eurozone [103], or on different targets such as poverty risk [219] and income distribution [186].

3.1.3 Prediction in Social Network

In social network, the prediction problems are treated as *link prediction*. Link prediction strategies may be broadly categorized into four groups: similarity based strategies, maximum likelihood algorithms, probabilistic models and supervised learning algorithms [198].

The first group defines measures of similarity as a score between each pair of nodes. All non-observed links are ranked according to their scores, and the links connecting more similar nodes are supposed to be of higher existence likelihoods. Despite its simplicity, the definition of node similarity is a non-trivial challenge.

The second set of methods is based on maximum likelihood estimation. Empirical studies suggest that many real-world networks exhibit hierarchical organization. Indeed, these algorithms presuppose some organizing principles of the network exploited with the rules and parameters obtained by maximizing the likelihood of the structure. From the viewpoint of practical applications, an obvious drawback of the maximum likelihood methods is that it is very time consuming and not among the most accurate ones.

The third group of algorithms is based on probabilistic bayesian estimation. Probabilistic models aim at abstracting the underlying structure from the observed network, and then predicting the missing links by using the learned model. Given a target network, the probabilistic model optimizes a built target function to establish a model based on a group of parameters which can best fit the observed data of the target network. Then the probability that a non-existent link will appear is estimated by the conditional probability.

The last group of methods employs supervised machine learning techniques. Link prediction through supervised learning algorithms was introduced in [190]. They studied the usefulness of graph topological features by testing them on co-authorship networks dataset. In these type of methods, a set of similarity features is proposed for each couple of node. Then, knowing if a link will be present or not in future, a classifier is trained and then used to predict new links. After [190], more recently new models have been proposed using very different approaches. In [39] are used topological features and node attributes in linear combination applied to a covariance matrix adaptation evolution strategy to optimize the prediction. Principal component regression is the algorithm used in [21] to determine the weight of statistically independent predictor variables used for the prediction. A rank approach is proposed in [241] to order unlinked nodes according to some topological measures. At the new instant time each measure is weighted according to its performance in predicting new links. In [263] the authors use textual and topological features to predict new citations applying an SVM as supervised learning method. Finally, in [271] it is used tensor factorization to select the more predictive attributes.

3.2 Recommendation Systems

Recommendation systems are a subclass of information filtering systems that seek to predict the *rating* or *preference* that user would give to an item [248]. They have become extremely common in recent years, and are applied in a variety of services. The most popular ones are probably movies, *music*, news, books, research articles, search queries, and products in general. A personalized recommendation system can help enterprises launch *one-to-one marketing*, i.e., individual marketing. The purpose of launching one-to-one marketing is to increase customer loyalty and to enhance selling [154]. Differently from traditional marketing which concentrates on “pushing” products to customers, one-to-one marketing focuses on understanding customers gradually to actively fulfill their needs by recommending appropriate products or services [302]. The well-known Pareto $20 - 80$ principle states that the 20% of the customers may generate as much as 80% of the company’s profits. Therefore, in this thesis, we focus on how to provide one-to-one services or products for existing customers. Recommendation systems typically produce a list of recommendations in one of two ways: through collaborative or content-based filtering.

Collaborative filtering approaches are based on collecting and analyzing a large amount of information on users’ behaviors, activities or preferences and employing them to build a model representing the user’s past behavior. Then, that model is used to predict what a user may have interested in based on her similarity with other users [45]. A key advantage of collaborative filtering is that it does not rely on machine analyzable content, and therefore, it is capable of accurately recommend complex items without requiring an “understanding” of the item itself. However, despite their success, their use has been constrained by two major limitations. The first limitation is the *sparsity problem* [68]. Conventional collaborative filtering recommendation systems require users to explicitly input preference ratings about many products. The number of ratings received is relatively small compared to the number of ratings required for prediction. Consequently, predicted ratings accuracy degrades significantly when the received ratings are sparse. The second limitation is the *scalability problem* [72]. As the number of users and targets increase, the computation time of algorithms, which perform product comparisons grows, respectively [112].

Content-based filtering approaches employ a series of discrete characteristics of an item in order to recommend additional items with similar properties [214]. These approaches have its roots in information retrieval and information filtering research. Content-based filtering methods are based on a description of the item and a *profile* of the user’s preference [47]. In other words, these algorithms try to recommend items that are similar to those that a user liked in the past: candidate items are compared with items previously rated by the user and the best-matching items are recommended [317]. A weak point of the content-based approach is the number of features that should be chosen to describe an item: the quality of the system is affected by the feature selection process. An example of profile in content-based recommendation systems can be found in [230] where is presented a system that creates a profile of the user that describes the types of items the user likes, and provide a means of comparing items to the user profile to determine what to recommend. The profiles are often created and updated automatically in response to feedback on the desirability of items that have been presented to the user.

These approaches are often combined in *hybrid recommendation systems*. Recent research has demonstrated that a hybrid approach, combining collaborative filtering and content-based filtering, could be more effective in some cases. Hybrid approaches can be implemented in several ways: by making content-based and collaborative-based predictions separately and then combining them, by adding content-based capabilities to a collaborative-based approach (and vice versa), or by unifying the approaches into one model [5, 50]. These methods can also be used to overcome some of the common problems in recommendation systems such as *cold start* and the *sparsity problem*.

In the above description, the treatment was made mainly from an economical point-of-view. Indeed, we used terms such as products and items. However, the entire treatment could be reformulated using mobility terms such as locations and regions, or social terms such as interactions and users. In the following, we report the state-of-the-art about recommendation systems in the field of mobility, shopping, and music.

3.2.1 Recommendation in Mobility Services

In [228] the authors propose a location-based recommendation system using bayesian user's preference model in mobile devices. Adopting the collaborative filtering approach, they exploit mobility user profiles based on factual data and movement preferences to suggest new destinations that should like the drivers. This idea is confirmed by [247] which illustrates that there are two groups of factors that influence destination choice: personal features and travel features. The first group contains both socioeconomic factors (age, education, and income) and psychological and cognitive ones (experience, personality, involvement, and so forth). The second group might list travel purpose, travel-party size, length of travel, distance, and transportation mode. The recommendation system introduced in [46] tries to integrate classical information retrieval information on web pages, like the number of clicks per page, the time spent in a page etc., with mobility information, like the position of the user when visiting a certain page. They use a historic database of locations and corresponding links used in the past by a set of users, that is profiles, and develop models relating resources to their spatial usage pattern. These models are used to calculate a preference metric when the current user is asking for resources of interest being in a certain location. Something similar is described in [316], where a procedure based on offline and online part is presented. The authors, in this case, recommend entire itinerary exploiting other users past experience with location interest graphs and according to classical travel features: elapsed time, stay time and interest ratio.

Route Planner. A particular type of recommendation systems in mobility are *route planners*. They are designed to provide information about the possible journeys in a certain area. Generally, route planners refer to means of transportation which are either private or public. The application prompts a user to input an origin and a destination and it recommends some routes which are considered to be the best for that query.

Route planners generally use some smart variations of well known shortest path algorithms to search a graph of nodes and edges [185]. Different cost weights such as distance, cost etc., can be associated with edges and nodes. However, it is generally quite difficult to plan high-quality routes [204]: *(i)* the notion of "route quality" is different from person to person, and *(ii)* available route networks rarely contain all the information needed for proposing the best route (e.g. traffic information, road quality etc.). Thus, even though the search can be optimized with respect to different criteria, e.g. the shortest, the fastest, the cheapest [233], and even the happiest ones [244], there is not guarantee that the route provided will be considered "the best" by the majority of the users.

Various efforts in different directions have been made to improve route planning applications. In particular, personalized route services able to deal with individual users preferences have been investigated recently. For example, in [209] complex users preferences are modeled into a route planner by means of the fuzzy set theory. In [191] the authors provide improved individual route plans for Dublin inhabitants by exploiting both historical data and estimated traffic flows. Similarly, according to an estimation of future travels obtained by mining public transport data, in [177] are recommended personalized tickets for London public transport network., and [318] introduces real-time information coming from GPS-equipped taxi together with historical data for an improved route planner which uses traffic conditions and driver behavior for selecting the best path. Finally, a multi-modal journey planner can consider at the same time various means of transport and minimize the uncertainty of catching a certain means [43], or it can provide for the same journey personalized public and private transportation solutions [42].

Carpooling. Besides recommending locations and paths, emerging recommender systems in mobility services are services related to *carpooling*. Nowadays, there are many websites already operative throughout the world. All of them allow the user to register, search for a ride and offer a ride. Anyway, they present several differences. Drivebook, Roadsharing and Blablacar[1] are some of the most famous ones because they are international, offering intra- and inter-country services. Indeed, they treat mainly long and occasional trips. Drivebook is characterized by the feature of being linked with various social networks to improve the confidence among users, while Roadsharing focuses on commuters.

The carpooling phenomenon is a subject widely studied in the literature. It has been analyzed from various, very different points of view. Carpooling is the second most popular way of commuting, and maybe one of the least understood – a fact that probably explains the need for such a large corpus of studies in literature.

An approach widely followed in the literature to analyze carpooling is Agent Based Modeling (ABM) [16, 30, 70, 107, 108, 260]. A multi-ABM in conjunction with the Dikstra's algorithm is used in [260] to efficiently answer real-time users' queries. In [16] it is designed an ABM to optimize transports by the ride sharing of people who usually cover the same route. The information obtained from this simulator are used to study the functioning of the clearing services and the business models. In [30] the authors face the problem by using a multi-ABM to investigate opportunities among simulated commuters and by providing an online matching for those living and working in close areas. [70, 109, 107] present a conceptual design of an ABM for the carpooling application to simulate the interactions of autonomous agents and to analyze the effects of changes in factors related to the infrastructure, behavior, and cost. They use agent profile and social networks to initiate the ABM, then employ a route matching algorithm and a utility function to trigger the

[1] http://www.drivebook.com/,http://www.roadsharing.com/,http://www.blablacar.com/

negotiation process between agents. In [108] the authors define an ABM for the individual mobility behavior during carpooling, the criteria and the function to constitute the carpooling community and a protocol for the negotiation of the details of the carpooling trips.

Many carpooling works are related to the *study and analysis* of mobility data to understand the carpooling phenomenon [76, 82, 182, 273, 276, 285, 290, 291]. In [276], for example, the authors describe the characteristics of carpoolers, distinguishing among different types of carpooler, and identifying the key differences between a carpooler, a Single Occupant Vehicle (SOV), and a transit commuter. They also describe how and why commuters carpool. In [285], it is introduced a methodology for extracting mobility profiles of individuals, and criteria to match common routes in order to develop a carpooling service. In [76] the authors derive home and work locations using Twitter and Foursquare data, then social ties are used to develop an algorithm for matching users with similar mobility patterns. [82] proposes a study club model to overtake psychological barriers associated with riding with strangers, to find compatible matches for traditional groups of users and also to find a ride in alternative groups. Using a multilevel regression model and a questionnaire which explains the share of carpooling employees at a workplace, [290, 291] predict the share of carpooling at large workplaces locations, organization and carpooling promotion. In [36] the authors develop an application for car sharing recommendation by exploiting a topic clustering algorithm applied to labeled trajectories.

In other studies [80, 169, 182, 202], the authors try to find simulated or theoretical *matches* among users asking for a ride in a carpooling scenario and evaluate it in terms of simulated users' feedbacks. [202] develops and implements the concept of real carpooling by allowing a large base of member passengers and drivers that declared their route to be matched against each other automatically and instantly using mobile phone calls. In [80], the problem is faced as an optimization task reduced to the *chairman assignment problem* [279]. [184] considers simulated straight-line trajectories observing only origin and destination of trips and classifies users as eligible or ineligible for carpooling by minimizing the time of the trip. In [169] it is built a user network that represents planned periodic trips, where the edges are labeled with the probability of negotiation success for carpooling. The probability values are calculated by a learning mechanism using the registered person features, the trip characteristics, and the negotiation feedback. The algorithm provides advice by maximizing the expected value for negotiation success. The differences between the approach proposed in [169] and ours is that we provide matches between couples of users in a pro-active way, suggested from data and not advertised from people. Moreover, [169] uses the network structure to model the negotiation feedback process, while our approach uses complex networks to model possible carpooling interactions to recommend possible assignments by taking into account real trajectories and systematic movements. [182] develops a methodology that finds feature points in trajectories and organizes them in a trie data structure to speed up and refine geographical queries for carpooling purposes.

3.2.2 Recommendation in Shopping Services

By exploiting both customers and products profile, and by analyzing customer's preferred brand or product, in [154] the authors present a recommendation system for one-to-one marketing able to suggest products to customers either at general or at specific level. In particular, it is used a product taxonomy to identify customers' shopping behavior in the following classes: product addictive, brand addictive or hybrid addictive. Also in [56], it is presented a fuzzy-based algorithm for a web marketing system that, by exploiting the features of a product, it retrieves the optimal products for the customer's current needs obtained from the system-user interactions. The authors of [256] propose an item-based collaborative filtering technique to overtake one of the main issue of traditional collaborative filtering systems: the amount of work increases with the number of participants in the system. They first analyze the user-item matrix to identify relationship between different items, and then use these relationships to indirectly compute recommendation for users. In [323] it is proposed a recommendation system aimed at maximizing customer satisfaction. By employing an associative classification method based on customers' characteristics, the customer's next product is predicted if the model has a high level of satisfaction.

With respect to the recommendation of products for the customer shopping list, [206] proposes a memory-based collaborative filtering method on transactional data as an extension of [256]. In particular, it investigates the suitability of such method for situations when only binary pick-any customer information (i.e., choice/non-choice of items, such as shopping basket data) is available. The authors of [246] learn the general taste of a user by factorizing the matrix over observed user-item preferences. Then, Markov chains are used to model sequential behavior by learning a transition graph over items. The chain is used to predict the next action based on the recent actions of a user. The personal transition graph over underlying Markov chains corresponds to the user profile: for each user a transition matrix is learned. Thus, in total, the method uses a collective transition cube. Finally, given a user's purchase history, the method proposed in [299] employs a hierarchical representation model for next basket recommendation. The model simultaneously considers the sequential behavior, i.e., buy one item leads to buying another next, as well as the users' global taste, i.e., what items a user is typically interested in.

Musical listening can be considered as a special kind of transactional data, thus treatable as shopping session. The treatment of musical listening is becoming valuable because in the last decade the music world has started receiving more attention from the scientific community. In [237] the authors measured different dimensions of social prominence on a social graph built upon 70k Last.Fm users whose listening were observed for 2 years. By analyzing the *width*, the *depth*, and the *strength* of local diffusion trees, the authors were able to identify patterns related to individual music genres. In [225] the authors formally defined the effect of social influence providing new models and evaluation measures for real-time recommendations with very strong temporal aspects. The authors of [242] analyzed the cross-cultural gender differences in the adoption and usage of Last.Fm: *(i)* men listen to more pieces of music than women, *(ii)* women focus on fewer musical genres and fewer tracks than men. In [37] the authors studied the topology of an online musical social graph asking for similarities in taste as well as on demographic attributes and local network structure. Their results suggest that users connect to "online" friends, but also indicate the presence of strong "real-life" friendship ties identifiable by the multiple co-attendance to the same concerts. All the global knowledge gathered from these analysis constitutes the key features to improve recommendation of musical listening.

Chapter 4

Personal Data Store: A User-Centric Model

In this final background Chapter we outline the concept of *Personal Data Store (PDS)* through an overview of the recent literature. A Personal Data Store is a digital space where each user can store her personal data. It acts as an intermediary between the user and all the external services requiring user's data. The goal of PDS is to allow the user to control her own personal data and to manage authorizations for third-path services in order to give everyone the right of managing personal information and communications freely and safely. The user decides if, what, and to whom she wants to share her personal data.

4.1 Towards a User-Centric Model

The raise of smartphones and web services together with their increasing usage on everyday activities is making possible the large-scale collection of personal data. The availability of data about people, which enclose information on their choices, preferences, actions, etc., represents an invaluable opportunity for the release of personal services. Currently, these data are gathered and managed mainly by big companies, which keep and exploit users' information necessary for offering various services. However, the so-called *organization-centric model* does not permit to take fully advantage from the potentiality of knowledge extraction offered by personal data. This happens because of various reasons:

- there are legal implications for possible sharing of data, i.e., entities that own data tend to keep them locked;

- data, and the knowledge related to them, represent a useful and valuable good, thus companies do not intend to share information with others;

- each company has only a limited view of individuals, i.e., the dimension described and captured by the data correlated with its activities (mobility, shopping, social, etc.), in such way it is not possible to exploit linking among different dimensions;

- users have a limited capability to control and exploit their personal data, thus they often exhibit skepticism and do not give access to data unless it is strictly necessary.

In order to overcome the problems listed above, it has been recently proposed a change of perspective towards a *user-centric model* for personal data management. This “vision” is compatible with the one promoted by the World Economic Forum [161, 239, 250].

In this model, the user acquires a central influence and gains an active role through a full control on the lifecycle of her own personal data. The basic idea is to introduce high levels of transparency regarding services and real use of data, to enable individuals to control a copy of their data and finally to give them the right to dispose or distribute data with the desired privacy level. Since most of the existing solutions consider an architecture where a central service provider releases the data to the final users only after having made the data private, this idea of user centrality brings to a change of perspective for privacy problems. Indeed, in the *user-centric model*, the privacy transformation is applied at individual level before the data sharing. The personal control of private data should encourage the users towards a voluntary participation by limiting the common suspicion associated with data sharing, and by augmenting the awareness of the profits that can be gained by extracting knowledge from personal data, both at personal and collective level [119].

The growing quantity and quality of personal data create enormous value for the global economy [250]: personal data plays a vital role in countless facets of our lives. Medical practitioners use health data to better diagnose illnesses, develop new cures and address public health issues. Individuals are using personal and collective data to find relevant information and services, coordinate actions and connect with people who share similar interests. Governments are using personal data to protect public safety, to improve law enforcement and strengthen national security. Businesses are using personal data to innovate, create services and design new products that stimulate economic growth. Moreover, the emerging communication systems are democratizing the access to information.

Understanding these systems is necessary to make our future stable and safe. We are getting beyond complexity and data science because we are including people as a key part of these systems [238]. As we begin to understand them, then we can build better systems, systems which encounter our needs, systems which are personalized. The promise is to have financial systems that do not melt down, governments that do not get mired in inaction, health systems that actually work, etc. Thus, we need technologies which respect the needs of involved actors and prevent the interesting content of the data.

4.2 What is a Personal Data Store?

The user-centric paradigm can be enabled by empowering individuals with the ability to control a copy of their personal data, the so-called *right-of-copy*, and by giving to them the right to dispose or distribute her data for receiving the desired services.

A *Personal Data Store (PDS)* is a personal, digital identity management service controlled by an individual. It is based on the *user-centric model* which gives to the user a central point of control for their personal information like interests, contact information, affiliations, preferences, friends, mobility, shopping, music, etc. The user’s data managed by the service may be stored in a co-located repository, or they may be stored in multiple external distributed repositories, or a combination of both. Users may be allowed to share portions of data with other users. However, while the question of data ownership and the creation of PDS have been discussed for a long time [29, 150] their deployment on large scale is still an open problem. Privacy and legal concerns, as well as the lack of technical

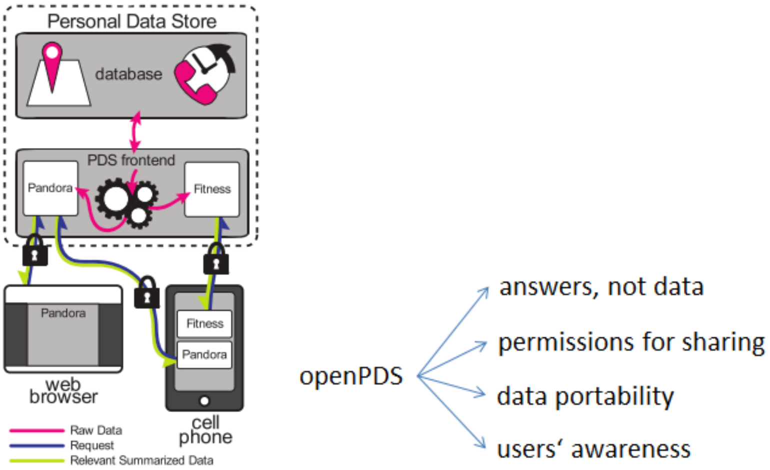


Figure 4.1: *(Left)*: *openPDS* systemâĂŹs architecture. *(Right)*: pillars of Personal Data Store.

solutions for personal data management, are preventing data from being shared and reconciled under the control of the individual [90]. This direction has been studied recently and in the following we report a detailed overview of the state of the art.

One of the most important examples is the *openPDS* framework [91, 90]. "Open", in openPDS, suggests its open source nature and PDS stands for Personal Data Store. The openPDS answers questions from external applications or services with answers that the user allows for being shared. Computations on user data are performed in the safe environment of the PDS, under the complete control of the user. The idea is that only the relevant summarized data for providing functionalities to the applications should leave the boundaries of the user's Personal Data Stores. In summary, openPDS, is a personal metadata management framework that allows individuals to collect, store, and give fine grained access to their metadata to third parties. openPDS is oriented to the protection of the metadata shared and on the privacy of the data contained in the system. For example, rather than exporting raw GPS data, it could be sufficient for an application to know if you are active or which geographic zone you are in. In Figure 4.1 *(left)*, the system is still exposing personal data of the user, but it is constrained to be what the app strictly needs to know, rather than the raw data objects the user generates.

Over time, user's openPDS would be filled with information collected by her phone, but also information about her tastes or her contacts, as well as a stream of other sensors of information that the user accumulates in her daily life. The user would have full control over these data, and could see exactly what data are relative to her phone, or other sensors and services, gathered about her over time. The PDS has access to historical data of the user, therefore every service chosen by the user can use all the possible information regarding the user, and it allows the more innovative companies to provide better data-powered services.

Users must know what data are captured and, on top of that, they must be able to control how they are shared and enabled to trust how other users employ them. Indeed, a privacy-preserving PDS like openPDS allows for greater data portability, as the user can seamlessly interface new services with her openPDS, and will not lose ownership or control of her personal data. Since the shared information is certified answers instead of raw data, the distribution and sharing of information could be possible. Finally, the user can decide whether such services provide enough value compared to the amount of data it asks for.

The user can reason on questions like "Is it finding out the name of a song worth enough to me to give away my location?". The openPDS can help the user in making the best decision for herself, and, in this way, she can acquire more awareness. The stated core principles of openPDS are summarized in Fig. 4.1 *(right)*.

The same principles are adopted in [207, 208], where the authors introduce the *Bank of Individuals' Data (BID)* as the provider of PDS features. The BID provides a secure and trusted space, i.e., a vault, where a person can put her personal data, and can operate on them by creating, lending or even selling her data. Like banks, the BID can act as catalysts of new opportunities which bring economic or social advantages to all the actors of the ecosystem. The proposed framework is organized in five layers, each one provides specific functions. The *data space for managing digital footprint layer* is responsible for data storage, automatic collection of personal data from different sources, enrichment, e.g., with meta-data, search/retrieval and visualization. The *trusted environment for personal applications layer* handles trusted environment, e.g., a sandbox, for the deployment, management and execution of "personal" applications. The *controlled sharing of personal data layer* enables a user-controlled sharing of data in a person's "digital footprint"; this defines temporary or permanent relations between an individual and a third-party. The *personal data negotiation layer* offers features to manage negotiation on personal data disclosure: these enable individuals to negotiate the conditions on the disclosure of their data to third parties, to get some economic or social advantages. Finally, the *data aggregation and analytics layer* provides features to aggregate personal data. These functions are in charge of analyzing and processing data provided by groups of individuals: *(i)* identifying (homogeneous) groups of people; *(ii)* creating aggregations of data disclosed by each of the group members; *(iii)* providing the aggregations to third-parties, and *(iv)* improving the quality of datasets by reducing statistical noisy effects.

Another recent work is [295] where the authors present *My Data Store*, a Personal Data Store tool allowing people to control and share their personal data. *My Data Store* enables to control and share data organized as a set of web-based services. The main services released are the following: *collection*: users can determine which data automatically collect and store; *sharing*: users can choose whether to disclose or not their data and in which detail, e.g. anonymously; *deletion*: users can delete single records or all data collected in a specific region and time interval, and *views* with different levels of aggregation: *(i)* individual increasing user's consciousness; *(ii)* social using data shared by. In [294], My Data Store has been integrated into a framework that permits the development of trusted and transparent services and apps whose behavior can be controlled by the user, allowing the growth of an eco-system of personal data-based services.

The proposal described in [1] is that each user can select which applications have to be run on which data, facilitating in this way diversified services on a personal server. In such a way, the personal server would contain all the user's favorite applications and all the user's data that are currently distributed, fragmented, and isolated. In theory, the user pays for the server, so the server does what the user wants it to do. The server resides in the cloud so it can be reached from anywhere. The user chooses the application code to deploy on the server and the server software is possibly open source. These guidelines are the proposal to achieve the aforementioned high levels of transparency.

Another aspect of PDS considered in the literature is how the information and the knowledge collected should be organized. Indeed, Personal Data Store systems require powerful and versatile tools able to represent a highly heterogeneous mix of data such as relational data, transactional data, mobility data, file content, folder hierarchies, emails and email attachments, data streams, log sessions, etc. [2]. The *iDM* data model for personal information management presented in [95] is a data organization model able to represent unstructured, semi-structured and structured data inside a single model.

Moreover, there are different works in the literature related to the condition of data disclosure and sharing. In [266], the authors study the problem of budgeted recruitment of participants in community sensing, which is a paradigm for creating efficient and cost-effective sensing applications by using the data of large populations of sensors (e.g., earthquake detection from the accelerometer data collected by smartphones, or real-time traffic maps from velocity data from GPS devices). The basic assumption is that each user establishes both a cost for sharing her data and specific privacy constraints. Authors propose a greedy algorithm to select a set of participants from a large population, optimizing the trade-off between total budget and benefits. The authors of [223] show that people's willingness to share information depends greatly on the type of information being shared, with whom the information is shared, and how it is going to be used. In [57] is studied how users value their *Personally Identifiable Information (PII)* while browsing. The experiments demonstrate that users have different valuations, depending on the type and information content of private data. Higher valuations are chosen for offline PII, such as age and address, compared to browsing history. In [267], the authors develop online incentive-compatible and budget feasible mechanisms for procurement, assuming minimal information about the distribution of workers' true costs but using a utility function where each participant provides a unit value. The authors of [75] study a privacy game in mobile commerce, where users choose the degree of granularity at which to report their location; thus the service providers offer them monetary incentives under budget constraints. In [313], authors discuss approaches to capitalize private data assets. They propose a model of privacy data negotiation between buyers and sellers. This means that protection of privacy data is necessary only if there is a group driven to buy the information.

The study of systems based on incentives allows us to encourage users to give access to her own data, with the desired privacy level, to receive advantages if they disclose them to a company or a public authority. The user benefits can be of various kinds: discounts in supermarket in exchange for her purchase history, a free proactive car-pooling service if she shares her GPS traces, or she could actively participate in the estimation of the social well-being of the community by sharing information about her posts or social relationships. In a PDS each user should have the ability to attribute the right value to her own data.

In this thesis we totally embrace this philosophy, and we collocate the users' PDSs into an ecosystem which allows users to gather data from different sources, to transform the data, and to offer an interface through external services. However, it is worth to notice that the majority of the works in the literature focus their attention on the PDS architecture and on how to treat data sharing and privacy issues. On the other hand, rather than in privacy issues, in this thesis, we are focusing on how to extract a *Personal Data Model (PDM)* able to summarize and characterize the user behavior in the PDS in order to obtain an added value from the personal data through the application of data mining techniques. In our context we would like that a Personal Data Store could allow an individual not only the storage and management of private data, but also the automatic extraction of systematic behaviors and the providing of proactive suggestions on the basis of the user's profile. Moreover, we show how these model can be exploited by innovative services developed for the users part of this ecosystem of shared data.

Part II

Personal Data Analytics

Chapter 5

Personal Data Analytics

The challenges posed by the great amount of big data availability at personal level open a novel and interesting scenario of *Personal Data Analytics*. Most of the state-of-the-art analysis and methods are related to global studies of the whole system. The research of personal patterns and individual systematic behaviors is still unexplored but it is surely a promising direction. We present the *personal data context* in Section 5.1, then, in Section 5.2 we describe how *personal data mining* can be employed to extract personal data models able to capture behavioral patterns and summarize human behavior for a Personal Data Store (PDS). In Section 5.3 we illustrate the ecosystem where the users' PDSs can act generating a new level of collective awareness. Finally, in Section 5.4 we describe the impact this change of perspective could have on our society.

5.1 We All Need To Own and Use Our Own Data

Every year, each person leaves behind her more than 5 gigabytes of *digital breadcrumbs*, disseminated by disparate systems that we use for our daily activities, to travel, communicate, pay for goods, bills and food, banking, sport, searching the web, listening music, reading, playing, texting, writing, posting or tweeting, screening our health. Five gigabytes, without taking into account photos and videos, otherwise numbers would grow considerably. An avalanche of personal information that, in most cases, gets lost – like tears in the rain. Yet, only each one of us, individually, has the power to connect all this personal information. No Google or Facebook has a similar power today, and we should very carefully avoid this possibility. The fact that in the contemporary initial phase of a measurable society there are few large harvesters, or "latifundists", who store data on masses of people in large inaccessible repositories in an *organization-centric* model, does not mean that *centralization* is the only possible model, nor the most efficient and sustainable.

Nowadays, data and information belong to big organizations which employ *top-down* control over these data. For example, users produce personal data like Facebook posts, or GPS movements using Google Maps, or online shopping through Amazon, and these data are collected and obscurely employed by these companies for marketing or to produce services. According to [158], this is a *Legrand Star* model, i.e., a centralized network model, where users can not directly control and exploit their own personal data. Data owning and usage would require not a *bottom-up* system, but a *Baran Web* model, i.e., a *peer distributed approach*, a network of peers, both individual and companies, in which no single node has absolute control of everything but everyone controls thyself, and has only a partial vision of the surrounding peers. The first brick that must be placed in order to build this *Web* and to start a change of perspective, is the development of Personal Data Models which are sewn on each individual user in order to fit their subjective behaviors.

Data Mining applied to personal data, i.e., *Personal Data Mining* creates an invaluable opportunity for individuals to improve their *self-awareness*, and for enabling *personalized services*. However, nowadays users have a limited capability to exploit their personal data. This is why they require to own and use their data: they need to handle their own personal data. As already mentioned, these needs are leading to a change of perspective towards a *user-centric* model for personal data management. This vision is compatible with the data protection reform of EU, and is promoted by the World Economic Forum [161, 239, 250].

5.2 Making Sense of Own Personal Big Data

The unstoppable rise of smartphones joint with their increasing ability to collect individual information is creating a huge increment in the production of personal data. Personal information like visited locations, web-searches, purchases, phone calls and even music listening are collected and stored without any clear benefit for the user. Consequently, it is being defined the need for personal models to manage and exploit these large amounts of data. As detailed in Section 4, in the last years is taking place the idea of the *Personal Data Store (PDS)*: a personal, digital identity management service controlled by an individual where each user can choose at which level she wants to share her own data [90].

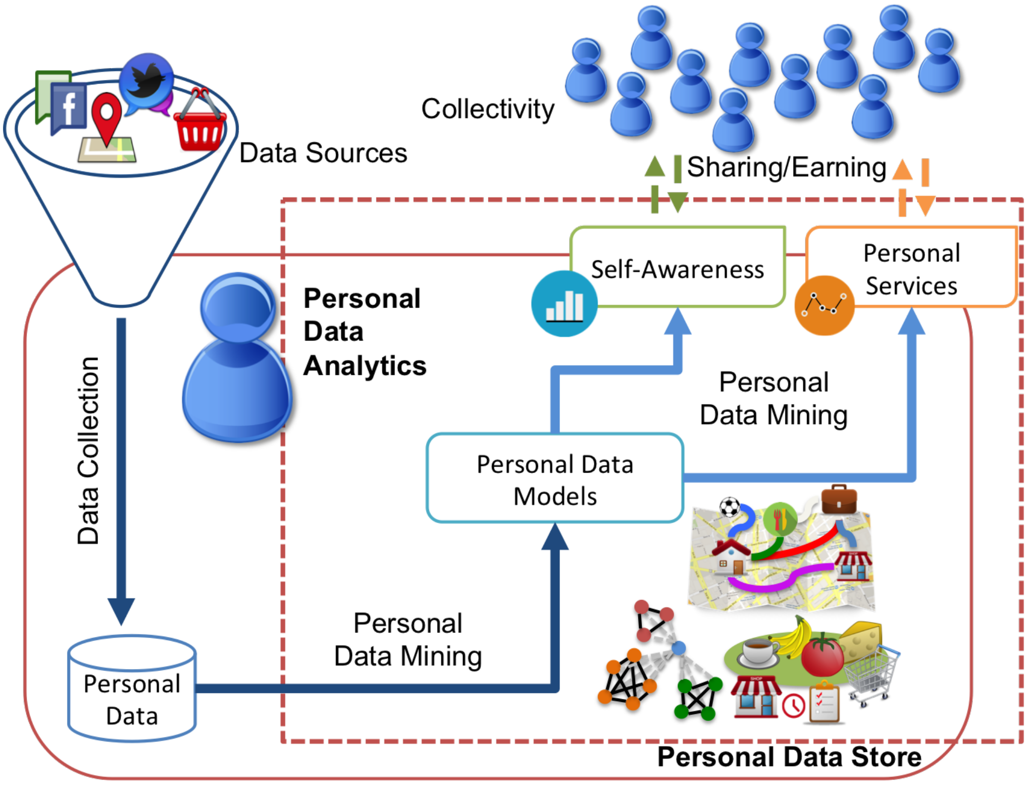


Figure 5.1: A Personal Data Store to collect and make sense of own personal data.

Our idea, illustrated in Fig. 5.1, is to introduce in such a service a *Personal Data Model (PDM)*, i.e., a user profile that is automatically extracted trough *Personal Data Mining.* Personal Data Mining refers to autofocus methods able to build subjective and auto-adaptive PDMs. These methods are able to automatically detect and extract the repetitive and valuable patterns delineating the user's systematic behaviors. The PDM can be exploited *(i)* to improve the user *self-awareness* thanks to the personal patterns they unveil, and *(ii)* to empower *personalized services* by providing proactive predictions and suggestions on the basis of the user's profile. As highlighted by the dotted rectangle in Fig. 5.1, with *Personal Data Analytics* we indicate the Personal Data Mining processes extracting the user profile models, and providing self-awareness and personalized services.

More formally, we define an abstract data type to apply Personal Data Analytics as:

Definition 1 (Individual Data Event)**.** *Given a user u, an* individual data event x *represents any event or action performed by u with a specific data type.*

Examples are a movement between two locations, a purchase of a set of items, or the listening of a song. Therefore, x can be a simple value like the amount spent in a month, or can be structured and formed by various components like a sequence of GPS points. The collection of individual data events forms the *Personal Data* or *individual history*:

Definition 2 (Individual History)**.** *Given a user u, her* individual history $H_u = \{x_0, \dots, x_{n-1}\}$ *is the set of the individual data event performed by u w.r.t. a certain data type.*

Given the individual history of a user u we can extract her *Personal Data Model*:

Definition 3 (Personal Data Model)**.** *Given a user u and her* individual data history H_u*, we define* $P_u = extract(H_u)$ *as the* personal data model *extracted from* H_u*, where the function* $extract(\cdot)$ *represents a* personal data mining *algorithm or data analysis process.*

Then, the Personal Data Model P_u can be exploited to improve the user self-awareness and for any kind of personalized service, e.g. $predict(P_u, x)$, $recommend(P_u, x)$, where x is the current event on which the prediction or recommendation are based on.

In our vision, the Personal Data Store allows an individual not only the data storage and management, but also Personal Data Analytics that enables the user to make sense of her own personal data and to exploit it [137]. We report in Fig. 5.1 the overall Personal Data Analytics approach. The individual data flow into the *Personal Data Store* and are collected and stored according to one of the possible technique described in the PDS literature [90, 295]. From Personal Data we can easily extract simple personal statistics like the average money spent per purchase, the distribution of the distance traveled, the most listened musical genre, etc. They can be useful to give a rough description of the user. However, they are generally not enough detailed to represent and summarize the user behavior. Therefore, the *Personal Data Models* forming the user profile are extracted from Personal Data through *Personal Data Mining* methods, e.g. clustering algorithms. The results of these techniques are the patterns and indicators forming the user profile. Along the analysis of the continuous digital breadcrumbs, the PDS must consider that it does not exist a unique and constant model describing human behaviors. Indeed, our behaviors will be never "in equilibrium" because we constantly move, we buy new things, we interact with our friends, we listen to music, etc., generating in this way a non-interruptible flow of personal data [20]. Therefore, the PDM must be *dynamic* and *adaptable* to continuous changes and updates. The user profile described by the PDM can be used both to improve the user *self-awareness* and for *personalized services* yet adopting Personal Data Mining methods. Self-awareness can be realized for example through a personal dashboard where the user can navigate and understand her models and patterns. On the other hand, examples of personalized services can be *recommendation systems* or *predictors* of future actions.

5.3 The Personal Data Ecosystem

Such a PDS is not just a place where all our personal data can be stored, but through the PDM extracted with Personal Data Mining techniques it offers us an augmented image of ourselves, our image reflected in a *digital mirror*. This mechanism can help us in understanding our behavioral, social, mobile, shopping patterns or, at least, how these emerge from the digital breadcrumbs we leave behind, and in providing us enhanced self-awareness. However, passive personal data collection and knowledge mining need to be balanced with *participation*, based on a much greater awareness of the value of own personal data for each one of us and the communities that we inhabit, at all scales.

The Personal Data Analytics approach proposed, provide us the opportunity to compare our individual patterns with the collective patterns of the communities we belong to, provided we have a way to interact and collaborate with other peers, individuals and institutions that are, in turn, equipped with their PDS's and connected to each other in the social network. In Fig. 5.1 top right is shown how, in order to provide and obtain improved *self-awareness* and *personalized services*, a user can share information and, at the same time, earn knowledge, by communicating with the *collectivity*.

This enables a *Personal Data Ecosystem (PDE)* illustrated in Fig. 5.2. A PDE is a distributed network of peers, which can be both individual users, and public or private institutions and companies, each one with their own type of PDS and PDM. The *Data-Service Provider* in Fig. 5.2 can represent a public or private institution or company that, through a distributed data platform, provides to the users a set of services and/or a safe storage space for the data produced by the service usage and for running the PDS with all the features of the Personal Data Analytics approach. Each peer is enabled to share (part of) her knowledge contained in her individual profile P_u with her trusted neighbors, and the benefits she obtains in return consists in a form of *collective awareness*:

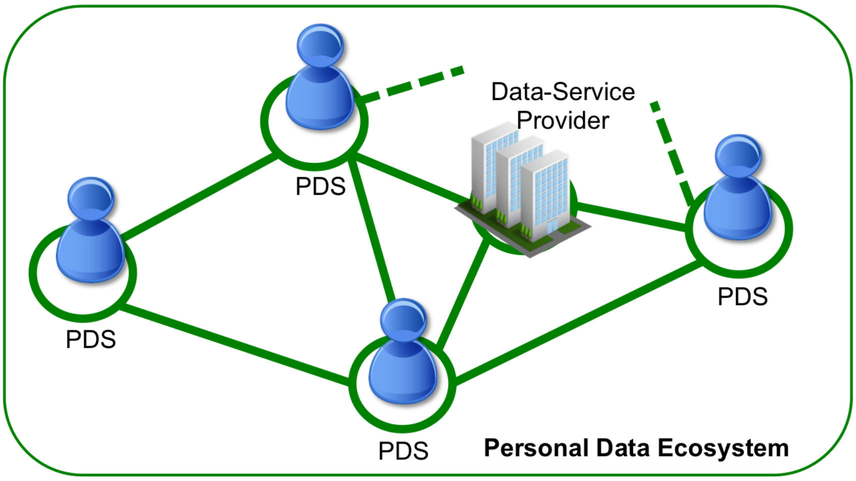


Figure 5.2: A Personal Data Ecosystem as a decentralized peer-to-peer network.

Definition 4 (Collective Data Model). *Given a set of users U, if they allow sharing their Personal Data Models P_u $\forall u \in U$, then we can define a* collective data model $P_C = \bigcup_{u \in U} P_u$ *obtained as combination of the Personal Data Models shared.*

This combination can be a simple union or a more complex approach involving Personal Data Mining aggregation techniques. Note that the collective data model P_C can be calculated independently from the underlying architecture. An appropriate choice for a totally distributed environment like the PDE would be a distributed protocol for communicating among the peers where a user u can start the process, sending her information P_u to some neighbors they trust, then combine P_u with their own PDMs according to the purpose of the data sharing, and send the updated information back to u and to their trusted neighbors, and so on and so forth. At a certain point, each peer would receive a collective data model P_C sufficiently enriched with the knowledge of the network to be exploited both to compare her behavior with those of the collectivity, and to improve some personalized services exploiting the "wisdom of the crowd", e.g. $predict(P_u, P_C, x)$. How to determine when exactly a service would have gathered sufficient information in order to exploit in a reliable way the wisdom of the crowd remains an open question with respect to this thesis. However, in the following we show the empirical evidence that the collaboration among users and the sharing of knowledge can markedly improve the analyzed services.

Therefore, the PDE generates an innovative form of *collective awareness*, characterized by a self-reinforcing loop where *(i)* superior individual knowledge is created by comparison with collective knowledge, enhancing individual ability to better align own goals with common interest, and *(ii)* superior collective knowledge is created through the active participation of individuals in a decentralized system, i.e., without centralization of unnecessarily large amounts of personal information. As an example, imagine a PDE containing PDMs for analyzing personal mobility data which contains the routine traveling patterns [137]. These, are a key element to show to the user her typical habits (self-awareness), and to help her to understand how her mobility behavior is positioned in comparison with the behavior of the mass, of the friends, of similar users and of users in the neighborhood, the so called *"Where I Am"* service. Moreover, the routine traveling patterns could be used by a proactive carpooling service to provide possible matching between drivers and passengers (collective awareness). Decentralized services like these would empower the individual to better understand her own role within the collectivity (society) by facilitating a better alignment of *self-interest*, e.g. by minimizing the travel time, and to promote a *collective interest* for social good, like minimizing the overall traffic congestion and pollution.

Once again, the proposed approach is in line with the *peer progressive* idea detailed in [158]: a decentralized network where news, ideas, money, and knowledge come from the periphery instead of from the center. In [143], Helbing compares the concept of "wise king" (centralized system) against the Adam Smith's "invisible hand" regulating a decentralized self-organizing system. Also in this work is shown how, thanks to the continuous flow of data and information, nowadays the self-organizing system can beat the centralized one. Furthermore, the PDE idea outlines the project described in [119] where is advocated the vision of *Nervousnet*: a globally distributed, self-organizing, techno-social system for answering analytical questions about the status of world-wide society, based on social sensing, social mining and the idea of trust networks and privacy-aware social mining.

Therefore, the user-centric data management alone is necessary but not sufficient to realize the Personal Data Analytics approach: data analytics both at individual and collective level is the key to success. If we help people to understand the importance of personal data in our daily lives to simplify, be more efficient and diversify, then we may boost the emergence of a totally different ecosystem, compared with the current mainstream, where information can flow without the need to concentrate data in large centralized repositories. An ecosystem where each one of us, rather than giving up own data by agreeing to some obscure disclaimer, decides whether to answer or not to questions asked by other people or entities, based on one's own interest in participating and on the trust we have on the interlocutors. An ecosystem that, seen from the outside, looks like a large database we can ask queries to, but in reality is a peer-to-peer network of people with their PDSs, as depicted in Fig. 5.2, who can choose to cooperate to reply to queries. Scientists like Philippe Pucheral of INRIA, Alex Pentland of MIT and Dirk Helbing of ETH Zurich are developing early ideas of architectures concentrating on the "collecting" part of the personal data (Data Collection and Personal Data in Figure 5.1).

In this thesis we develop the Personal Data Analytics needed for the PDE (dotted rectangle in Fig. 5.1), a problem hardly tackled so far, which is crucial to boost users' self- and collective awareness, together with the capability to obtain higher-quality services.

In the user-centric vision, the user acquires an active role and full control on the lifecycle of own personal data. This includes enabling individuals to control their data and the knowledge that can be extracted from it, and granting them the right to dispose or distribute data, with the desired privacy level, to get the desired services. This sheds a new light on data protection: most of the existing solutions consider an architecture where central sites make the data private before releasing them. In the new model, the privacy-based transformations must be performed before data leave the user. This should encourage the voluntary participation of users, reducing the skepticism that often leads people to not access the benefits of their own data. In this thesis we assume *Privacy-by-Design* methodologies [58] can be applied to prevent privacy attacks and maintain data anonymity in the PDE. They are proactive approaches where privacy is taken into account throughout the data mining process. However, we leave as future works a deep investigation of techniques to perform data and models sharing respecting personal privacy constraints.

What is a possible way to develop the PDE? Which can be a transparent technology able to regulate the exchange of data and patterns in a safe and ethical context? [264] provides a framework describing *blockchain* as a "fifth horizon of networked innovation". Blockchain is a peer-to-peer network that broadcasts data to all nodes on the network. It represents a technology innovation that enables transparent interactions of parties on a more trusted and secure network which distributes access to data. Although the technical components have been in existence for decades, blockchain is a novel, resilient, and ubiquitous approach to data, transaction analytics, and networks. It holds the potential to address inefficiencies, reduce cost, unlock capital, improve trust in societal fabric, and open new business models. Blockchain has generated extensive interest and enthusiasm in financial markets because trust and confidence in the promise to meet the obligations are the cornerstones of any financial transaction. Blockchain is the technology behind Bitcoin. Bitcoin [217] is a decentralized electronic fiat currency implemented using cryptography and peer-to-peer technology, i.e., Blockchain [174]. To prevent double spending, Bitcoin players engage in a peer-to-peer protocol that implements a distributed timestamp service providing a fully-serialized log of every Bitcoin transaction ever made. Transactions, i.e., data exchanges, are organized in the log into blocks, which contain a sequence number, a timestamp, the cryptographic hash of the previous block, some metadata, a nonce, and a set of valid Bitcoin transactions. The blocks form a hash chain: each new block contains the cryptographic hash of its predecessor, allowing *anyone* to verify that no preceding block has been modified. The block chain contains backward links but not forward links (a block cannot link forward to a future block that has not yet been created) so there is a unique path backward from each block to the beginning of the log (the genesis block) but the forward path from a block might not be unique. Thus the log has the form of a tree whose branches fork as it grows. Therefore, Blockchain's highly resilient architecture and distributed nature make it an interesting platform to deliver in the society of the PDE. We reported this technology because we believe that can be a turning point in the real development of a PDE. However, as for privacy issues, we leave this aspect for future work, while in this thesis we focus on Personal Data Analytics algorithms, models, and services.

In summary, the objective of this thesis is the design of Personal Data Mining methods and models to augment a PDS with personal patterns, as well as the definition of services which exploit both the individual knowledge, and a collective knowledge emerging from the cooperation of the users in the PDE. In the following, we show how we apply *Personal Data Analytics* through the definition of algorithms, models, and services.

5.4 Potentialities and Socio-Economic Impact

Sir Tim Berners-Lee, in his keynote in Rome on 13 January 2016, set the goal to: "make people owners of their own data and free to decide if, when and how to share them". This reasoning is at the basis of our thesis, which adds to this goal: "make people *aware* of the value of their own data for themselves and for the communities they belong to".

The PDE proposed has the potential to support the development of a new generation of user-centric, data-driven services that empower people in their interaction with service providers at all scales. The adoption of a PDE technology would rebalance today's information asymmetry between users and service providers, providing consumers/citizens with stronger power when negotiating with businesses/institutions. It would also support the development of so-called "sharing economy", "peer networks" and "liquid democracy" in all sectors, because a clearer perception of own behavioral profiles would put users in a better position to find peers with similar needs and interests, that may want to share and collaborate on common goals. Examples range from sharing commuting rides, to supporting policy options, to funding projects, to negotiating collective deals with businesses.

Personal Data Analytics in PDE has the potential to become an enabler for the so-called "Collective Awareness Platforms for Sustainability and Social Innovation" (CAPS), at the center of an initiative of the EU within the H2020 program, aimed at offering collaborative solutions to complex social sustainability problems based on networks of people and ideas. CAPS are expected to support self-organizing processes to share knowledge, to promote changes in lifestyle patterns, and to facilitate participatory processes. In the vision of the European Commission, as well as an increasing number of scientists in various disciplines, CAPS platforms may have concrete impacts in emerging socio-economic domain.

The ultimate goal is to foster a reinforcement spiral between collective awareness of communities at various scales, from local to global, and individual self-awareness of citizens obtained through enhanced management and understanding of own personal information; such spiral is expected to generate higher levels of both collective and self-awareness, providing a favorable techno-social ecosystem for social innovation and self-organization.

From a scientific/technological perspective, our approach combining Personal Data Mining within a decentralized peer-to-peer trust-based network has the potential to foster a new generation of techno-social models for distributed data management, analysis, and data-driven computing. In particular, it has the potential of inspiring a new generation of analytical methods capable of producing statistical outputs of high quality while respecting peers' privacy and trust choices and minimizing the flow of personal information.

Finally, the successful deployment of a PDE technology across the population would enormously facilitate programs like United Nations' Global Pulse or similar innovation initiatives targeted on harnessing big data safely and responsibly as a public good for monitoring the health status of our society, for sustainable development and humanitarian action. A running PDE would enhance the participation of people in producing information by (automatically) responding through their PDS to continuous surveys by trusted institutions, possibly with secure, anonymity-preserving interaction protocols (e.g. Blockchain). Also, the PDE technology would enhance the possibility for citizens to become direct users of the produced statistics and socio-economic-health indicators, using the visualization tools offered by the PDE for comparing individual and collective statistics.

Chapter 6

Proxies of Human Behavior

The term “Big Data" generally refers to any collection of data so large and complex that it becomes difficult to process using traditional data tools. We are under the *Big Data microscope*: as biologists observe micro-organisms under their microscopes, we can observe our personal actions under the powerful lenses of *Big Data*. We are in the Big Data era and almost everything we do nowadays requires the use of some digital device: from communications to travels, every human action is digitalized in some form. This digitalization permits to minutely describe each individual through the personal patterns forming her PDM.

As proxies of human life, in this thesis we focus on four types of data which are generated by the digitalization of some events or actions.

- *Mobility data*: car movements are stored in form of GPS points and trajectories.
- *Retail Market data*: retail purchases are stored in form of shopping sessions.
- *Complex Network data*: social opinions and status are stored in form of tweets.
- *Music listening data*: web listening sessions are stored in form of music listenings.

Table 6.1 summarizes the characteristics of the datasets used for the analysis described in this thesis. In the following, we provide some details for each dataset.

6.1 Human Mobility Data

The *Global Positioning System (GPS)* is a satellite navigation system that utilizes more than two dozen satellites. It broadcasts precise timing signals by radio to GPS receivers, allowing them to accurately determine their spatio-temporal location (longitude, latitude, timestamp). A GPS receiver calculates its position by precisely timing the signals sent by

Dataset	Type	Users	Events	Period	Area
Octo	GPS traces	150,000	9.8 M	1 month	Central Italy
Coop	transactions	1,600,000	300.0 M	7 years	Central Italy
Twitter	tweets & friendship	130,000	3.8 M	3 months	Rome & San Francisco
LastFM	listenings	30,000	6.0 M	1 year	UK

Table 6.1: Information about the datasets used in the thesis.

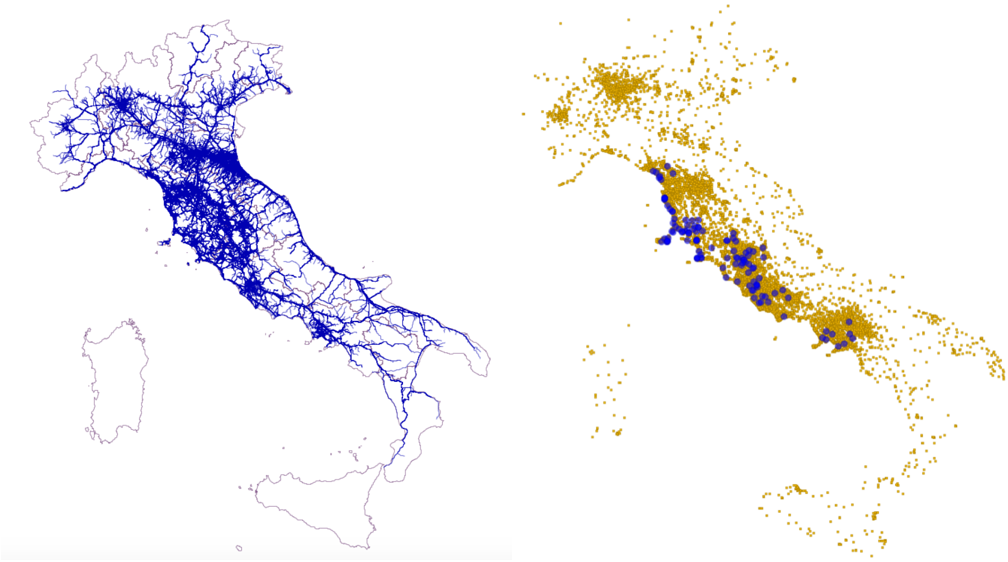


Figure 6.1: *(Left)* Octo dataset: GPS trajectories passed through central Italy in May 2011. *(Right)* Coop dataset: geographical distribution of shops (blue) and customers (yellow).

GPS satellites high above the Earth. Each satellite continually transmits messages which specify the precise positioning information, and the time the massage was transmitted. The receiver computes the distance to each satellite by determining the transit time of each message it receives. These distances along with the satellites' locations are used to compute the position of the receiver, in form of latitude, longitude and other information like elevation, direction, and speed. GPS-enabled devices provide us with all the required information for trajectory tracking, giving access to accurate, time-stamped locations of each tracked moving point. Nowadays GPS receivers are embedded in many devices we use every day like smartphones and vehicles, allowing to easily track human mobility.

In this thesis we use a massive real-life GPS dataset, the *Octo dataset*, obtained from tens of thousands private vehicles with on-board GPS receivers. The owners of these cars are subscribers of a car insurance contract, under which the tracked trajectories of each vehicle are periodically sent to a central server for antifraud and anti-theft purposes. This dataset has been donated for research purposes by *Octo Telematics Italia S.r.l*[1]. The market penetration of this service is variable on the territory, but in general covers around 3% of the total registered vehicles. The Octo dataset stores information of approximately 9.8 Million different car travels from 150,000 cars tracked during May 2011 in a geographical area corresponding to Tuscany, central Italy (see Figure 6.1 *(left)*).

The GPS device automatically turns on when the car starts, and the sequence of GPS points that the device transmits every 30 seconds to the server forms the historical movement of a vehicle. When the vehicle stops no points are logged nor sent. By employing an advanced version of [201], we exploit these stops to split the historical movement into several sub-movements named *trajectories*, that correspond to the travels performed by the vehicles. Clearly, the vehicle may have stops of different duration, corresponding to different activities. To ignore small stops like gas stations, traffic lights, bring and get activities and so on, we choose a stop duration threshold of 20 minutes: if the time interval between two consecutive GPS points is longer than 20 minutes, the first observation is considered

[1] http://www.octotelematics.it/

as the end of a trip and the second observation is considered as the start of another trip. We also perform the extraction of the trips by using different stop duration thresholds $\{5, 10, 15, 20, 30, 40\}$ minutes, without finding significant differences in the sample of short trips and in the statistical analysis we present in the current thesis.

6.2 Retail Market Data

With respect to retail market data, i.e., transactional data, the dataset we employ is the *Coop dataset*. *UniCoop Tirreno*[2] is one of the largest Italian retail distribution company. The market chain serves several million customers covering an extensive part of the Italian territory. The 138 stores of the company sell about 347,000 different items. In particular, the stores of the company mainly cover the west coast of central Italy (see Fig. 6.1 *(right)*). The shop distribution is not homogeneous: shops are located in a few Italian regions and therefore, the coverage of these regions is much more significant while customers from other regions usually shop only during vacation periods in these regions. The chain operates three different tiers of shops according to their size: *Iper* shops are the largest, the Italian equivalent of a US mall; *Super* are the middle level, a large supermarket; and *Small* is the smallest shop type, whose size is comparable to a dollar store. The dataset contains retail market data in a time window spanning from January 1st, 2007 to June, 30th 2014. The active and recognizable customers in that interval are about 1,600,000. A customer is active if she has purchased something during the data time window, while she is recognizable if the purchase has been made using a membership card. Through the card, customers can get a discount. The company is able to tie each shopping session to the card. In particular, for each shopping session, or basket, the company knows:

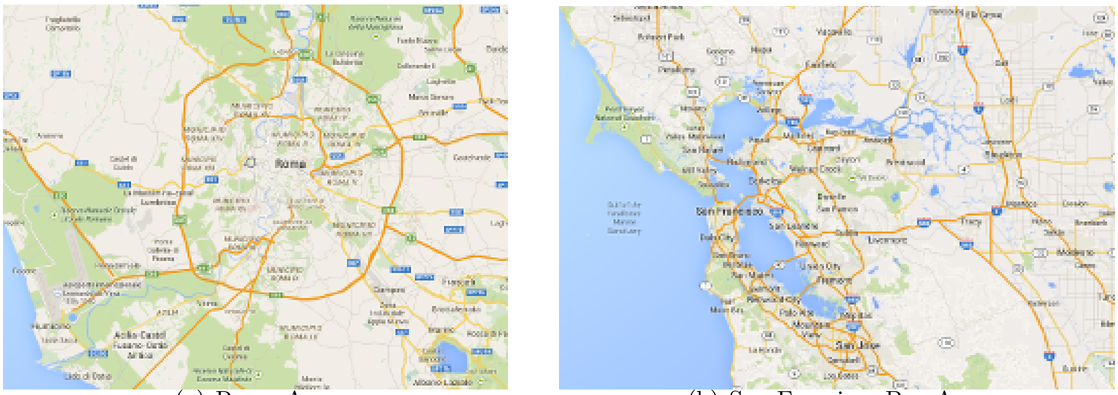
- which customer made the purchase;
- all single items composing the basket;
- the time and the day of the shopping session;
- in which shop the transaction happened.

6.3 Social-Network Data

Twitter is an online social networking service that enables users to send and read short 140-character messages called *tweets*[3]. Registered users can read and post tweets, but those who are unregistered can only read them. Users may subscribe to other users' tweet: this is known as “following" and subscribers are known as “followers". Thus, Twitter generates a social network of relationships among users sharing comments and opinions. Moreover, besides the content of each tweet, some other meta-data are available like the tiemstamp of when the tweet was published and a geo-tag indicating the latitude and longitude with GPS coordinates. This information is fundamental to analyze areas and periods of interest.

[2] https://www.unicooptirreno.it/

[3] https://twitter.com/



(a) Rome Area

(b) San Francisco Bay Area

Figure 6.2: Twitter data: geographical areas analyzed. GPS coordinates bounding box: Rome (12.234498, 41.655642, 12.85576, 42.141028), San Francisco (-122.667, 36.8378, -121.2949, 38.0771)

We used the Twitter's Streaming API[4] to obtain the *Twitter dataset* which consists in two large datasets of geotagged tweets. We queried the API using two bounding boxes on the area of Rome (Fig. 6.2(a)), and the bay of San Francisco, hereafter referred to as San Francisco (Fig. 6.2(b)), for 50 days from the beginning of October 2014. As a result, we collected about 558,000 geo-tagged tweets from 17,600 different users in Rome, and about 3,286,000 geo-tagged tweets from 113,000 different users in San Francisco.

6.4 Music Listening Data

Last.Fm is an online social network platform[5], where people can share their own music tastes and discover new artists and genres basing on what they, or their friends, like. Each user produces data about her own listening. Each listening is characterized by the song, artist, album, genre and the timestamp in which the listening took place. For each song, a user can express her preference and attach tags to each song, describing the subjective genre of the musical piece. As in other online social networks like Facebook, in Last.FM users can add friends search for "neighbors", i.e., other users with similar musical tastes. Then a user can see, in her homepage, her friends' activities.

Using Last.Fm APIs[6], we extracted the *LastFM dataset* made of 30,000 users resident in the UK. We started from an initial seed and we explored the Last.FM network with a breadth-first approach, up until the fifth degree of separation from the initial seed of users. Through this procedure, for each user we retrieved *(i)* friendships connections, and *(ii)* the information characterizing the last 200 listenings. In particular, a listening event contains: the timestamp indicating when the listening occurred, the title of the song, the artist who sings the song, the album the song belongs to, and the genre associated with the artist.

[4] https://dev.twitter.com/docs/streaming-apis

[5] http://www.last.fm/

[6] http://www.last.fm/api/, retrieval date 2016-04-04

Part III

Algorithms and Models for Personal Data Analytics

Chapter 7

Autofocus Algorithms for Personal Data Mining

The building blocks necessary to obtain a sound and stable user profile are algorithms able to extract and summarize the user's behaviors. *Autofocus* and *efficient* clustering algorithms are the best methods to capture the Personal Data Model describing the systematic patterns for a given user, that is the *extract* function of Personal Data Analytics.

As highlighted in Section 2.3, the vast majority of clustering algorithms present in the literature suffer from various drawbacks when repeatedly applied to different personal datasets: either they require a parameter tuning process that is not automatic, or they require an extremely heavy automated process that not scale to large user databases. As a consequence, the repeated application for each users' dataset of any of the existing procedures for the purpose of finding personal clusters is not feasible in presence of a large population of users like those part of the PDE. Also, a fixed parameter setting for all users could lead to misleading patterns since each individual might show specific features that require a treatment different from the others. Moreover, generic clustering algorithms are often focused on specific optimization criteria, that is not always the best choice and therefore the resulting clusters are not a good summary of user's behaviors even though they are optimal with respect to the optimization criteria. In addition, in most cases, the resulting clusters are affected by some assumptions the algorithm makes about the data.

According to the Personal Data Analytics approach, in this thesis we propose two autofocus and efficient clustering algorithms for mobility data and transactional data respectively. They overcome the issues and weaknesses faced by the existing clustering algorithms since they are specifically designed for personal data mining and for the extraction of users' profiles and individual patterns aimed at forming Personal Data Models.

7.1 Clustering Algorithm for Personal Location Detection

One of the key tasks in mobility data analysis for many applications related to GPS mobility data is *users' locations detection.* Its objective is to identify the users' *personal locations*, i.e., the areas where each user performs her activities, based on the analysis of the locations (essentially, GPS points) where she stopped, here called *stop observations.* Examples of locations are home, the workplace, a supermarket, a gym, a fuel-station, etc. Correctly discovering such personal locations is therefore a very important problem.

In literature this problem is typically addressed using generic clustering algorithms (see Section 2.3.1) which are able to group the user's stop observations by means of some distance function, thus yielding clusters that will be interpreted as user's locations. Such algorithms suffer from various drawbacks. First, some of them are focused on specific optimization criteria, such as compactness maximization or density connectivity, that not always correspond perfectly to the notion of locations, and therefore the results, though optimal w.r.t. its own criteria, are not good locations. Second, in some cases the algorithms need parameters that are not easy to guess. Indeed, an experienced analyst or some expensive self-tuning procedure might be needed to select accurately the parameters. Also, in most cases such parameters are fixed for all users, while each individual might show specific features that require a treatment different from the others. Finally, some algorithms do not scale well enough to be used on very large datasets. This is especially true for solutions that include a parameter tuning phase requiring multiple runs of the basic algorithm. To overcome these drawbacks we designed *TOSCA* [136], a *TwO-Steps Clustering Algorithm* explicitly shaped for user's locations detection. TOSCA is a robust, efficient, statistically well-founded and parameter-free personal location detection process. The two steps are realized with the combination of clustering methods and statistical analysis that enable TOSCA to produce high quality clusters with a low computational cost.

7.1.1 Problem Definition

In this section we formally define the *Locations Detection Problem (LDP).* The data event treated in the LDP are the *users' GPS stop observations*:

Definition 5 (User's Stop Observations)**.** *Given a user u, the set of her stop observations is defined as* $S = \{s_1, s_2, \dots s_n\}$*, where each* $s_i = (x_i, y_i)$ *represents GPS coordinates expressed as longitude and latitude.*

The locations associated with a set of stop observations basically group the latter into partitions that define the places (or areas) they cover. E.g. a location can be home and the observations belonging to it are all the parking lots used by the user in the nearby.

Definition 6 (Location set)**.** *Given a set of observations* S*, a* location set $\mathcal{L}$ *for* S *is a partitioning of* S *into disjoint sets:* $\forall l \in \mathcal{L} : \ l \subset S$, $\bigcup_{l \in \mathcal{L}} l{=}S$ *and* $l, l' \in \mathcal{L} \wedge l{\neq}l' \Rightarrow l{\cap}l'{=}\emptyset$.

The locations can be either provided as input, to be considered as ground truth, or they can be inferred (detected) directly from the stop observations through algorithms:

Definition 7 (Real and Detected Locations)**.** *Given a set of observations* S *we denote the* real locations *associated to* S *as* $\mathcal{L}_S = \{L_1, L_2, \dots L_k\}$*, and the locations inferred (detected) from* S *through any algorithm with* $\mathcal{D}_S = \{D_1, D_2, \dots D_k\}$.

The Locations Detection Problem, then, is simply defined as the task of inferring locations as close as possible to the real ones, across all the users:

Definition 8 (Locations Detection Problem)**.** *Given a set of users* U *and their observations* $\mathcal{S}_U{=}\{S_u\}_{u \in U}$*, the* Locations Detection Problem *(*LDP*) consists in producing for each* $S_u{\in}\mathcal{S}_U$ *a partition* $\mathcal{D}_S$ *that is similar to the corresponding real partition* $\mathcal{L}_S$.

We consider the most common case, where no real locations are known a priori, and therefore the locations detection problem requires to perform an unsupervised learning. In particular, it can be seen as a partitive clustering task, and is generally solved in literature through the adoption of a clustering algorithm. The result is a set of clusters of observations, which correspond to the detected locations, i.e., $\mathcal{D}_S$.

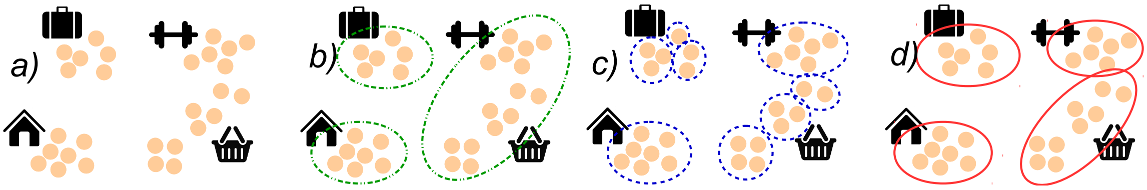


Figure 7.1: Example of personal location detection problem over four real locations *(a)*. Density-based clustering *(b)* and center-based clustering *(c)* are compared with the ground truth *(d)*.

7.1.2 Method

The algorithm we propose is called TOSCA, a two step clustering algorithm for location detection. The idea behind TOSCA comes from the need to detect the locations of the users in the PDE (independently from each other) in an efficient way without sacrificing the clustering quality and, most important, without any tuning phase for the parameters.

Motivations

Figure 7.1 (a) depicts the fictitious example of a user with four real locations (symbols represent the kind of activity performed there), and her several stop observations distributed around them. The desired partitioning of observations is shown in Figure 7.1 (d). Simply applying one of the basic clustering-based solutions present in the state-of-art can correctly find some locations, yet making mistakes on others. For instance, single linkage hierarchical clustering or density-based methods like DBSCAN [102] (Figure 7.1 (b)) might be able (through an appropriate parameter setting) to group correctly all observations relative to the work location, despite the fact that some observations are slightly peripheral; yet the leisure and shopping locations (right-hand of the figure) will probably put together, since there is no clear separation between the two groups of observations, and therefore they would be density-connected and put together. Similarly, a center-based clustering approach like X-Means [232] (Figure 7.1 (c)) might be able to isolate the leisure location, yet it might easily break down other locations into several small groups of observations, due to their not perfectly globular shape and not uniform density. TOSCA combines two algorithms from the two families of methods, each one compensating the shortcomings of the other one, as described in the following.

Rationale and General Schema of TOSCA

Through a large experimentation emerged that center-based clustering methods tend to correctly identify subgroups of observations that should belong to the same location. That is obtained by enforcing a strong compactness of the clusters. The side effect of such constraints is that the result usually splits real locations into several pieces that are connected with each other in a relatively loose way. On the other hand, single-linkage and density-based clustering methods are very good in spotting such loose connections, with the drawback of not distinguishing well those loose connections that are actually boundaries with other clusters, as in the case of the leisure vs. shopping locations in Figure 7.2. Our empirical study, yet, revealed that such boundaries become much sharper if we compare the (usually relatively small) clusters identified by center-based methods. In particular, the medoids of neighboring sub-groups that should belong to the same location tend to

Algorithm 1: TOSCA(S, *cut-criteria*)

```
Input  : $S$ - set of stop observations
Output: $\mathcal{D}_S$ - set of clusters
$\mathcal{D}^*_S \leftarrow$ x-means($S$);                         } First Step
$M \leftarrow$ get-medoids($\mathcal{D}^*_S$);                     }
$\mathfrak{D} \leftarrow$ single-linkage($M$);                       }
dist $\leftarrow$ select-cut($\mathfrak{D}$, cut-critera);             }
$\mathcal{D}_M \leftarrow$ cut-dendogram($\mathfrak{D}$, dist);        } Second Step
$\mathcal{D}_S \leftarrow$ aggregate-clusters($\mathcal{D}^*_S, \mathcal{D}_M$); }
return $\mathcal{D}_S$;                                           }
```

Algorithm 2: select-cut($\mathfrak{D}$, *cut-critera*)

```
Input  : $\mathfrak{D}$ - aggregation dendogram
Output: $d$ - cut distance
diffs $\leftarrow [d_1 - d_0, \ldots, d_{|\mathfrak{D}|-1} - d_{|\mathfrak{D}|-2}]$;
for $i \in [1, |\mathfrak{D}| - 1]$ do
    if $i > \rho$ then cut $\leftarrow$ cut-critera(diffs[1..$i-1$], diffs[$i$]);
    else cut $\leftarrow (d_i > 2 * d_{i-1})$;
    if cut then return $d_{i-1}$;
end
return 0;
```

be closer to each other than to those belonging to different clusters. That suggests us to start from the small clusters obtained by center-based methods and try aggregating them based on their medoids. This second step can be done through an iterative procedure like single linkage hierarchical clustering, stopping the aggregation when the effort of merging two clusters results to be much larger than the previous merges. Detecting the precise moment such effort becomes critical – i.e., determining the cut threshold – is a non-trivial issue, since it is generally impossible to fix a priori thresholds. The idea of TOSCA, then, is to interpret this as an outlier detection problem: the cut should be performed when the increasing of the distance between two consecutive merges grows abnormally with respect to previous iterations. For this task, then, we adopt a few statistical tests for anomaly detection. To summarize, the TOSCA approach is the combination of two steps:

1. extract (sub)clusters and corresponding medoids through center-based methods. In particular, the X-Means algorithm was selected through empirical evaluations;
2. cluster the medoids through a Single Linkage hierarchical algorithm. Stop the iterative clusters aggregation (or, equivalently, cut the *dendogram* resulting from a complete run of the algorithm) through a statistically-determined threshold on the increase of the distance between the clusters to be merged at each iteration.

TOSCA Algorithm

Algorithm 1 summarizes the process performed by TOSCA to detect the locations of a given set of points, highlighting the two logical steps involved. The input is the set of stop observations S of a user u and a *cut-criteria*. From the PDS point of view, the stop observations S are raw data contained in the user individual history H_u.

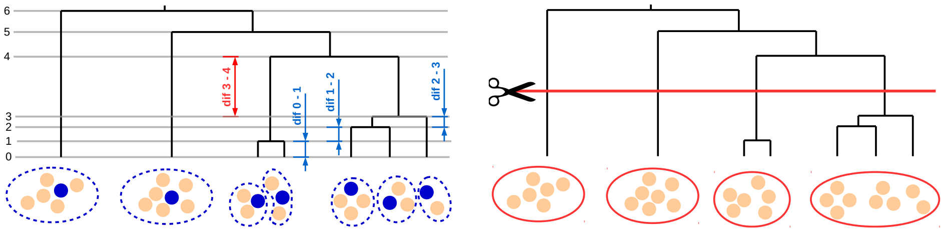


Figure 7.2: Orange points are the stop observations. Blue dotted circles correspond to X-Means clusters and the blue points to their medoids, which are then processed by Single Linkage. On the resulting dendogram we highlight the differences among distances. The red line is a possible cut.

Step One. Lines 1–2 perform the first clustering with X-Means on S, and the corresponding medoids M are extracted. *X-Means* [232] is a fast and statistically founded refined version of K-Means. Given an interval $[k_{min},k_{max}]$ it finds the set of clusters exploiting the *Bayesian Information Criterion (BIC)*. In the general case, the parameters can be simply set to k_{min}=2 and k_{max}=$|P|$-1, while smaller intervals can be used if knowledge about k is available, to reduce the search space and speed-up the computation. X-Means clusters and medoids are represented by the blue objects in Figure 7.2 (left).

Step Two. Lines 3–6 of Alg. 1 realizes the Single Linkage clustering on the set M of medoids. *Single Linkage* [265] is a standard agglomerative hierarchical clustering method that builds a hierarchy of clusters by progressively joining the two closest elements at each step. The distance between clusters is computed as the minimum distance between all pairs of elements taken from the two clusters compared, giving preference to those pairs that have close borders. The resulting hierarchy is called *dendrogram*, and it shows the sequence of cluster fusions and the distance at which each fusion took place (see Fig. 7.2 (left) for an example where the distance is represented by the height of the fusion point). The final clustering is generated by cutting the dendogram $\mathfrak{D}$ at distance (height) *dist* according to the *cut-criteria*. The dendogram can be mathematically represented by a list $\mathfrak{D}$=$[d_0, d_1 \dots d_{|M|-1}]$ of the distances computed by Single Linkage to aggregate the clusters, i.e., d_i is the distance at which two clusters are aggregated at iteration i. Note that, due to the functioning of Single Linkage, the d_i's grow monotonically, i.e., d_{i-1}≤d_i for all i=$1 \dots |M| - 1$. The *cut-criteria* selects the distance *dist* used to cut the dendogram. The clusters produced by Single Linkage are made by the set of medoids belonging to the same tree of the cut dendogram. The cutting algorithm, according to the cut-criteria, decides which set of medoids must belong to the same group. Finally, in *line* 6 the clusters generated by X-Means are aggregated according to the clusters produced by Single Linkage. Fig. 7.2 (right) shows an example where the dendogram is cut and all the observations associated to medoids in the same cluster are grouped together (red circles).

Cut Criteria. Alg. 2 shows how the value for cutting the dendogram $\mathfrak{D}$ is selected. This procedure analyzes the differences between the distances at which two clusters are aggregated by Single Linkage at each iteration (line 1). The example in Fig. 7.2 (left) shows them visually on the dendogram as vertical intervals (here, $dif0 - 1$ stands for $d_1 - d_0$). Then, the first difference that is significantly dissimilar from those observed so far is selected and the corresponding distance is returned. The procedure takes the dendogram $\mathfrak{D}$ and the *cut-criteria*, and returns the value to cut the dendogram. Lines 3–6 sequentially scan the list of distance differences previously computed, and two different tests are performed, depending on whether or not the values seen so far are large enough to apply statistical tests (line 3). ρ indicates the minimum number of observations to do it, and in our experiments we fixed ρ=3 (thus not requiring any further parameter selection) following the indications in [25]. If enough values are available, then the statistical test is run to decide whether to cut the dendogram (Boolean variable *cut*). Otherwise, a simple test is performed, which checks whether the current distance is more than twice the last one. If the test gives a positive result, the distance d_{i-1} is returned, to be used as cutting threshold. If no difference passes the test, then no cut is possible, and zero is returned, i.e. the dendogram is cut at level zero and no aggregation is needed ($\mathcal{D}_P$=$\mathcal{D}_P^*$).

From empirical studies on real GPS data, we observed that the dendograms produced by Single Linkage always have significant high peaks representing possible turning values for the cut and the statistical criteria select the best one. As preliminary experimentation, we explored also the option of looking for the cut distance applying the statistical criteria directly on the distances between the medoids $\mathfrak{D}$ (d_i) instead of on their differences (d_i−d_{i-1}). However, in this way the right value for the cut is not well identified, usually leading to large locations erroneously aggregated. This happens because the distribution of distances is smooth and it is difficult for the *cut-criteria* to identify a peak.

The *cut-criteria* considered in TOSCA come from the outlier detection theory. The idea behind this choice is the fact that in our empirical experiments we discovered a common distribution of the value of *diffs* showing a sudden spike indicating the change of trend in the aggregations of the clusters. In particular, we adopted the following criteria [25]:

- *Thompson Tau Test* takes into account the mean μ and standard deviation σ of a distribution, and provides a statistically determined rejection region τ determined as

$$\tau = t_{\alpha/2}(n-1)/\sqrt{n}\sqrt{n-2+t_{\alpha/2}^2}$$

where n is the number of values in the distribution, and $t_{\alpha/2}$ is the critical Student's t value based on $\alpha = 0.05$. Given x, if $|(x-\mu)/\sigma| > \tau$ then x is an outlier.

- *Interquartile Range* defines an interval out of which a value is considered an outlier.

$$IR = [Q_1 - k(Q_3 - Q_1), Q_3 + k(Q_3 - Q_1)]$$

where Q_1, Q_2 and Q_3 are, respectively, the 25th, 50th and 75th percentile of the values distribution, and k is fixed to $k = 1.5$. Given x, if $x \notin IR$ then x is an outlier.

- *Chauvenet's Criterion* is based on how many times t the given value x differs from the mean μ in terms of standard deviations σ. It uses the normal distribution to determine the probability p that a value will be at the value of x: p is the probability of having a value at distance $t \cdot \sigma$ from μ. Then p is multiplied by n (number of values in the distribution), and if $p \cdot n < 0.5$ then x is an outlier.

TOSCA can remind the BIRCH algorithm [322]. They both first seeks to do a fine-grained clustering of points and then a second clustering step is performed. Despite similar hierarchical approaches, they differ from the second step. TOSCA re-cluster the medoids of the fine-grained clusters into coarse-grained clusters, while, on the other hand, BIRCH incrementally and dynamically clusters incoming points by attempting to further refine the fine-grained clusters by clustering the points inside these clusters.

Complexity

The complexity of TOSCA is dominated by the complexity of X-Means, i.e., $\mathcal{O}(|S|\log(|M|))$ [155]. Indeed, *single-linkage* complexity is $\mathcal{O}(|M|^3)$ which becomes $\mathcal{O}(|M|^2)$ if, as in our case, Sibson's version is used [265], with $|M| \ll |S|$. Finally, *get-medoids* is linear in $|\mathcal{D}_S|$, *select-cut* is linear in $|\mathfrak{D}|$, and *cut-dendogram* and *aggregate-clusters* are linear in $|M|$.

Name	**Measure**
δ_k	$k - k^* \in (-\infty, +\infty)$
ari	$\frac{ri - E[ri]}{\max(ri) - E[ri]} \in [-1, 1]$
ami	$\frac{mi - E[mi]}{\max(H(\mathcal{L}_P), H(\mathcal{D}_P)) - E[mi]} \in [0, 1]$
$v - measure$	$\frac{2hc}{h+c} \in [0, 1]$ $h = 1\text{-}\frac{H(\mathcal{L}_P, \mathcal{D}_P)}{H(\mathcal{L}_P)}$ $c = 1\text{-}\frac{H(\mathcal{D}_P, \mathcal{L}_P)}{H(\mathcal{D}_P)}$
$f - measure$	$\frac{2pr}{p+r} \in [0, 1]$ $p = \frac{TP}{TP+FP}$ $r = \frac{TP}{TP+FN}$
acc	$\frac{TP+TN}{TP+TN+FP+FN} \in [0, 1]$
$silhouette$	$\frac{b-a}{max(a,b)} \in [-1, 1]$
sse	$\sum_i^{\|P\|} \sum_j^{\|\mathcal{D}_P\|} dist(p_i, d_j) \in [0, \infty)$

Table 7.1: Evaluation measures: top group external, bottom group internal.

7.1.3 Experiments

In this section we provide a broad experimental comparison of TOSCA algorithm against a wide selection of competitors. The comparison will be performed on synthetic datasets generated through two different generative models, and the evaluation will be based on several standard quality measures. Finally, a case study on real GPS data is presented, with a summary of qualitative and quantitative results.

Evaluation Measures

Clustering evaluation measures evaluate how well a given clustering defines separations of the data similar to some ground truth set of clusters (*external measures*), or how much it meets some specific assumption such that members of the same cluster are more similar than members belonging to different ones (*internal measures*). All the measures we adopted – listed below and formally defined in Tab. 7.1 – quantify the quality of the clustering, i.e., the larger the value obtained, the better, with the exception of the first (Delta K) and the last one (Sum of Squared Error).

In the following definitions, S is a set of stop observations and $\mathcal{D}_S$ is the set of locations discovered by an algorithm. External measures assume to know the set $\mathcal{L}_S$ of real locations.

- *Delta K (δ_k)* is the deviation between the real number of locations k=$|\mathcal{L}_S|$ and the number of detected locations k^*=$|\mathcal{D}_S|$, the closer is to zero, the better the clustering.
- *Adjusted Rand Index (ari)* [245] measures the similarity of $\mathcal{L}_S$ and $\mathcal{D}_S$ ignoring permutations and normalization. The *unadjusted* Rand index ri is the percentage of pairs of elements $(a, b) \in S^2$ for which $\mathcal{L}_S$ and $\mathcal{D}_S$ agree, i.e. such that a and b belong to the same location in $\mathcal{L}_S$ iff they belong to the same location in $\mathcal{D}_S$. The *adjusted* Rand index is computed as the normalized deviation of ri from its expected value.
- *Adjusted Mutual Information (ami)* [296] measures the agreement of $\mathcal{L}_S$ and $\mathcal{D}_S$, ignoring permutations. The unadjusted mutual information is defined as mi=$\sum_i^{|\mathcal{L}_S|}$ $\sum_j^{|\mathcal{D}_S|} p(L_i, D_j) \log(\frac{p(L_i,D_j)}{p(L_i)p(D_j)})$, with $p(X)$=$|X|/|S|$ and $p(L_i, D_j)$=$|L_i \cap |D_j|/|S|$. Such value is then normalized where $H(\mathcal{S}_S)$=$\sum_i^{|\mathcal{S}_S|} p(S_i) \log(p(S_i))$.
- *V-Measure (v-measure)* [251] is the harmonic mean of *homogeneity* h and *completeness* c, as defined in the table. Here, $H(\mathcal{S}_S, \mathcal{Q}_S) = -\sum_i^{|\mathcal{S}_S|} \sum_j^{|\mathcal{Q}_S|} p(S_i, Q_j) \log($ $p(S_i, Q_j))$, while $H(\mathcal{S}_S)$ and the probabilities p are defined as for *ami*.

The following measures are based on *precision* and *recall*, which are computed as combinations of True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) rates. They are defined as $TP = |S_{\mathcal{D}} \cap S_{\mathcal{L}}|$, $TN = |\overline{S_{\mathcal{D}}} \cap \overline{S_{\mathcal{L}}}|$, $FP = |S_{\mathcal{D}} \cap \overline{S_{\mathcal{L}}}|$ and $FN = |\overline{S_{\mathcal{D}}} \cap S_{\mathcal{L}}|$, where $S_{\mathcal{D}}$ represents the set of pairs that belong to the same group with respect to $\mathcal{D}_S$, and similarly for $S_{\mathcal{L}}$, while $\overline{X} = S \setminus X$.

- *F-Measure (f-measure)* [240] is the harmonic mean of *precision* p and *recall* r.
- *Accuracy (acc)* [240] is the proportion of true results (i.e. true positives and true negatives together) over the total number of cases examined.

If the ground truth $\mathcal{L}_S$ is not known, the evaluation must be performed using only $\mathcal{D}_S$.

- *Silhouette Coefficient (silhouette)* [275] is defined for each point in $\mathcal{D}_S$ and is composed of: a the mean distance between a point and all the other points in the same cluster, and b the mean distance between a point and all the other points in the next nearest cluster. The *silhouette* for $\mathcal{D}_S$ is defined as the mean of the *silhouette* for each observation. A weakness of *silhouette* is that it is generally higher for density-based clusters (e.g. obtained with DBSCAN, OPTICS) than other concepts of clusters.
- *Sum of Squared Error (sse)* [275]. Let $\{c_j\}_{j=1...|\mathcal{D}_S|}$ be the centroids of the sets D_j, *sse* evaluates the partitioning of $\mathcal{D}_S$. The closer it is to 0.0 the better are the clusters. A drawback of *sse* is that it always gets close to zero when the number of clusters becomes very high (equal to 0.0 if all clusters contain only one observation), yet it does not mean that $\mathcal{D}_S$ is a good clustering.

Synthetic Data Generators

Since we have not a ground truth available for real data, in order to apply external evaluation measures we implemented two generative random models to synthesize real locations, that is we generated synthetic clustered observations $\mathcal{L}_S$.

The *Null Model (NM)* generator randomly generates the centers of k locations in a $[0, 1]^2$ Euclidean space, and then populates each location with observations around its center. Such observations are obtained by adding a Gaussian noise to the corresponding center, and for each location a different standard deviation is randomly selected in order to simulate different contexts: $\sigma_1 \dots \sigma_k$. A total of N observations are generated, distributed among the locations in a uniform random way. In our experiments we used the following intervals of parameter values: $k \in [15, 30]$, $\sigma \in [0.005, 0.015]$ and $N \in [200, 500]$. We used *NM* to generate the dataset $\mathcal{S}^u_{NM}$=$\{\mathcal{L}^1_S \dots \mathcal{L}^u_S\}$ containing the set of locations for u users.

The *Mobility-Like Model (MM)* follows three principles described in [268]: *(i)* the mobility of an individual gravitates around a center of mass with a certain radius of gyration, *(ii)* every individual has two main locations that are frequently visited, and *(iii)* the number of visits in every location is regulated by a Zipf distribution (few locations are visited many times and many locations are visited few times). For these reasons, *MM* creates a set of observations and clusters that simulate the locations generated by human behavior according to two statistics extracted from real GPS dataset described in Section 6.1: the *number of observations* N, and the *radius of gyration* R. We extracted from the real data the distributions of the number of observations and of the radius of gyration in order to use them during the generation of the dataset. We discovered that they are not correlated (Pearson's coefficient -0.2027), therefore we can select N and R independently from the distributions. Following principle *(i)*, *MM* generates observations based on a center of mass m and the radius of gyration R, besides the parameters (N, k, σ_i) employed in *NN*.

For each user, the following steps are performed. First, a Gaussian distribution with mean m and standard deviation R (chosen from the distribution inferred from real data) is used to generate the k locations μ_i. The principle *(ii)* is taken into account by generating μ_1 and μ_2 in such a way that m is their center of mass. To respect *(iii) MM* assigns n_i observations for each location $\forall i{=}1 \ldots k$ according to a Zipf distribution, where $\sum n_i{=}N$. The remaining k-2 locations are generated in the space described by $\mathcal{N}(m, R)$ using $\mathcal{N}(\mu_i, \sigma_i)$ until the center of mass m' and the radius of gyration R' of the generated observations are close to m and R within some thresholds T_m and T_R. The observations generated by *MM* are expressed as latitude and longitude. We used *MM* to generate the dataset $\mathcal{S}^u_{MM} = \{\mathcal{L}^1_P \ldots \mathcal{L}^u_P\}$ containing the set of locations for u users. Besides N and R, which are extracted from data-driven distributions, in our experiments we randomly select m, $k \in [15, 30]$ and $\sigma_i \in [25, 250]$ $\forall i = 1...k$, and we set $T_m{=}50m$ and $T_R{=}100m$.

Competitors

We evaluated TOSCA in its variants: Thompson Tau Test (*TT*), Interquartile Range (*TI*) and Chauvenet's Criterion (*TC*). These different versions of TOSCA were compared against the following *parameter-free* and *parameters-based* methods[1]:

- *Parameter-Free Methods (PFM):* Mean Shift (*MS*) [79], Affinity Propagation (*AP*) [104], Single Linkage (*SL*) with dendogram cut at the knee of the curve of distances $\mathfrak{D}$, X-Means (*XM*) [232], K-means silHouette (*KH*) with k selected as elbow of the *silhouette* curve, K-means SSE (*KS*) with k selected as the knee of the *sse* curve, and K-means Rule of thumb (*KR*) with $k = \sqrt{|P|/2}$. For *KH* and *KS* we have multiple runs of K-Means, with $k \in [2, |P| - 1]$ [275].

- *Parameter-Based Methods (PBM):* Grid (*GR*) [147], DBSCAN (*DB*) [102], OPTICS (*OP*) [15] and Bisecting K-means (*BI*) [272]. They were tested with $\varepsilon = \{0.001, 0.005, 0.01, 0.025, 0.05, 0.1, 0.2\}$ in the null model, and $\varepsilon = \{25m, 50m, 100m, 250m, 500m, 1km, 2km\}$ in the mobility-like model. For *DB* and *OP* we set $MinPts = 2$ and we considered each outlier a cluster.

Performance Evaluation

We evaluated the performances on the datasets generated with the null model $\mathcal{S}^{1000}_{NM}$ and the one generated with the mobility-like model $\mathcal{S}^{1000}_{MM}$. Since the two generators use different coordinates systems we used the *Euclidean distance* to solve LDP in $\mathcal{S}^{1000}_{NM}$, and the *great-circle distance* to solve LDP in $\mathcal{S}^{1000}_{MM}$. The experiments were run on a Mac OS X 10.10.2 64 bit, 8 GB RAM, 2.60 GHZ Intel Core i5 processor.

Null Model Dataset. Fig. 7.3 shows the performances of the methods tested w.r.t. *ari* and *f-measure* on the null model dataset $\mathcal{S}^{1000}_{NM}$. The best results are achieved by *SL*, *KH*, *DB* and *OP* with $\varepsilon{=}0.025$, and *BI* with $\varepsilon{=}0.050$. These methods return the correct clusters for almost all each $S{\in}\mathcal{S}_{NM}$, as reflected by the high means values and the compact boxplots in the figure. However, all parameter-based methods prove to be very sensitive to

	Null Model								Mobility-Like Model							
	v-measure		ami		acc		δ_k		v-measure		ami		acc		δ_k	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
TT	0.93	0.21	0.91	0.24	0.89	0.22	***-0.6***	4.6	**0.97**	0.04	**0.94**	0.07	**0.83**	0.17	***0.7***	3.7
TI	0.92	0.22	0.89	0.25	0.87	0.24	***-0.6***	4.57	***0.97***	0.04	0.93	0.08	***0.83***	0.16	***0.7***	3.7
TC	0.98	0.03	0.95	0.07	0.91	0.10	-0.7	3.6	0.95	0.04	0.90	0.08	0.78	0.16	1.6	4.3
MS	0.57	0.07	0.38	0.08	0.20	0.06	-18.0	5.3	0.76	0.14	0.60	0.18	0.20	0.19	-14.8	6.5
AP	0.84	0.05	0.70	0.13	0.50	0.11	53.8	65.9	0.73	0.08	0.49	0.19	0.13	0.08	65.5	73.4
SL	***0.99***	0.01	***0.98***	0.02	***0.96***	0.04	4.0	3.04	0.89	0.17	0.80	0.22	0.65	0.28	7.2	39.8
XM	0.99	0.01	0.97	0.03	0.94	0.06	2.5	3.11	0.91	0.05	0.80	0.10	0.65	0.15	6.3	4.4
KH	**0.99**	0.01	**0.99**	0.02	**0.98**	0.03	**-0.5**	0.88	0.97	0.03	***0.93***	0.05	0.71	0.18	-4.7	3.8
KS	0.98	0.02	0.95	0.06	0.91	0.11	-1.9	3.6	0.93	0.06	0.86	0.11	0.69	0.16	**-0.5**	6.0
KR	0.89	0.05	0.77	0.10	0.59	0.16	-9.8	5.3	0.92	0.05	0.83	0.10	0.39	0.21	-11.6	5.4

Table 7.2: Mean μ and std deviation σ of *v-measure*, *ami*, and δ_K for TOSCA and the PFMs. **Bold** is the best value, ***italic*** the second.

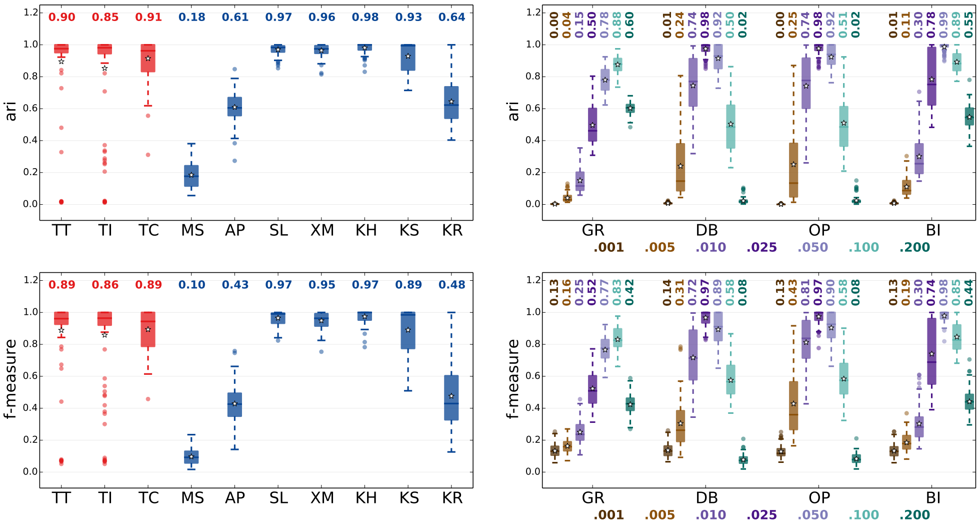


Figure 7.3: Results on $\mathcal{S}_{NM}^{1000}$: *ari* and *f-measure* for TOSCA and the PFMs (left), and for the PBMs varying ε (right). Each box depicts the scores distribution of a method. The mean value, i.e., the star, is reported at the top.

ε, and even small deviations from the optimal one lead to poor performances, while most parameter-free methods achieve good results. We notice that on the null model dataset, the three variants of TOSCA do not yield any improvement to XM. On the contrary, the *second step* of TOSCA introduces errors in the clustering, decreasing *ari* and *f-measure* by around 0.05. Tab. 7.2 (left columns) reports the results obtained with the other external evaluation measures, and shows very good performances of KH algorithm of the K-Means family. Although TT, TI and TC belong to this family, too, they produce a clustering with scores of *v-measure*, *ami* and *acc* about 0.1 points lower than KH and XM.

Mobility-like Model Dataset. The improvement introduced by TOSCA w.r.t. XM to solve the LDP becomes clear when observing the performances in the mobility-like model dataset $\mathcal{S}_{MM}^{1000}$ (Fig. 7.4 and Tab. 7.2 (right)). Among the parameter-free methods, TT and TI obtain the best two performances. They have the highest scores in terms of *ari*, *f-measure*, *v-measure*, *ami*, and δ_K improving the results of XM from 0.1 to 0.2 for each measure. Also, it is important to notice that the box plots of TT and TI are considerably more compact than those of the others – with the only exception of KH – making TT

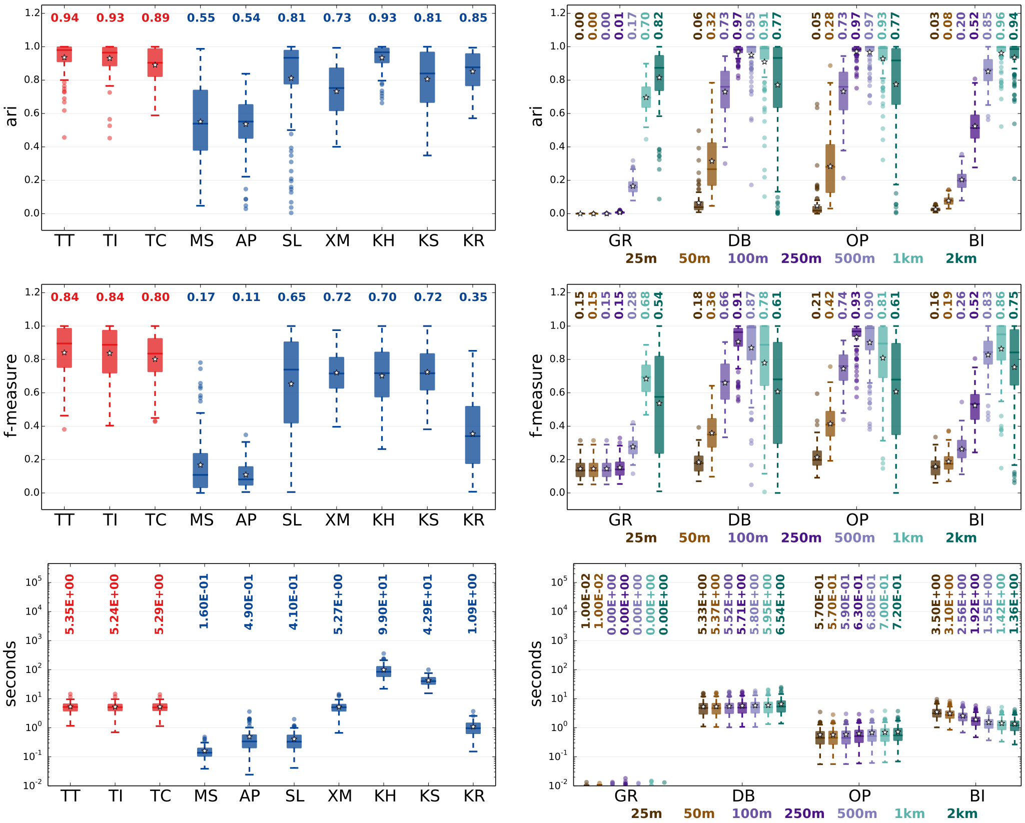


Figure 7.4: Results on $\mathcal{S}_{MM}^{1000}$: *ari*, *f-measure* and *running time* for TOSCA and the PFMs (left), and for the PBMs varying ε (right). Each box depicts the score distribution of a method. The mean value, i.e., the star, is reported at the top.

and TI results more robust than competitors. We can observe the same behavior w.r.t. the standard deviations of the measures in Tab. 7.2. The best competitor in terms of *ari* is KH, yet its performances in terms of *f-measure* are significantly worse and, more important, KH is slower an order of magnitude w.r.t the others. This is a significant weakness of KH due to the calculation of the *silhouette* that is repeated for each point and for each value of k. These results suggest that, among the parameter-free methods, TOSCA better achieves the objective of yielding high-quality results and low computation times.

Moving to parameter-based methods, we notice how DB, OP and BI with ε=250 achieve results better than those of TOSCA. This means that with an appropriate parameter tuning phase they can discover a setting extracting good clusters. As already discussed, this phase usually is very expensive in terms of time and expertise needed by the analyst. Moreover, in real cases the ground truth is not available, and therefore it is impossible to validate the quality of the results or understand how to vary the parameters. This is a strong limitation in the application of these methods.

Finally, we studied the behavior of TOSCA in terms of internal measures, i.e., *silhouette* and *sse* (see Tab. 7.3). With respect to the parameter-free methods, the three TOSCA versions achieve scores which are exceeded only by KH and (quite surprisingly) by KR for the *silhouette*, and by XM for the *sse*. Since KH uses the *silhouette* measure to select the best clustering, its good results on this value were expected, while the values for KR are not supported by the external evaluation measures analyzed before. The *sse* value is

	Mobility-Like Model			
	silhouette		**sse**	
	μ	σ	μ	σ
TT	0.79	0.13	5.19e+04	5.64e+04
TI	0.79	0.12	5.05e+04	5.05e+04
TC	0.74	0.14	***4.65e+04***	3.08e+04
MS	0.69	0.13	6.36e+05	6.59e+05
AP	0.49	0.15	2.40e+05	2.94e+05
SL	0.67	0.20	2.63e+05	5.27e+05
XM	0.62	0.12	**4.04e+04**	6.20e+04
KH	**0.83**	0.12	6.53e+04	6.31e+04
KS	0.69	0.17	6.67e+04	7.97e+04
KR	***0.80***	0.10	1.67e+05	1.46e+05

Table 7.3: Mean μ and std deviation σ of *silhouette* and *sse* for TOSCA and the PFMs. **Bold** indicates the best value, while ***italic*** the second one.

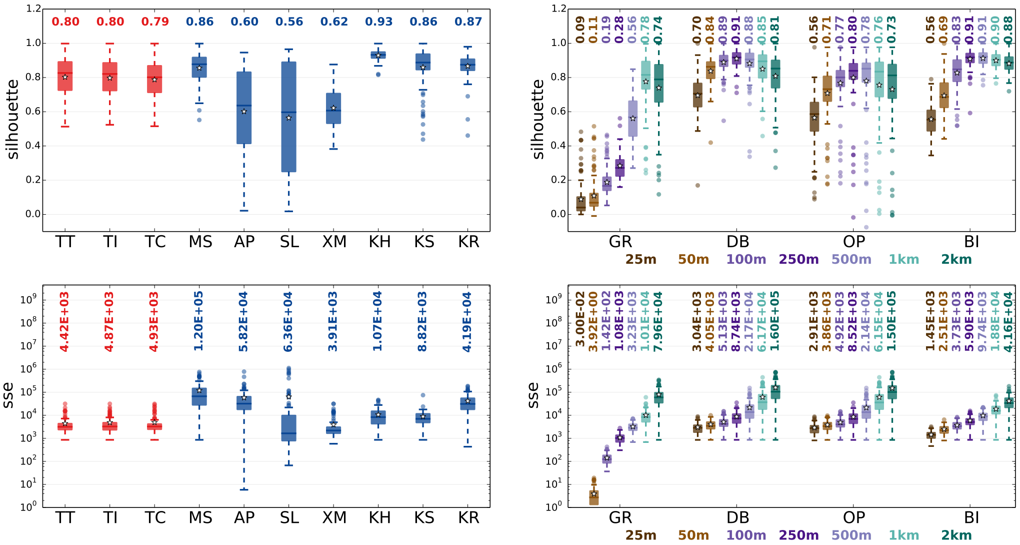


Figure 7.5: Results on the real dataset: *silhouette* and *sse*.

lower for XM than for TOSCA because it has not the second step for the aggregation. Once again, by considering both the external measures and the *see*, we can see how the low level of *sse* in this case is not indicating good clustering. This shows the weaknesses of XM for the LDP previously anticipated. As before, the comparison of TOSCA with the parameter-based methods tells that, for certain values of ε, the performances of the latter are comparable or even better, but at the cost of an expensive parameter tuning step.

7.1.4 Case Study

In this section we show the results on the real dataset of Section 6.1. Note how, in this context, for each user u, the set $\mathcal{D}_u$ of detected locations corresponds to the user profile P_u, i.e. it is a mobility Personal Data Model capturing the personal locations of user u. Fig. 7.5 shows the good performances of TOSCA with respect to the internal evaluation measures. These performances are comparable with those of the mobility-like model dataset. TOSCA is one of the best performers among the parameter-free methods both for *silhouette* and *sse*. Obviously, KH has the best *silhouette* but it has an overall *sse* higher than TOSCA. DB_{250m}, BI_{250m} and BI_{500m} achieve the best results among the competitors with parameters. They all have a *silhouette* higher than TOSCA but they also

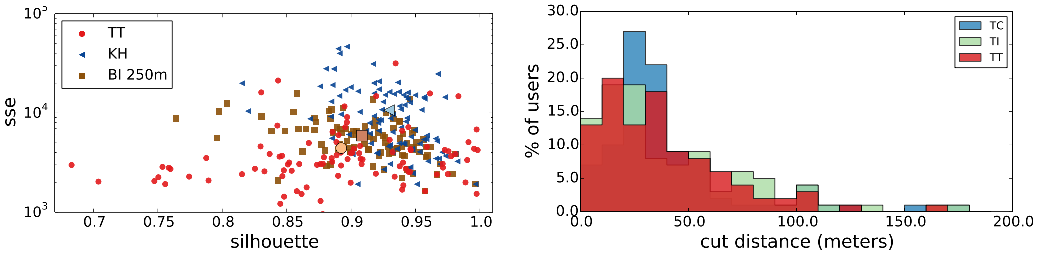
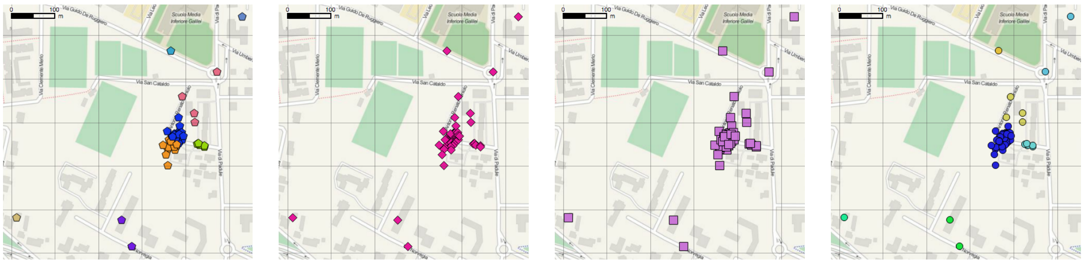


Figure 7.6: *(Left)* Scatter plot of performances of several runs of three algorithms, with the big points representing their centers of mass. *(Right)* Cut distance distribution for TOSCA.



XM DB_{250m} BI_{250m} TT

Figure 7.7: An example of a location detected with XM, DB_{250m}, BI_{250m} and TT.

have a higher *sse*. Hence, it seems that by considering simultaneously *silhouette* and *sse* TOSCA is the method which returns the best clusters. This behavior is captured by Fig. 7.6 (left). Every point represents a run of TT (red circles), KH (blue triangles) or BI_{250m} (brown square), which are the best performers for *silhouette* and *sse*. The big points are the centroids of each algorithm. Even though TT has a *silhouette* slightly worse than the others, it has a *sse* considerably lower since the plot is in log scale w.r.t. the *sse*.

A by-product of TOSCA is the availability of the *cut distance dist*. Fig. 7.6 depicts the distributions of the cut distances for the three variants. A peak appears before $50m$ with a different shape for the three distributions. The fact that the cut distance is always spread and not focused on a single value suggests that the use of a fixed value of ε in methods like DB, OP and BI might be too general, not capturing the variability of mobility behaviors.

Finally, we remark that, good scores in terms of *silhouette* and *sse* do not necessarily mean that the clusters are a good representation of the real locations. Fig. 7.7 illustrates a case in which TT returns a good approximation of the expected cluster for a certain location while XM keeps close observations separated, while DB_{250m} and BI_{250m} produce a large cluster made of observations far away from each other.

7.1.5 Conclusion

We have proposed *TOSCA*, a two-steps parameter-free clustering algorithm for users' locations detection. In contrast to algorithms commonly used in literature, TOSCA automatically detects a good distance threshold for the clusters produced, thus adapting the clustering to the individual mobility behavior of each user in the data. Therefore, it is perfectly suitable as autofocus clustering algorithm to extract Personal mobility Data Models for the PDSs. We evaluated TOSCA against a large set of competitors over data generated from a *null model* and a *mobility-like model*. The results have shown that in the *mobility-like model* and in the *real case study* TOSCA performs better than the general-purpose algorithms producing the desirable clustering for personal mobility data mining.

7.2 Transactional Clustering for Personal Data Mining

Among the large amounts of data generated by each individual, a considerable part consists of *transactions*, i.e. a special kind of categorical data in the form of sets of events like the items purchased in a shopping basket, the web pages visited during a browsing session, the songs listened in a time period, etc. To extract Personal Data Models we need clustering algorithms able to automatically adapt in an efficient way to the wide diversity of individual behaviors. Transactional clustering is the task of discovering into the collection of transactions groups of homogeneous transactions sharing many common items [298].

In the state-of-the-art, all existing methods for transactional clustering either require a parameter tuning process that is not automatic, or require an extremely heavy automated process that does not scale to large user bases (see Section 2.3.2). As a consequence, the repeated application for each user's dataset of any of the existing procedures for the purpose of finding personal clusters is not feasible in presence of a large population of users. In the literature, several approaches have been proposed to address the problem of clustering transactional data. The majority of existing techniques suffer from a drawback: they are dependent on multiple parameters which are difficult to tune, especially in real-life and personal applications. In addition, they do not provide a representative transaction of each cluster, i.e., the set of items that characterize the transactions contained in each cluster.

Consequently, in line with Personal Data Analytics, we propose *txmeans*, an efficient and parameter-free method for clustering transactional data in personal data mining applications. *Txmeans* overcomes the deficiencies of the existing methods, returns a representative transaction for each cluster, and is especially designed for the case where the clustering must be separately applied to a large set of users like those present in the PDE, each one with her personal transactional dataset, i.e., *mass transactional clustering.*

7.2.1 Problem Definition

In this section we define the context and the problem we want to solve. Let $B = \{b_1, \dots, b_N\}$ be a set of N baskets (or transactions) and $I = \{i_1, \dots, i_D\}$ a set of D items. A basket b_i is defined as a subset of items where $\emptyset \subset b_i \subseteq I$.

Personal Transactional Clustering. Given the set of baskets B_u of a user u, the *personal clustering* problem consists in partitioning B_u into K of disjoint sets $\mathcal{C}=\{C_1, \dots, C_K\}$ and extracting a corresponding set of representative transactions $R=\{r_1, \dots, r_K\}$ such that $\mathcal{C}$ is optimal in terms of homogeneity and simplicity. This means that the baskets in each C_i must exhibit a high degree of overlap in comparison to any transaction in $B \backslash C_i$, while keeping the clustering structure concise. Notice that, in general, the subset of items of a cluster $I_i \subseteq I$ might be not disjoint to the subset of items I_j of other clusters.

Mass Transactional Clustering. Given a large set of users U, the *mass clustering* problem consists in solving the personal clustering problem for each user $u \in U$.

Since the number of users $u \in U$ can be very large, the above problem definition implies some technical requirements on the methods aimed to solve it. First, the personal clustering of each different user can yield a different number of clusters, which needs to be automatically determined, since the intervention of an expert is impractical. Second, since several runs of the clustering are needed, one for each user, the algorithm needs to be efficient. Also, each personal dataset can be large, depending on the application and the temporal period covered by the data, thus the algorithm needs to be scalable and

Algorithm 3: *txmeans*(B)

```
   Input  : $B$ - set of baskets
   Output: $\mathcal{C}$ - set of clusters, $R$ - set of representative baskets
 1 $r \leftarrow getRepr(B)$;                                        // extract representatives
 2 $\mathcal{Q}.push(\langle B, r\rangle)$;                                    // initialize queue
 3 $R \leftarrow \emptyset$; $\mathcal{C} \leftarrow \emptyset$;                                       // initialize result
 4 while $|\mathcal{Q}| > 0$ do
 5     $\langle C, r\rangle \leftarrow \mathcal{Q}.pop()$;
 6     $I \leftarrow \bigcap_{b \in C}$;                                   // calculate common items
 7     $C^* \leftarrow \{c \setminus I | c \in C\}$; $r^* \leftarrow r \setminus I$;                // remove common items
 8     $C', C'', r', r'' \leftarrow bisectBaskets(C^*)$;                       // split cluster
 9     $bic_o \leftarrow bic(\{C^*\}, \{r^*\}, |C^*|_N, |C^*|_D)$;                    // BIC original
10     $bic_s \leftarrow bic(\{C', C''\}, \{r', r''\}, |C^*|_N, |C^*|_D)$;              // BIC split
11     if $bic_s > bic_o$ then
12         $C' \leftarrow \{c \cup I | c \in C'\}$; $r' \leftarrow r' \cup I$;             // restore common items
13         $C'' \leftarrow \{c{\cup}I | c{\in}C''\}$; $r'' \leftarrow r'' \cup I$;            // restore common items
14         $\mathcal{Q}.push(\langle C', r'\rangle)$; $\mathcal{Q}.push(\langle C'', r''\rangle)$;              // update queue
15     else
16         $\mathcal{C} \leftarrow \mathcal{C} \cup \{C\}$; $R \leftarrow R \cup \{r\}$;                      // update result
17     end
18 end
19 return $\mathcal{C}, R$;
```

applicable to big data. While each of the above requirements is satisfied by some existing algorithm, there is no approach meeting all of them together. According to Personal Data Analytics, in the following we introduce an algorithm able to do it, also being competitive against the state-of-art on each single requirement.

7.2.2 Method

Txmeans: is a parameter-free hierarchical divisive clustering algorithm based on iterative bisections with the extraction of a representative transaction for each cluster. It implements a greedy search solution to the personal clustering problem, seeking a local optimum in terms of the *BIC* measure. Its primary objective is to reach efficiency without sacrificing the clustering quality and without any parameter tuning phase.

Txmeans Algorithm

In analogy with [60, 232], we address the clustering problem through a top-down, divide-and-conquer strategy: we start from an initial set containing a single cluster, then, iteratively we try to split a cluster into sub-clusters. In the literature it has been proved that clustering methods using *bisecting* strategies [136, 232] are able to produce the best clusters in several different contexts. The general schema of *txmeans*, which implements this approach, is specified in Algorithm 3. The algorithm starts extracting a representative basket r for the whole dataset B, and puts both B and r into a queue $\mathcal{Q}$ (lines 1–2). The queue $\mathcal{Q}$ keeps track of the set of baskets to be considered for splitting. The core of *txmeans* is the body of the loop between lines 5 and 16. For each iteration, a cluster C and its representative r are extracted from the queue (line 5). Then the items I which are

```
Algorithm 4: getRepr(B)
Input  : B - set of baskets
Output: r - set of representative baskets
$I \leftarrow \bigcup_{b \in B} b \setminus \bigcap_{b \in B} b$;                // calculate not common items
$\forall i \in I . freq(i) \leftarrow |\{b \in B | i \in b\}|$;            // calculate frequencies
$i \leftarrow 0$; $r_{(i)} \leftarrow \bigcap_{b \in B} b$; $d_{(i)} \leftarrow \infty$;    // initialize variables
while $I \neq \emptyset$ do
    $m \leftarrow \mathrm{argsmax}_{i \in I} freq(i)$;                  // set of max-freq items
    $r_{(i+1)} \leftarrow r_{(i)} \cup m$;                         // update representative
    $d_{(i+1)} \leftarrow \sum_{b \in B} dist(b, r_{(i+1)})^2$;                   // compute SSE
    if $d_{(i)} \leq d_{(i+1)}$ then
        $I \leftarrow \emptyset$;                          // best representative found
    else
        $i \leftarrow i + 1$; $I \leftarrow I \setminus m$;                      // update variables
    end
end
return $r_{(i)}$;
```

common to all the baskets are removed from all the transactions of the cluster and also from the representative. The results are denoted by a star $(*)$. The point of this task is that such items *(i)* provide no useful knowledge for the bisecting step that comes next, and *(ii)* a large number of common items might *flatten* the similarity values, making it more difficult to appreciate the variability that lies in the other parts of the transactions, and therefore potentially affecting the cluster splitting step. Then, the partitioning of C into two disjoint sub-clusters C', C'' is calculated using *bisectBaskets* over the *clean* transactions (line 8). After that, lines 9–10 calculate *BIC* on the original cluster (bic_o) and on the two sub-clusters (bic_s). Here, $|C|_N$ is the number of baskets in C while $|C|_D$ is the number of different items in C. If the split is useful (line 11) the common items are reinserted, and C', C'' and r', r'' are added to $\mathcal{Q}$ (lines 12-14). Otherwise, the original cluster C and its representative basket r are added to the final sets $\mathcal{C}$ and R (line 16).

Txmeans Stopping Criterion

Given a partitioning of cluster $C \subseteq B$ into two sub-clusters C', C'', we need a criterion to decide whether the splitting is actually useful, i.e., it significantly improves the homogeneity of C, in which case it is performed and the procedure reiterates on each sub-cluster. In the literature various quality measures and cost functions have been proposed. However, they are all *global* measures and need to consider the whole partitioning $\mathcal{C}$, and not just C against C', C''. A more *local* measure to drive this decision is the *Bayesian Information Criterion (BIC)* [259], which selects the model with the highest *BIC* value. *BIC* has been successfully employed in various clustering contexts to control the splitting process [232], and to determine the number of clusters. Yet, to the best of our knowledge, it was never considered for transactional clustering, since it involves a variance computation [163] and thus requires central values for each cluster which are unavailable in most transactional clustering methods. The representative baskets computed in our solution provide this kind of information, thus enabling the use of the *BIC* criteria for our purpose. We remark that the Bayesian Information Criterion can be reliably adopted only when the size of the data sample is larger than the data dimensionality, which means $N \gg D$ (see notation in Tab. 7.4). This requirement is typically satisfied by the input dataset B. Moreover, *BIC* is evaluated over each single cluster C, and not the whole dataset (excepted for the first iteration), therefore the actual dimensionality of the transactions can be reduced to the number D_C of items that appear in at least one transaction of C, usually having $D_C \ll D$, since clusters group similar transactions. *Txmeans* further strengthens this property by removing the common items in each cluster before splitting.

A crucial aspect of *txmeans* is the removal of the items that are common to all the transaction of the current cluster, before computing any similarity. Does this step influence the splitting mechanism and the computation of *BIC*? The answer is that it actually changes the Jaccard similarity values involved in most computations, and therefore in theory it can have large effects on both the splitting (*bisectBaskets*) and the BIC values. Yet, as we will see later, empirical testing shows that these changes do not really affect the final outcome, excepted in some cases where the effects are actually positive.

We start by evaluating the effect of common items removal on the Jaccard similarity. Let consider the case where, right after removing γ items from a cluster, two baskets A and B are compared. Then, their similarity J will be computed as $J = |A \cap B|/|A \cup B|$, while the similarity we would have without the items removal, called here $J^{(+)}$, is $J^{(+)} = \frac{\gamma+|A\cap B|}{\gamma+|A\cup B|}$. Keeping the items leads to an increase of Jaccard similarity equal to $J^{(+)} - J = \frac{1-|A\cap B|/|A\cup B|}{1+|A\cup B|/\gamma}$, which is larger when the similarity between A and B is small, when also the two baskets are small and when γ is large. The result is a flattening of similarities towards the value 1. This also suggests an example where removing items affects the cluster assignment in *bisectBaskets*. Let take centroids r' and r'', and a basket A, such that $r' \subset A \subset r''$ and $|r'| = 99$, $|A| = 200$ and $|r''| = 400$. We have that $J(A, r') = 99/200 = 0.495$ and $J(A, r'') = 200/400 = 0.5$, and therefore A would be assigned to centroid r''. Assuming to have removed $\gamma = 10$ items, without the removal the results would be $J^{(+)}(A, r') = 109/210 \simeq 0.54$ and $J^{(+)}(A, r'') = 210/410 \simeq 0.51$, therefore this time A would be assigned to centroid r'. In this case, r' was smaller and therefore the flattening of its J was larger.

Our experiments showed, however, that this kind of situations are uncommon both in real and synthetic data – notice that no attempts were made to filter them. Indeed, several ad-hoc tests have been made by running Algorithm *txmeans* with and without items removal on the same datasets showed little differences of results. On the side of BIC computation, we have again that different values can be obtained with items removal, since it involves the usage of Jaccard similarities. However, *(i)* removing items increases the values of BIC, since the dimensionality of data decreases, meaning (from a more theoretical perspective) a smaller number of free parameters and therefore a better model; and *(ii)* in practice, we verified experimentally that at each candidate split the effects of item removal on the original cluster (bic_o) and on the new pair of clusters (bic_s) are usually similar enough to keep the split decision unchanged.

Txmeans Bisecting Schema

Center-Based Optimization. The cluster splitting process invoked by *txmeans* tries to divide the cluster into two compact subgroups. The criteria adopted to do that is based on a distance function between the cluster elements and a *representative basket*. More formally, given a cluster C our problem is to find a partitioning $\{C', C''\}$ such that: *(i)* C' and

Algorithm 5: *bisectBaskets*(B)

```
   Input  : $B$ - set of baskets
   Output: $C', C''$ - baskets partitioning; $r', r''$ - representatives
1  $i \leftarrow 0$; $SSE_{(i)} \leftarrow \infty$;                                   // initialize variables
2  $r'_{(i)}, r''_{(i)} \leftarrow selectInitialCentroids(B)$;                          // initialize variables
3  while True do
4      $\{C', C''\} \leftarrow assignBasket(B, \{r'_{(i)}, r''_{(i)}\})$;                      // assign baskets
5      $r'_{(i+1)} \leftarrow getRepr(C')$; $r''_{(i+1)} \leftarrow getRepr(C'')$;                  // calc repr
6      $SSE_{(i+1)} \leftarrow \sum_{b \in C'} dist(b, r'_{(i+1)})^2 + \sum_{b \in C''} dist(b, r''_{(i+1)})^2$;
7      if $SSE_{(i+1)} \geq SSE_{(i)}$ then
8          return $C', C'', r'_{(i)}, r''_{(i)}$;
9      end
10     $i \leftarrow i + 1$;                                                          // update variable
11 end
```

C'' are associated to corresponding representative baskets r' and r''; *(ii)* the partitioning minimizes the Sum of Squared Errors $SSE = \sum_{b \in C'} dist(b, r')^2 + \sum_{b \in C''} dist(b, r'')^2$, where $dist(a, b)$ is a distance function based on the measure of overlap of items between sets a and b. A consequence of our notion of optimality is that each basket belongs to the cluster minimizing the distance with its "centroid", i.e. maximizing the overlap among the items.

Distance Function. While the proposed method could in principle incorporate any *distance function* for comparing transactions, in practice the design of *txmeans* reduces the number of reasonable choices. In particular, the function should be based on the number of items shared between the transactions, which suggests measures such as set intersection, match similarity or Jaccard coefficient. Since the latter is known to be more robust and adequate for sparse vectors (like transactions), *txmeans* adopts Jaccard distance as default. We want to stress that a strategy like the one proposed by *k-modes* algorithm for categorical data does not work for sparse transactional data. Indeed, if we binarize a sparse transactional dataset in the corresponding categorical dataset, the zeros usually predominate, and therefore the corresponding modes will be zero for most of the columns, i.e. the centroids will be empty. Also, alternative approaches where lower thresholds are adopted (i.e. lower than the 50% involved by the mode) might avoid empty centroids but would introduce a new parameter to set, since each dataset might require a different threshold value.

Representative Baskets (*getRepr* function). Following [115], we extract the *representative baskets* with a parameter-free heuristics that first selects the items present in all the transactions of the cluster (lines 1–3 of Alg. 4), then refines such approximation by adding the most frequent items (lines 5–6: notice that m is, in general, a set of items having the same frequency), iterating the process as long as each step improves the solution in terms of error (lines 7–8), thus stopping when a locally optimal representative is generated.

Bisecting Schema (*bisectBaskets* function). In our algorithm we exploit the representative baskets for partitioning a given set of baskets B into two disjoint sets C' and C'' by means of a *bisecting* procedure. Alg. 5 reports the pseudo-code of this method. First of all, two representatives are selected among the baskets of B (line 2). These initial centroids are selected by function *selectInitialCentroids*, which randomly picks several pairs of baskets, and then returns the pair that shows the highest distance value. Then the *assignBasket* in line 4 compares each basket $b \in B$ to the two representative basket and associate it to the closest one. When all baskets have been assigned to a cluster, the representatives are computed through the *getRepr* function (line 5). The process is reiterated as long as the SSE error obtained at step $i+1$ is better than that of step i (lines 6–8).

Txmeans Theoretical Analysis

We provide a few theoretical properties about our algorithms, in terms of termination and computational complexity.

Theorem 1 (Termination)**.** *The* txmeans *algorithm terminates for any input dataset.*

Proof. Both *getRepr* and *bisectBaskets* terminate for any input data. Indeed, the stop condition of the loop in *getRepr* is that set I becomes empty, since it decreases strictly monotonically at each iteration (steps 9 or 11–12). Also, *bisectBaskets* follows the classical k-means structure, and the loop stops when the SSE does not strictly increase. Since the number of possible clusterings, and thus of possible SSE values, is finite, the strictly monotonic sequence of SSE values produced throughout the iterations must eventually reach a (local) minimum in a finite number of steps. Finally, the loop in Alg. 3 iteratively removes a cluster and replaces it with strictly smaller ones. In the worst case all clusters will be broken down to singletons in a finite number of steps, then each set will pass through the *else* branch of the condition on line 11, since the splitting will result in another singleton plus an empty set, which does not improve the *BIC* value. That avoids any possible unbounded loop, leading to termination. □

Theorem 2 (Complexity)**.** *The computational complexity of* txmeans *is $O(It \cdot N^2 \cdot D)$, where It represents the number of iterations required to reach convergence in a single run of* bisectBaskets*, $N{=}|B|$ is the number of transactions in input and D is the number of distinct items in the dataset.*

Proof. In the worst case, *txmeans* ends only when singletons are obtained, i.e. the tree representing the bisections is rooted in B and has N leaves. That implies that the number of clusters produced in the process is $O(N)$, corresponding also to the number of iterations of the loop executed in the worst case. All the operations performed at each iteration involve scanning the transactions in the cluster only once, thus the overall cost is $O(N \cdot D)$, notice that the size of the clusters is always $O(N)$ in case all iterations produce extremely unbalanced splits. The only exception is the execution of *bisectBaskets*. It follows a k-means structure with $k{=}2$, and all the operations performed at each step are linear in the number of transactions and their length. There is no clear bound on the number of iterations required to converge, which is then kept as a parameter It of the complexity. That leads to a cost of *bisectBaskets* equal to $O(It \cdot N \cdot D)$, which dominates the complexity of each iteration of *txmeans*. The overall complexity, thus, results to be $O(It \cdot N^2 \cdot D)$. □

The theoretical complexity of *txmeans* is similar or smaller than most competitors in the literature[2]: *Tkmeans* [115], *clope* [311] and *practical* [44] follow a *k-means* structure, i.e., $O(It \cdot N \cdot K \cdot D)$ that is the same as *txmeans*; *Coolcat* [24] has a similar cost, plus a $O(S^2)$ due to the initialization over a sample of size S, that dominates the complexity if $S > \sqrt{N}$; *Rock* [125] has a larger cost, equal to $O(N^2 \cdot D \cdot \log N)$; and, *Atdc* [60] iteratively performs a partitioning having cost $O(It \cdot N)$ followed by a stabilization step $O(It' \cdot N \cdot K)$. The two steps are repeated till convergence over each current cluster, thus leading to an overall cost of $O(It \cdot It' \cdot It'' \cdot N^2 \cdot K^2)$, where It, It' and It'' represent the number of iterations for each component of the algorithm.

[2] Where not explicitly presented by the authors, we inferred the complexity of each method from the corresponding papers.

Algorithm 6: *txmeans-sampling*(B)

```
Input  : $B$ - set of baskets
Output: $\mathcal{C}$ - set of clusters, $R$ - set of representatives
$S \leftarrow selectSample(B)$;                          // random sampling
$\langle C_S, R\rangle \leftarrow txmeans(S)$;           // run txmeans
$\mathcal{C} \leftarrow assignBasket(B, R)$;             // k-nearest-neighbour
return $\mathcal{C}, R$;
```

We remark that the proof of complexity considers the worst case where all clusters processed have a size and number of distinct items similar to the input dataset. In practice, extremely unbalanced splittings, which would produce clusters of large size, are unusual on real data; also, clusters tend to group together similar transactions that contain a relatively small subset of items. That is also amplified by the removal procedure of common items performed in *txmeans*. The result is that empirical run times tend to grow much more slowly than what predicted by the theoretical analysis.

Dealing with Large Personal Datasets

The *txmeans* method has been designed for Personal Data Analytics, therefore in a data mining context where the transactions of a user are processed separately from the others. While this situation usually results in executing *txmeans* on several small- or medium-size dataset, we might need to move to a Big Data context where each single user has a very long history of transactions, therefore calling for a scalable approach. The *txmeans* method can be easily adapted to integrate a sampling strategy, where the clustering structure is computed on a subset of transactions, and then it is used to classify (i.e. associate to a cluster) the rest of the input dataset. That is made efficient thanks to the computation of representative baskets, which are used to classify transactions following a standard *k-nearest-neighbour* strategy with k=1, i.e. each transaction is associated with the closest representative basket and to its cluster. Alg. 6 shows the structure of the method. Step 1 randomly selects a number $S_N{=}|S|$ of transactions from dataset B. In particular, we follow the approach proposed in [172], where such number is estimated as $S_N{=}ss/(1{+}(ss{-}1)/N)$, where $ss{=}Z^2p(p-1)$. Z and p are fixed and set, respectively, to the z-score of confidence level 99% and to 0.5 (but potentially modifiable for very special cases by expert users). Steps 2 and 3 cluster the transactions sample and then classify all the dataset through the procedures *txmeans* and *assignBasket*. Empirical results show that, not only the quality of the results obtained with samples is very high, but also sampling even improves them. This is mainly due to the presence of noisy rare items, whose impact appears to be reduced by sampling, since most of them will not appear in the sample and will not distort the clustering structure. We will refer to Alg. 6 simply as *txmeans*.

7.2.3 Experiments

In this section we accurately evaluate the performances of *txmeans* both for *personal* and *mass* transactional clustering. According to the literature [44, 60, 125, 308], we evaluated our clustering approach and its competitors on both synthetic and real datasets[3].

[3] The python code for the algorithm proposed, the competitors and the synthetic data generators can be found at https://goo.gl/uuKWSi.

Evaluation Measures

To evaluate the clustering quality we compared the results of *txmeans* with the real clusters. To quantify the similarity between the two sets of clusters we used the *Normalized Mutual Information (NMI)* [296]. *NMI* was preferred over *purity* because *(i)* it is more sensitive than purity to the change in the clustering results, and *(ii)* it takes into account unbalanced distributions and does not necessarily improve when the number of clusters increases (as purity does). Given two sets of clusters $\mathcal{C}$ and $\mathcal{G}$, we have

$$\mathrm{NMI}(\mathcal{C}, \mathcal{G}) = \frac{I(\mathcal{C}, \mathcal{G})}{0.5 * H(\mathcal{C}) + 0.5 * H(\mathcal{G})} \in [0, 1]$$

where $I(\mathcal{C}, \mathcal{G}) = \sum_k \sum_j \frac{|c_k \cap g_j|}{N} \log \frac{N|c_k \cap g_j|}{|c_k||g_j|}$ is the mutual information [296], and $H(\mathcal{C})$ is entropy [262]. Good clusterings have a *NMI* $\sim$1, bad clusterings $\sim$0. In addition, like in *TOSCA*, we keep track of the *deviation* δ_k between the real number of clusters and the number of clusters detected, $\delta_k = |\mathcal{C}| - |\mathcal{G}|$. Finally, we indicate with *RT* the running time (in seconds) for the clustering computation. All experiments were run on a Mac OS v10.11.4, 2,6 GHz Intel Core i5, 8GB DDR3.

Competitors

We evaluated our method against several competitors sharing some features with *txmeans*, yet following different algorithmic structures. Algorithms *practical* [44] and *atdc* [60] are both parameter-free. Algorithm *practical* has two main steps: in the *allocation* step it scans the data and assigns each basket to an existing cluster or to a new one according to a cost function inspired by "tf-idf" [255]; then, through the *refinement* step it moves the baskets from a cluster to another one. Note that the structure of *practical* and *txmeans* are completely different. On the other hand, *atdc* adopts a divisive approach similar to *txmeans*, but *atdc* scans the baskets and iterates between a partitioning and a stabilization phase. The algorithm *tkmeans* [115] adapts the definition of distance used in *k-means* [275] to represent transactions dissimilarity, and computes centroids using the same approach of *txmeans*. Finally, *coolcat* [24] works on a random sample of the baskets, as *txmeans*. We also report the performances of *clope* [311] and *rock* [125], since they represent reference approaches and were thought for market-basket data, which is analyzed in our case study. *Clope* requires a *repulsion* parameter r that is difficult to be interpreted, while *rock* requires the number of clusters and a similarity threshold θ. We avoid the comparison with *subcad* [110], *limbo* [14], *clicks* [320] and *largeitem* [298] because *practical* outperforms *subcad*, *atdc* outperforms *limbo* and *clicks*, and *clope* outperforms *largeitem*.

We must notice that *txmeans*, *practical* and *atdc* automatically estimate the number of clusters, while *tkmeans*, *coolcat*, *clope* and *rock* are parameter-based methods. In the first set of experiments for general purpose applications we advantage this last category by setting them the optimal parameters: we used the real number of clusters k for *tkmeans*, *coolcat* and *rock*; for *clope* we set as r the value minimizing δ_k for $r \in [1.0, 3.5]$ (step of 0.1); and for *rock* we set as θ the value of the best clustering for $\theta \in [0.1, 1.0]$ (step of 0.1). Finally, for *coolcat* and *rock* we selected the sample S using the same function adopted by *txmeans*.

Symbol	Description
N	number of baskets
D	number of items
T	average length
C	number of clusters
P	percentage overlap
O	outliers percentage

Table 7.4: Symbols and descriptions.

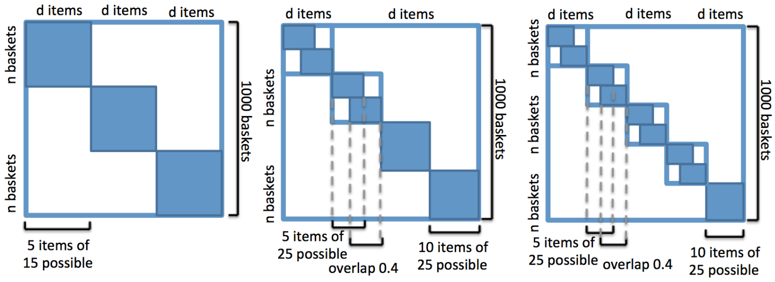


Figure 7.8: Structure of synthetic data for $DS1$, $DS2$ and $DS3$.

Personal Clustering Evaluation on Synthetic Datasets

In this section we analyze the standard context of clustering over a single transactional dataset, corresponding to our *personal transactional clustering* problem. We used synthetic data to study the performances of *txmeans* following the experimental approach of [44, 60, 308]. The advantage of synthetic data is that we can control experiments through the tuning of the clustering structure. Tab. 7.4 reports the variables analyzed to study performances variation. We compare the algorithms with respect to *cluster quality* and *running time*. For *txmeans* we also evaluate *scalability* and *sample size* impact.

Synthetic Datasets. We generate three types of synthetic datasets. The first dataset, named $DS1$, has a one-layer clustering structure (Fig. 7.8 *(left)*) with D=75 items and N=1000 baskets. It has C=3 clusters of the same size. Each basket has length T=5 while each cluster is characterized by d=D/C=15 different items. Thus, in order to generate different patterns for each cluster, a basket in the c-th cluster can contain an item j and j=j'+$(c{*}d)$ such that $j' \in [0, 15]$. The second and third datasets, named $DS2$ and $DS3$, are built in the same way of $DS1$, but they have a two-layer clustering structure: in $DS2$ the top layer has four clusters, two of which have sub-clusters (Fig. 7.8 *(center)*); in $DS3$ the top layer has five clusters, four of which have sub-clusters (Fig. 7.8 *(right)*). In both cases the items overlap in sub-clusters is 0.4 and the average basket length is $T \in \{5, 10\}$. More details in Tab. 7.5. These datasets have a well defined clustering structure, and each basket distinctly belongs to one cluster. For each dataset structure we generate ten datasets.

Besides $DS1$, $DS2$ and $DS3$, we generate another family of datasets, named $DS4$ by using the synthetic data generation method described in [60] which was kindly provided by its authors and employed also in [44]. Besides N, D, T and C, the parameters used by the synthetic data generator for $DS4$ are the percentages of outliers O (i.e. proportion of items that do not contribute to form any cluster), and of of overlap P among transactions of distinct clusters. Different combinations of O and P allow to simulate various situations, which enable an objective experimental validation.

Evaluating Cluster Quality. The goal of these experiments is to evaluate the ability of our algorithm to correctly identify clusters in various situations. In Tab. 7.5 are illustrated the performances of *txmeans* and of the competitors for $DS1$, $DS2$ and $DS3$. The deviation δ_k is considered only for parameter-free methods. For each dataset $DS1$-i, $DS2$-i, $DS3$-i, $1 \leq i \leq 10$, we run the algorithms ten times and we report in Tab. 7.5 the average for each indicator. *Clope* is the best performer but we must consider that the best r is provided as input. In practice, this requires an extensive tuning. A deep analysis of this aspect is reported in the next section. The algorithm *txmeans* is the second best performer w.r.t. NMI for $DS2$ and the third for $DS1$ without requiring any tuning phase. Also *tkmeans*,

Algorithm	**DS1**			**DS2**			**DS3**		
	(N=1000,D=75,C=3)			(N=1000,D=100,C=6)			(N=1000 D=125,C=9)		
	NMI	δ_k	RT	NMI	δ_k	RT	NMI	δ_k	RT
txmeans	0.66	**11.5**	***0.42***	***0.87***	**6.8**	***0.43***	0.83	**10.4**	***0.62***
practical	0.57	51.8	20.96	0.73	***20.7***	6.06	0.68	***36.6***	12.22
atdc	0.53	***49.1***	130.86	0.65	48.4	189.52	0.72	49.5	261.92
tkmeans	***0.67***	-	2.12	0.75	-	3.41	***0.86***	-	4.79
coolcat	0.01	-	14.4	0.22	-	24.97	0.34	-	31.94
clope	**1.00**	-	**0.16**	**0.99**	-	**0.10**	**0.99**	-	**0.10**
rock	0.00	-	10.73	0.0	-	9.15	0.01	-	8.79

Table 7.5: Performances on $DS1$, $DS2$ and $DS3$. **First** best performer, ***second*** best performer.

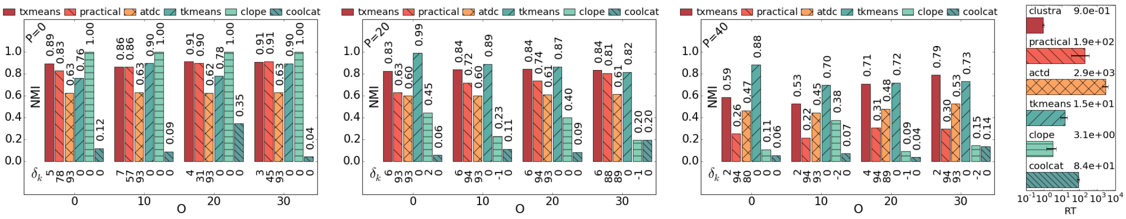


Figure 7.9: NMI, δ_k and RT evaluation for comparing algorithms on synthetic datasets $DS4$ with N=2000, D=200, T=10 and C=6.

that is a fundamental building block of *txmeans*, has very good performances. All the parameter-free algorithms overestimate the number of clusters, but the overestimation of *txmeans* is much smaller than that of *practical* and *atdc*. Finally, *txmeans* has the smallest RT. Since *rock* performances are very poor we do not report its results in the following.

To provide a variety of data structures we exploited $DS4$ for generating different groups of synthetic data sets with controlled overlap percentage $P \in \{0, 10, 20, 30, 40, 50\}$ and outliers percentage $O \in \{0, 10, 20, 30\}$. We do not consider baskets containing patterns with overlap higher than 50% because in real datasets this would not appear frequently. Similarly, it is quite unreal to have more than 30% of outliers. With respect to the other dimensions we fixed $N = 2000$, $D = 200$, $T = 10$, and $C = 6$. We remind that parameter-based methods are provided with the optimal parameter setting.

Fig. 7.9 delineates the performances through barplots of NMI, and with the δ_k reported below the bars. When varying the percentage of outliers O for a given levels of overlap P, the NMI of all the algorithms has small fluctuations. For P=0 we have results comparable with those of $DS3$, while the overall level of NMI decreases significantly when P grows. The two most stable algorithms are *txmeans* and *tkmeans*, but the latter is given the right number of clusters as input. However, *txmeans* maintains good performances for $P \geq 30$ and O=30 and overcomes also *tkmeans*. *Clope* is the best performer when there is no overlap and no noise, then its performances rapidly decrease for growing P. *Coolcat* is always the worst performer. Finally, it is worth to notice that, even though the difference of NMI between *txmeans* and *practical* is not very big, *practical* has a significant deviation δ_k, which represents a clear weakness of this approach. The average RT values are reported on the right of Fig. 7.9. The algorithm *txmeans* is the most efficient: its RT is one order of magnitude smaller than *clope* and two orders of magnitude smaller than *practical*. Moreover, like *coolcat*, due to the sampling function *txmeans* is also the most constant in RT.

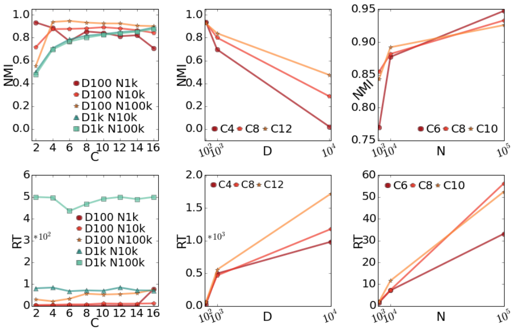


Figure 7.10: Dataset $DS4$. From left to right: scalability w.r.t. clusters C, items D, and baskets N.

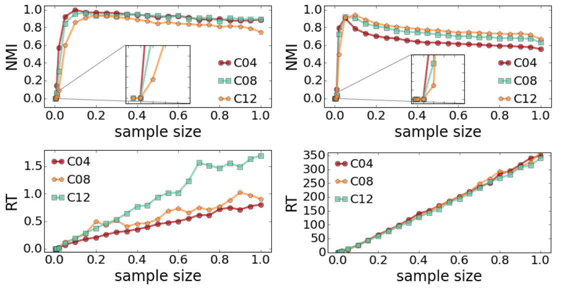


Figure 7.11: NMI and RT by varying sample size on synthetic datasets $DS4$ with N=1000, D=100, T=20,P=20 and O=10.

Evaluating Scalability. We evaluate the performances of *txmeans* on $DS4$ by varying N, D and C. In all datasets considered we fix T=20, P=20% and O=20%. The *first column* of Fig. 7.10 shows the scalability varying C. For NMI we observe that when there are few clusters there are better performances with small datasets while with many clusters there are better performances for large datasets. As expected, the RT does not fluctuate when varying C. In the *second column* of Fig. 7.10 we find the performances varying D. We fix N=100,000. The NMI decreases more slowly for higher number of clusters. The RT grows less than linearly in D and it is higher for high C. Finally, in the *third column* we observe the scalability varying N (we fix D=100). The NMI grows with N: the more the baskets, the less the noise, and the better are the representatives. The RT grows linearly and it is not influenced by the number of clusters.

Evaluating Sample Size. In this section we evaluate *txmeans* on $DS4$ varying the size of the sample S. Fig. 7.11 illustrates the NMI and RT when changing the *sample size* for T=20, D=100, N=1,000, P=20, O=10 (left); and T=20, D=1,000, N=10,000, P=20, O=10 (right). As previously, we report the average values of ten runs. A clear trend appears for both datasets: a peak of high values of NMI is positioned in a range of sample size between 0.05 and 0.15, then the trend decreases a little before stabilizing. The range $[0.05, 0.15]$ confirms that the function [172] we adopted to select the size S_N of the sample S is a good choice. Indeed, it returns samples S with a size which is generally between the 5% and the 20% of the whole dataset. Moreover, we can observe a *dual effect* with respect to C: for the "small" dataset (left) we achieve better results when $C \leq 8$, vice versa for the "big" dataset (right) dataset we get better performances with $C \geq 10$. Finally, as expected, the RT grows linearly with the sample size.

Personal Clustering Evaluation on Real Datasets

We used three real datasets: *Mushrooms*, *Congressional Votes* and *Zoo*. They are from the UCI Machine Learning Repository[4]. We ignore the class labels during clustering and we use them as ground truth. For each dataset we perform 100 runs for each algorithm and we report the average values. As depicted in Tab. 7.6, our algorithm performs well on all the real-world datasets. The sampling function is employed only for the *Mushroom* dataset. For the *Mushrooms* dataset *txmeans* is the second best performer with respect to NMI and δ_k and the best performer with respect to RT. Good performances are obtained only

[4] http://archive.ics.uci.edu/ml/

Algorithm	Mushrooms			Zoo			Congress		
	(N=8124,D=22,C=2)			(N=435,D=16,C=2)			(N=10 D=16,C=7)		
	NMI	δ_k	RT	NMI	δ_k	RT	NMI	δ_k	RT
txmeans	***0.41***	***3.5***	**1.08**	**0.83**	**-2.2**	0.07	0.36	7.0	0.30
practical	0.01	**-0.5**	***5.76***	0.43	-5.3	***0.03***	***0.47***	**0.3**	***0.12***
atdc	0.22	4.0	33.42	0.74	***-3.0***	0.09	0.30	***1.0***	0.32
tkmeans	0.16	-	302.65	0.77	-	2.16	**0.48**	-	5.45
coolcat	0.01	-	73.18	0.54	-	0.89	0.41	-	13.01
clope	**0.42**	-	7.75	0.80	-	**0.01**	0.38	-	**0.04**
rock	0.01	-	40.60	0.425	-	0.18	0.44	-	12.95

Table 7.6: Clustering performances on real-world data sets. **First** best performer, ***second*** best performer.

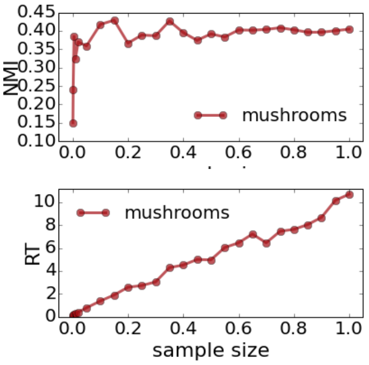


Figure 7.12: *NMI* and *RT* by varying sample size on *Mushrooms*.

by *txmeans* and *clope*: *practical* underestimates the number of clusters, while *atdc* overestimates it. All the parameter-free algorithms underestimate the real number of species in the *Zoo* dataset. Despite this fact, *txmeans* is the best performer and produces a partitioning even better than parameter-based algorithms for which the number of cluster was correctly specified. In *Congressional* dataset our algorithm does not perform well with respect to the *NMI* obtained by the others and overestimates the number of clusters. This is probably due to the nature of the dataset. Overall, the experiments on real datasets suggest that *txmeans* provides consistent and stable results in comparison to the competitors. These results confirm the suitability of our algorithm previously observed on the synthetic data. Finally, we investigate whether also for real-world datasets there is the same trend observed on synthetic data when varying the sample size. The result is reported in Fig. 7.12. Since the trends emerging both for *NMI* and *RT* are very similar to those previously observed we can conclude that this effect is provided by *txmeans* and is not due to synthetic data.

Mass Clustering Evaluation on Synthetic Datasets

In this section we test *txmeans* and its competitors in the context of *mass clustering*, i.e., clustering transactions of several users separately. This more realistic setting emphasizes the challenges that motivated the development of *txmeans*, i.e. efficiency and freedom from parameters. Since a real dataset containing customers transactions annotated with cluster labels is not available, we used $DS4$ synthetic data generator. Unlike previous experiments, in this section we are performing the clustering on a wide set of datasets generated with random structures and we evaluate the performances considering the personal clusterings of all the datasets. We generated $10k$ datasets with characteristics selected uniformly in $N{\in}[1000, 10000]$, $D{\in}[100, 1000]$, $T{\in}[10, 30]$, $C{\in}[4, 16]$, $P{\in}[0, 50]$, $O{\in}[0, 30]$. Moreover, since we are simulating a real application scenario, we are not suggesting the optimal parameters to parameter-based methods. For these algorithms we adopted two versions: the *fixed parameter (fp)* version for which we fix a parameter setting for all datasets, and the *parameter tuning (pt)* version for which we simulate the search of the best parameters for each dataset by running each method several times varying the parameters.

Since in the previous section *atdc*, *coolcat* and *rock* had relatively poor performances, we do not include them in these experiments. We name the parameter-based competitors with fixed parameters *tkmeans-fp* and *clope-fp*, and those with parameter tuning *tkmeans-*

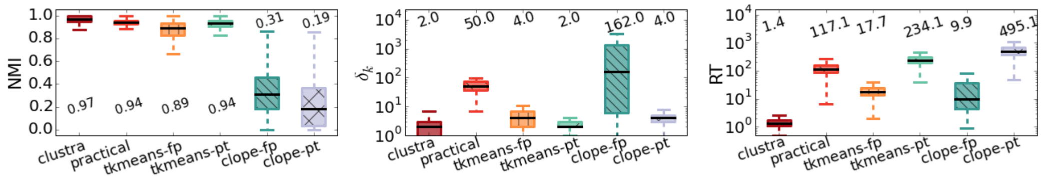


Figure 7.13: NMI, δ_k and RT evaluation for comparing algorithms on 10k synthetic datasets $DS4$ for the mass data mining clustering scenario.

pt and *clope-pt*. For *tkmeans-fp* and *clope-fp* we fixed k=6 and r=2 respectively. We used the heuristic technique known as "knee method" to select the best k and r. In practice, we run *tkmeans* for $k \in [4, 16]$ and we store the Sum of Squared Error (SSE) for every run. Then we select as best k the one corresponding to the point in which the trend of the SSE curve changes, i.e., the "knee" of the curve [275]. We adopted a similar technique for *clope* considering the trend change of the *Profit* function [311] with $r \in [0.1, 3.5]$.

Fig. 7.13 depicts the boxplots of NMI, δ_k and RT. The numbers reported represents the median values, i.e., the black straight line in the middle of each boxplot. For this application *txmeans* shows the best performances: it is able to return for each dataset the purest clusters (a median level of 0.97 NMI) in a few seconds without any parameter tuning and it deviates on average only of 2 clusters from the real number. Considering these three aspects at the same time, this level of performances is reached by none of the competitors. *Practical* and *tkmeans-pt* have a NMI gap w.r.t. *txmeans* of only 0.03. However, both have a median RT two orders of magnitude greater than *txmeans*. Moreover, while *tkmeans-pt* succeeds in minimizing δ_k, *practical* has an average deviation of ~50 clusters, that is quite unacceptable because, even if almost all clusters extracted are pure, they are more than necessary to describe the patterns contained in a dataset. On the other hand, *tkmeans-pt* is highly penalized in terms of RT by the multiple runs for tuning the number of clusters k. Its counter-part, *tkmeans-fp* has lower running times, but also lower NMI and higher δ_k. Note that, *tkmeans-fp* has overall good performances because the most frequent number of clusters for the generated datasets is exactly 6. *Clope* is not competitive in the parameter tuning version, nor in the fixed parameters version. Hence, *txmeans* is the best algorithm for mass transactional clustering.

7.2.4 Case Study

The efficiency and the freedom from parameters brought by *txmeans* make it possible to adopt clustering-based strategies in applications that need to handle massive transactional personal data like those part of the PDE. In this section we present an application of this kind, showing a case study in the domain of recommendation systems that requires analyzing a massive real dataset containing millions of shopping sessions. In this context transactional data are typically treated with very simple and ad hoc strategies, while more complex approaches – including clustering at the individual level – are usually avoided exactly for the same reasons that motivated the development of *txmeans*: difficulty to tune parameters, efficiency issues, etc. The solution proposed represents a first attempt to go in the opposite direction, an approach enabled by the capabilities of *txmeans*.

The application consists of a *Personal Cart Assistant (PCA)* service that in real time suggests to the customers of a retail seller potential products to add to their current basket. Such suggestions, as detailed below, are based on the users profiling through the representative baskets (and therefore clusters) obtained from their purchasing history.

	μ	σ	ν	η	κ
N	236.04	208.44	172.00	150	2.02
D	140.61	35.05	139.00	139	0.11
T	9.96	5.06	8.83	10	1.35
K	4.45	3.59	3.00	2	1.81

Table 7.7: Statistics: mean μ, stddev σ, median ν, mode η, skewness κ.

In the rest of this section we will describe our solution in detail, introduce alternative approaches and show comparative empirical results.

Personal Cart Assistant. Given the baskets B of a customer, we process them with *txmeans* to obtain clusters $\mathcal{C}=\{C_1, \dots, C_k\}$ and representatives $R=\{r_1, \dots, r_k\}$. According to the Personal Data Analytics approach, the tuple $P_u=\langle \mathcal{C}, R \rangle$ represents the profile of customer u, and is used as basis for a model-based collaborative filtering approach [4, 274, 286]. Given the current, incomplete basket $L=\{i_1, \dots, i_n\}$ of the user, we find the representative $r_i \in R$ which is closer to L in terms of Jaccard distance, and then use the transactions in C_i to generate suggestions. A weight is associated with each candidate item to consider, computed as the sum of similarities between current basket L and all baskets in C_i that contain the candidate item, and then only the highest-weight items are suggested. This process can be interpreted as a particular instance of the general *collaborative filtering* approach, with a set of users (here corresponding to single baskets), user's preferences (the items in L), users selected as similar w.r.t. the user's preferences (the baskets in C_i) based on what they bought in the past (r_i). A difference from classical collaborative filtering is the fact that our user's preferences (i.e., the shopping list L) are binary instead of scores. An alternative to the users/baskets similarity-based solution consists of the complementary item similarity-based approach described in [256]. This approach was tested in our showcase, yet the results were basically the same of the first one, therefore we omit it here for the sake of space and readability.

Baselines. We compared the performances of our method, named *pca*, against the following baselines: *last* suggests the items in the last basket purchased; *rand* produces suggestions by picking random items; *most* recommends the most frequent items; *mbcf* is the memory-based collaborative filtering method on transactional data proposed in [206] using the whole set B for collaborative filtering. Although other personalized methods exist in literature for basket recommendation, e.g. [246, 299], none of them try to complete the current shopping list, and thus cannot be directly compared.

Real Dataset. Experiments have been conducted over the real dataset of shopping session described in Section 6.2. We considered about $2,670,343$ shopping sessions that occurred in Leghron province over the years 2010–2013, corresponding to about $10k$ loyal customers, i.e., customers active in at least ten months every year. Tab. 7.7 reports some data statistics (top 3 lines). The number of shopping session N follows a long-tailed distribution with mode 150. The number of items D is a Gaussian distribution with mode $\sim$140 and small standard deviation. Finally, the average basket length T is typically $\sim$10 items.

Recommendation Evaluation. The first three years of data were used to extract the profiles $\{P_c\}$, which were then tested over the last year. The bottom line of Tab. 7.7 shows that the number of clusters K found for each customer follows a long-tailed distribution with mode $\sim$2-3: few patterns are needed to represent the shopping behavior of a customer. There is a strong correlation, 0.74 with p-value equal to zero, between the number of baskets N and clusters K, as expected: the higher the number of shopping sessions, the higher the probability to have many representative baskets.

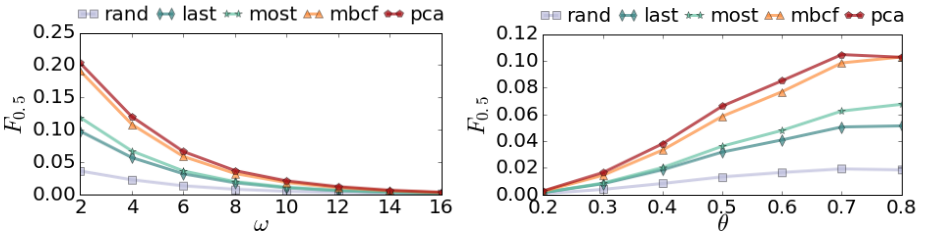


Figure 7.14: Recommendation performances as $F_{0.5}$-measure on real dataset varying minimum basket length ω *(left)*, and basket split θ *(right)*.

For each customer we tested the recommender systems over each session L, in chronological order, updating the models as follows: *pca* assigns L to a cluster with respect to the representative baskets, *mbcf* considers also L into the model, *most* updates the frequencies of each item in L, *last* becomes L, *rand* also consider the items in L in its choices. Only baskets having length at least $\omega \in [2, 16]$ were considered for applying the recommender systems, and the current basket L in the test set was split into two parts w.r.t. a percentage $\theta \in [0.2, 0.8]$. As quality measure we report the $F_{0.5}$-measure [200], which puts more emphasis on precision than recall, aggregated by averaging the scores of all the customers.

Fig. 7.14 shows the performances of the recommender systems varying ω *(left)* and θ *(right)*. The *pca* performs better than the others providing an average improvement of 0.02 with respect to *mbcf*: it means from 1 to 3 additional items correctly suggested. Fig. 7.14 *(left)*, where ω is varying and θ=0.5, shows that the larger is the minimum basket length required, the worse are performances. The fact that for short baskets it is easier to predict the item composition reveals that, with a high probability, frequent items are regularly purchased. Moreover, as expected, Fig. 7.14 *(right)* shows that, for $\omega \geq 6$, the larger is the portion θ of L considered by the recommender systems, the better are the performances. Finally, even for low values of θ *pca* can achieve performances higher than its competitor.

7.2.5 Conclusion

We have presented *txmeans* a parameter-free clustering algorithm for personal transactional data. *Txmeans* is able to solve efficiently the mass transactional clustering problem, i.e., partition efficiently many different personal datasets and provide for each cluster a representative transaction. Our proposal to avoid parameters applies a bisecting strategy to find groups of similar transactions and gives the possibility to work on a sample of the initial data to allow its application also in the context of big data. These features empower *txmeans* to outperform existing algorithms both on synthetic and real-world datasets. Therefore, *txmeans* becomes an eligible Personal Data Analytics method for Personal Data Mining transformations, for the extraction of Personal Data Models, and for the creation of novel personal services in the PDE. Indeed, we have shown an application of *txmeans* on a real dataset of shopping sessions. We built on top of *txmeans* results a *personal cart assistant* able to suggest to the customers the items to put in her shopping list. Finally, it is worth to notice that, due to its ability to efficiently solve both the personal transactional clustering and the mass transactional clustering, *txmeans* could be applied in the distributed environment of the Personal Data Ecosystem both individually by each peer, but also collectively by some peer charged to collect and process the data for a group of users.

Chapter 8

Personal Data Models

A *model* is an abstract representation of a real phenomenon able to explain how the phenomenon works. Very often a model is a simplification of the phenomenon and admits a mathematical formalization. If the phenomena we are trying to explain is completely random, then a model for that phenomena can not be formalized. However, to some extent, human behavior is predictable. Humans do not change their behavior randomly from one day to the other and their patterns usually follow a given routine. This is true at individual level: bursty patterns of activities have been observed and can be predicted, for instance in writing e-mails. Also individual mobility is predictable: most people will commute every working day between the same two points, and can be predicted to do so with very high accuracy [23, 297]. But humans are also predictable at collective level: groups of humans flock together in predictable patterns. For instance, people are more mobile early in the morning and late in the afternoon, around the working day, creating an M-shaped pattern.

Therefore, human predictability, both at individual and at collective level, is a precondition for building a model which catches the bursty behaviors. Through data mining, it is possible to extract and capture the patterns which represent human behavior, and to summarize them in usable and understandable data models. For each user u, these patterns are the Personal Data Models forming the user's individual profile P_u. A profile can be either formed by a set of indicators describing the user attitudes, or by a complex data structure capturing the routinary actions performed by the user. However, in both cases a Personal Data Model is obtained after a Personal Data Mining process applied on the individual history H_u. In the following, we show how it is possible to outline the Personal Data Analytics approach for the extraction of diversified Personal Data Models obtained through different Personal Data Mining processes on distinct types of data.

8.1 Personal Behavioral Entropy and Profitability in Retail

The features describing and summarizing users' habits are often expressed through complex indicators. Indicators are a valuable component in a Personal Data Store because through them a user can quickly understand her own behavior, and improve her self-awareness. Therefore, expressive indicators capturing complex patterns of human behavior are a fundamental component of the models which are part of PDS. In particular, for a user can be useful to figure out how much she is systematic and repetitive in her daily activities. On the other hand, comprehending the level of predictability of a set of users can be a great value for commercial enterprises because knowing for each customer if she is systematic, or not, and in which different dimension, might have important consequences for sales.

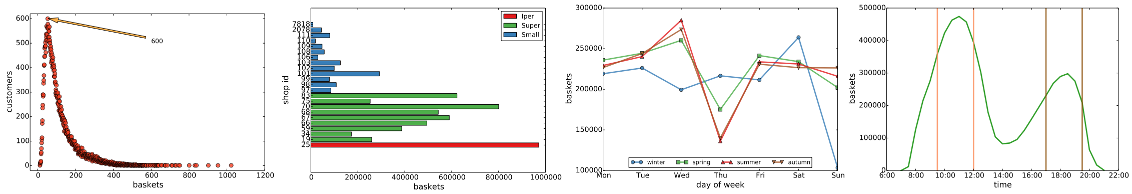
In this section we consider as users the customers of the retail market chain described in Section 6.2. A customer might not be predictable at the time of the day she visits the shop, but she might be highly predictable in the products she always purchases. Having this information expressed as an indicator of predictability, a customer could consider to change her dietary habits, or to focus her shopping sessions in a time of the day when the shops are less crowded instead of going randomly. In addition, from the enterprise point of view, a systematic customer could be more valuable because she spends more, then the shop might want to encourage more and more people to be systematic.

By applying Personal Data Mining techniques, we design two personal indicators for Personal Data Analytic that can be part of the user profile revealing the level of unpredictability of a customer [129]: the *Basket Revealed Entropy (BRE)* measure of how unpredictable a customer's basket is with respect to typical basket compositions, the *Spatio-Temporal Revealed Entropy (STRE)* quantifies how unexpected each customer shopping session in relatively to the spatio-temporal dimension, i.e., the shop, day of the week and time of the day. Thus, instead of using a pure business intelligence approach that can be summarized with the OLAP framework [170] to treat retail market data, we use a more data mining oriented approach. However, we do not use the mined patterns directly as component of the user profile P_u, but we use them to construct systemic measures estimating the degree of an individual's predictability. Then, on top of these systemic measures, we apply a collective data mining step, identifying the main customer classes based on their predictability. The two indicators for the Personal Data Model and the Personal Data Mining procedure to extract them improve over the state of the art by combining both dimensions: they evaluate customer predictability in what they buy and in where they buy it.

8.1.1 Dataset

We build these two indicators on the *Coop dataset* described in Section 6.2. For data cleaning purposes, we perform a series of filters on this dataset. First, we select all the observations recorded during 2012. Second, we focus on a narrow area of operation. The supermarket company was founded in Leghorn and we consider exclusively the shops that are in this Italian province. We do so because the market penetration of the company in this province is so high that we can effectively say that all inhabitants of Leghorn are represented in the data. Finally, we drop all customer who did not perform at least a shopping session per month. The area around Leghorn has a high influx of tourists from other areas of Tuscany, so supermarket customers from other provinces might sporadically use their card in shops in Leghorn province, thus introducing noise in our estimates.

After this filter phase, we have 56,448 customers. We underline that "customers" refers to customer cards and a card can be shared by an entire family. The province of Leghorn had a population of 343,003 in 2012. Assuming an average size of three people per household, we estimate that we cover at least 50% of the population. The total number of distinct products bought is 84,362. The total item scans in the dataset amount to 71,172,672, and it has been generated from 23 shops. Fig. 8.1 depicts stylized facts about shopping sessions (baskets). Fig. 8.1 *(a)* is the number of baskets per customer. The mode is ∼100, meaning that customers usually visit the shops around twice a week. The distribution does not follow the Zipf law because Leghorn does not have enough inhabitants to support it, since



(a) (b) (c) (d)

Figure 8.1: Distributions of baskets per customer (a), distribution of baskets per shop (b), distribution of baskets per weekday (c) and distribution of baskets per time of the day (d).

50% or more of them are actually regulars. Fig. 8.1 *(b)*: the number of baskets per shop. Each of the 23 shops is represented here. There is a correlation between shop type and the number of customers it attracts. Fig. 8.1 *(c)*: the number of baskets per weekday. Customers have a remarkable preference for some days instead of others, also given the season. Fewer shopping sessions happen on Thursday, while Wednesday is the most popular day. Fig. 8.1 *(d)*: the number of baskets per time of the day. An M-shaped pattern appears: most shopping sessions happen in the morning or after working hours. From Fig. 8.1 we see that there are some general patterns in the customer behavior. Customers tend to shop twice a week, they are likely to be attracted to larger shops, they have favorite weekdays and time of the day to perform their shopping sessions. On these observations, we build our personal behavior entropy measures.

8.1.2 Method

The methodology we propose aims at estimating the behavioral entropy of each customer. The two entropy measures are the *Basket Revealed Entropy (BRE)*, and the *Spatio-Temporal Revealed Entropy (STRE)*. These measures tell us respectively how unpredictable is the basket composition and the visiting pattern of a given customer. Thus, the customer profile is formed by the couple of these measures, i.e., $P_u = \langle bre_u, stre_u \rangle$.

Basket Revealed Entropy

The objective of the mining step is to detect what are the behavioral patterns of a customer. There are two types of behavioral patterns in which we are interested: basket composition and spatio-temporal behavior. For the basket composition, we apply a frequent itemset mining algorithm [111]. For each customer, we apply the *Apriori* algorithm [6] on her baskets to detect her patterns. We drop the non-frequent patterns, i.e., the ones that are not present in at least *minsup* baskets. Then, we assign each of her baskets to the largest pattern it contains. Note that each basket must be assigned to a pattern, and a pattern can classify multiple baskets. As alternative, we could have applied the *txmeans* algorithm described in Section 7.2. However, it was developed and refined only after the finalization of the unpredictability measures described in this section.

To better understand the procedure, consider the following example:

1. {Cheese, Banana, Tomato, Bread}
2. {Cheese, Banana, Tomato}
3. {Cheese, Banana, Tomato, Coffee}
4. {Cheese, Banana, Tomato, Bread}
5. {Cheese, Meat, Shoes, Bread}

Setting *minsup*=3, i.e., each pattern has to be present in at least three baskets, the mining algorithm will find the following patterns:

- Support = 5: {Cheese}
- Support = 3: {Bread, Cheese}, {Bread}
- Support = 4: {Cheese, Banana, Tomato}, {Banana, Tomato}, {Cheese, Banana}, {Cheese, Tomato}, {Banana}, {Tomato}.

We name those patterns *representative baskets*. In the following we use patterns and representative baskets as synonyms. Finally, we classify baskets 1 to 4 with the {Cheese, Tomato, Banana} representative basket, because it is the longest pattern contained in them; and basket 5 with the representative basket {Cheese, Bread}. We now have a series of representative baskets with a given probability of appearance for our customer.

The Basket Revealed Entropy (BRE) is calculated following the information-theoretic concept of entropy [262] where RB is the set of representative baskets of our customer, rb_i is the i-th representative basket frequency (i.e., number of occurrences), $\mathrm{f}(rb_i)$ is the representative basket's relative frequency and n=$|RB|$ is the number of representative baskets:

$$BRE(RB) = -\sum_{i=1}^{n} \mathrm{f}(rb_i) \log \mathrm{f}(rb_i) / \log n \quad \in [0,1]$$

BRE is normalized with $\log n$, that is the expected entropy of a fully random set of patterns. In our example we have two representative baskets, with relative frequencies 4/5 and 1/5. Thus, the BRE of our hypothetical customer is ∼0.72.

Spatio-Temporal Revealed Entropy

The calculation of the Spatio-Temporal Revealed Entropy (STRE) is similar in spirit to the procedure outlined in the previous section. However, the first computational step is easier. Here, we connect each basket to its spatio-temporal characteristics. These characteristics are always represented by a tuple of three elements: the shop in which the basket was purchased (which provide the spatial dimension), the time of the day and the day of the week (the temporal dimension). Since all tuples always have three elements, we do not need to perform a mining step, and we can just count the relative frequency of each possible tuple. The relative frequencies are then fed into the entropy formula.

Let consider the time and space of the shopping sessions of our hypothetical customer:

1. Shop 25, Weekend, Evening
2. Shop 19, Weekday, Late Afternoon
3. Shop 19, Weekday, Late Morning
4. Shop 19, Weekday, Late Afternoon
5. Shop 19, Weekday, Early Morning

We have four patterns, three with probabilities 1/5 and one with probability 2/5, which results in an entropy ∼0.96. Note that we aggregate days in two bins, weekday and weekends, as keeping days separate would generate too many fluctuations.

Algorithm 7: BRE(*baskets*, *minsup*)

```
   Input  : baskets - personal set of baskets
            minsip - minimum support for representative baskets
   Output: bre - basket revealed entropy
1  $IS \leftarrow$ getItemSet($baskets, minsup$);
2  $RB \leftarrow$ getReprBasketsCount($baskets, IS$);
3  $bre \leftarrow -\sum_{rb \in RB} \mathrm{f}(rb_i) \log \mathrm{f}(rb_i) / \log(|RB|)$;
4  return $bre$;
```

Algorithm 8: getReprBasketsCount(*baskets*, *IS*)

```
1  $RB \leftarrow \emptyset$;
2  for $b \in baskets$ do
3      $D \leftarrow \{rb \in IS \mid rb \subseteq b\}$;
4      if $D = \emptyset$ then { $RB_b \leftarrow 1$; continue; };
5      $D' \leftarrow \mathrm{argmax}_{rb \in D} |rb \cap b|$;
6      if $|D'| = 1 \wedge D' = \{rb\}$ then {$RB_{rb} \leftarrow RB_{rb} + 1$; continue; };
7      $D'' \leftarrow \mathrm{argmax}_{rb \in D'}\, sup(rb)$;
8      if $|D''| = 1 \wedge D'' = \{rb\}$ then {$RB_{rb} \leftarrow RB_{rb} + 1$; continue; };
9      $D''' \leftarrow \mathrm{argmin}_{rb \in D''}\, lift(rb)$;
10     if $|D'''| = 1 \wedge D''' = \{rb\}$ then {$RB_{rb} \leftarrow RB_{rb} + 1$; continue; };
11     for $rb \in D'''$ do $RB_{rb} \leftarrow RB_{rb} + \frac{1}{|D'''|}$; ;
12 end
13 return $RB$
```

BRE Algorithm

In this section we provide and discuss the pseudocode of our analytic framework. We focus on pseudocode for BRE as it is the most complex. STRE computation does not require a mining step and every basket is already naturally associated with its own triple (shop, day-of-week, time-slot). The full procedure has three logical steps that are reported in Alg. 7.

Step #1 is the detection of the frequent patterns from the baskets of a customer, i.e., the individual history H_u. It can be implemented with any itemset mining algorithm. In our experiments we implemented *getItemSet*($baskets, minsup$) with *Apriori* [6]. In the case of STRE, we simply calculate the relative frequencies of the triple (shop, day-of-week, time-slot). The set IS contains all frequent patterns appearing for the customer at least *minsup* times. We want all the patterns returned by Apriori and not only maximal and closed patterns because otherwise we could not consider useful patterns (e.g. the pattern {Cheese} in the example). Note that in our experiments we used *minsup* as a relative frequency.

Step #2 is the core of our contribution. It classifies each basket of the customer with the maximum matching frequent pattern. Its routine is expanded in Alg. 8. For every basket, we define set D as the set of all IS patterns that are completely contained in the pattern. Note that there might be no patterns included, depending on the *minsup* threshold choice. In this case we say that this basket can be only represented by itself, and its frequency is set to 1 (Steps #4). Otherwise we have to extract from D the most significant representative basket that we use to classify the basket. The cascade of “if” conditions selects the representative basket rb as the most significant if: *(i)* there is one representative basket larger than any other representative basket in D (i.e., it contains the absolute highest number of elements, Step #6); *(ii)* there is one basket with the highest support (Step #8); *(iii)* there is one basket with the lowest $lift$ (Step #10). In data mining, $lift$ tells us how much more than expected a given customer purchased a given product: $lift > 1$ means higher than expected, $lift < 1$ means lower than expected. We use the lowest $lift$ because being the least unexpected means to be more representative. In all cases, the representative basket rb is found and its frequency (RB_{rb}) is increased by 1. If it is impossible to reduce the set of included representative baskets to only one element, we classify the basket with all the remaining representative baskets, which are weighted one over the number of surviving representative baskets (Steps #11). Once we have the frequency of all representative baskets, we calculate BRE in Step #3 of Alg. 7.

Therefore, given a customer u, her Personal Data Model capturing her shopping behavior P_u, can be equipped with the shopping unpredictability indexes BRE and STRE $\langle bre_u, stre_u \rangle$. By oberving these indexes a customer can understand how much is systematic in her basket composition and in when and where she purchases.

8.1.3 Case Study

In this section, we deploy our indexes to analyze the relationship between the behavior of customers and their shopping sessions[1]. Before moving to the results, we provide our motivations for the required parameter *minsup* (see Alg. 7). For this specific study we wanted a common concept of minimum support among all the customers analyzed, therefore we did not use a personal strategy like those generally employed in this thesis. We tested different values for *minsup*, from 18% to 36%. Note that *minusp* influences the average pattern length we found. Higher *minsup* generates shorter, and less descriptive, patterns: the more elements a pattern has, the least likely it is to appear in full. Fig. 8.2 *(left)* depicts this effect. *minsup* also influences the distribution of BRE values. Higher *minsup* generates less patterns and therefore BRE tends to take lower values, as each pattern is a new symbol and more symbols require more bits to be encoded. Fig. 8.2 *(right)* depicts this effect. We chose *minsup*=24 as a good balance between the expressiveness of the detected patterns, and it does not skew the BRE distribution too much. Note that *minsup* has no effect on STRE, as for STRE we consider all possible triples (shop, day-of-week, time-slot) and we do not use any frequent itemset mining technique. In particular, in our experiments we selected day-of-week in {weekday, weekend}, and time-slot in {07:00-9:30, 09:30-12:00, 12:00-17:00, 17:00-19:30, 19:30-21:00} according to Fig. 8.1 *(d)*.

We calculate the BRE for all customers included in the dataset using *minsup* = 24%, Fig. 8.3 *(left)* depicts the distribution: a skewed bell shape, peaking at 0.79; where 80% of the customers take values between 0.62 and 0.84. The 10th and 90th percentiles are highlighted in green and purple, respectively. Customers beyond the 90th percentile are "casual", customers below the 10th percentile are "systematic", and the remaining customers are "standard". Fig. 8.3 *(right)* reports the STRE customer distribution. The distribution is a skewed bell shape, similar to the one observed for the BRE measure. The peak is now around 0.85, and 80% of the customers take values between 0.64 and 0.88. Also in this case, we report the 10th and 90th percentiles with green and purple lines.

[1] A dataset sample and the code to calculate BRE and STRE are available at https://goo.gl/UCqrUq

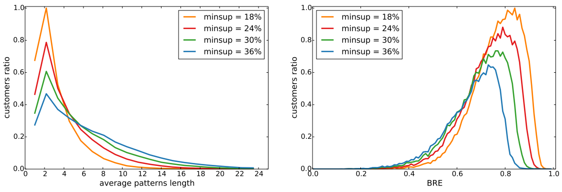


Figure 8.2: The effect of *minsup* on the length of the extracted patterns (left) and on the distribution of the BRE values (right).

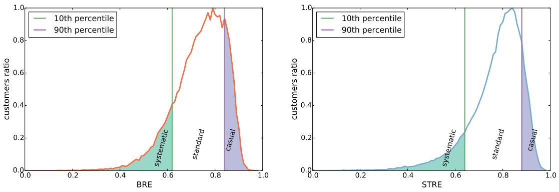


Figure 8.3: Distributions of the BRE and STRE measures in our dataset, with highlighted 10th and 90th percentiles.

Product	Sup	Product	Sup
Bananas	82.44	Fresh Eggs	64.08
Vine Tomatoes	74.22	Parsley	62.71
Sugar	72.04	Nectarines	62.55
Fennels	69.12	Green Tomatoes	62.49
Dark Zucchini	67.80	Fresh Eggs (Organic)	62.23
Bright Zucchini	67.37	Roma Tomatoes	61.49
Cherry Tomatoes	65.52	Melons	61.17

Table 8.1: The list of products of the systematic customers.

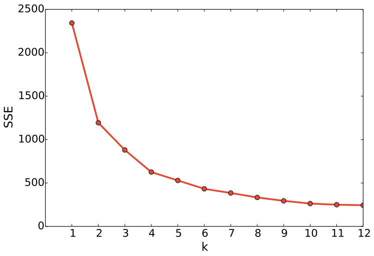


Figure 8.4: SSE for k-means runs.

Systematic Basket

We analyze here the products purchased by the systematic customers. We define a systematic customer as a customer who is below the 10th percentile either for the BRE or for the STRE unpredictability measures. Tab. 8.1 reports the list of the systematic products. We dropped from this list all the meaningless products such as discount coupons and the plastic shopping bag. From the list, we see that this selection includes mostly perishable products, from the fruit, vegetable and diary sectors. The only exception is sugar. It appears that the systematic customer's basket is characterized by very fresh products, that have a short shelf life and need to be purchased often.

Customer Classification

We now classify customers according to their observed $\langle bre_u, stre_u \rangle$ values. We represent each customer as a point in a two dimensional space. Her coordinates in this space are her BRE and STRE values. Then, we apply the k-means clustering algorithm [141] to detect clusters of customers in this space. The k-means algorithm requires to specify k, the number of clusters, or customer classes. The standard approach for determining k is to run k-means with varying ks (from 1 to 20 in our case), to calculate the Sum of Squared Errors (SSE) for each k and choose the highest k beyond which SSE does not improve significantly. In our case, we have k=5. Fig. 8.4 depicts the evolution of the SSE values.

For each detected cluster, k-means automatically detects the centroid, i.e., the most representative point of the cluster. If a point x belongs to cluster A, then the centroid of A is the closest centroid to x. The centroids are also representative of the cluster, as their BRE and STRE values are the averages of the cluster. In Tab. 8.2 we report the statistics of the five detected clusters. We can see that the cluster sizes are well balanced, where three clusters contain around 20% of customers as expected. The exceptions are the larger C cluster and the smaller E cluster. We also report the normalized BRE and STRE values

Cluster	Size	BRE	STRE
A	19.5%	0.45	0.57
B	17.3%	1.00	0.84
C	34.6%	0.50	1.00
D	21.7%	0.00	0.81
E	6.9%	0.17	0.00

Table 8.2: Statistics of the detected customer clusters: relative size, and normalized average BRE and STRE scores.

		STRE		
		Low	Medium	High
	High			B
BRE	Medium		A	C
	Low	E		D

Table 8.3: The relative positions of the detected clusters in the BRE-STRE space. Note the triangular structure.

of each cluster's centroid. The normalization simply rescales BRE and STRE such that the minimum of all centroids equals to zero and the maximum equals to one.

From these values, we can easily characterize the five clusters. To aid the understanding of our interpretation, we display a simplified representation of the relative position of the centroids in the two dimensional space in Tab. 8.3. We can see that each cluster can be characterized as follows according to BRE and STRE respectively: *A* medium BRE medium STRE; *B* high BRE high STRE; *C* medium BRE high STRE; *D* low BRE high STRE; *E* low BRE low STRE. From these results we can infer that the BRE-SPRE space has a triangular shape. Unpredictability in basket composition implies unpredictability also in the spatio-temporal dimension. On the other hand, unpredictability in the spatio-temporal dimension does not imply anything in the basket dimension (see cluster *D* for instance): the fact that we cannot predict when and where a customer will buy next time does not hinder us in predicting what products she is going to buy.

Before looking at more advanced statistics, we point out that the very regular customers, the ones characterized by both a low BRE and a low STRE, are the ones classified in cluster *E*. Cluster *E* is the smallest cluster, including the fewest number of customers, just below 7%. We conclude that the set of very regular customers is actually a large minority, at least in this supermarket chain. When projecting on one dimension, we see that the customers with regular basket are less than 29% (clusters *D* and *E*), while the spatio-temporal regular are still just 7%, due to the triangular shape of our space (they can be found only in cluster *E*). In fact, we can conclude that most of the customers are spatio-temporal irregular, but somewhat basket regular. The two largest clusters are *C* and *D* and while they both have high spatio-temporal entropy, they also have low or medium basket entropy. We can conclude that customers are more predictable in what they buy, rather than in when and where they perform their shopping sessions.

We can now describe how the behavioral differences of the customers classified in the various clusters impact the profitability. For each cluster, we can calculate the total and per capita expenditures generated by customers classified in it, and we can calculate the total number of baskets and the per capita average. All these statistics are reported in Fig. 8.5. The most remarkable feature of this Figure is that it shows cluster *E* scoring the highest in expenditure per capita. We can calculate each cluster's leverage, that is the ratio between revenue share and relative size of the cluster. For *E*, since it includes 6.9% of customers and the total revenue is 178.41 million euros, the leverage equals to (15.84/178.41)/0.069=1.29. The second best is cluster *D* with a leverage of 1.15, while clusters *C* and *B* lag behind with a leverage of 0.9 and 0.79 respectively. This fact is hinting that the regularity might have a connection to her expenditure.

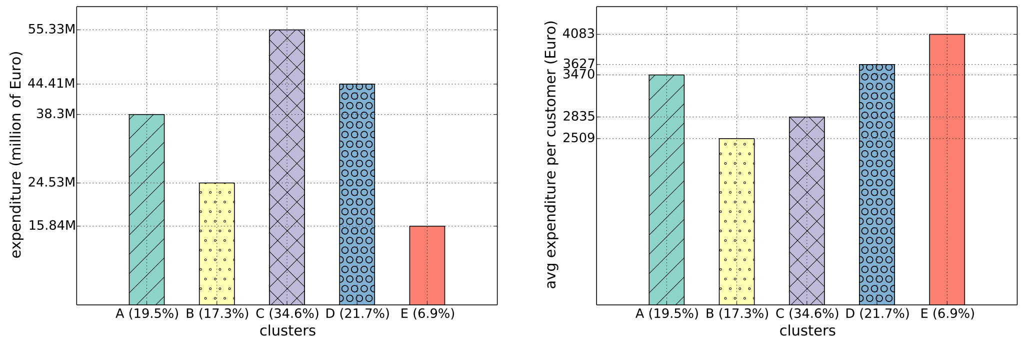


Figure 8.5: The characteristics of customers belonging to the different clusters: on the left the customer expenditures, totals on the left and per capita on the right.

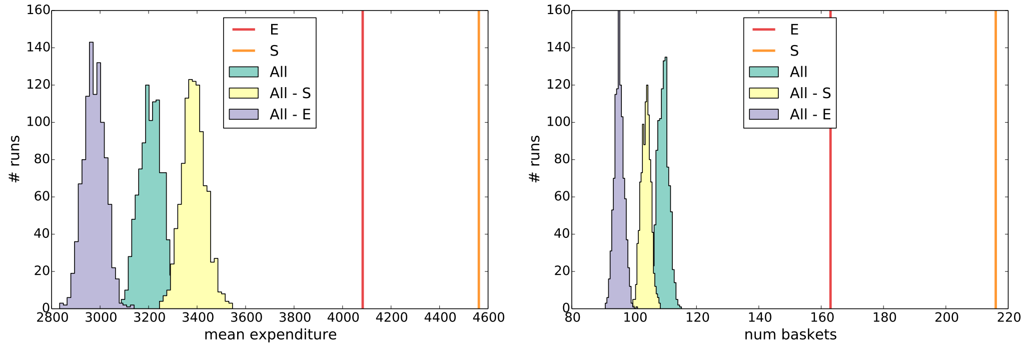


Figure 8.6: The distributions of the average expenditure (left) and the average number of baskets (right) for null models "All", "All except systematic" ("All - S") and "All except E" ("All - E").

Looking at the broader picture, we see that there is a negative relationship between irregularity and per capita expenditure. We sort clusters from the most to the least irregular (by summing their BRE and STRE centroid values): $B \rightarrow C \rightarrow A \rightarrow D \rightarrow E$. We obtain a reverse order with the average per capita expenditure (see Fig. 8.5): $E \rightarrow D \rightarrow A \rightarrow C \rightarrow B$. We already saw that most customers are irregular in their spatio-temporal patterns. From the Figure, we also see that spatio-temporal irregular customers visit the stores more sporadically. The average revenue generated per customer is higher for customers with low behavioral entropy. Cluster E is only two fifths in size of cluster B, but generates almost two thirds of cluster B's total revenues. However, the revenue from regular customers is low in absolute terms. The observed patterns have profound implications both for individual customers and for the supermarket company.

Validation

We now turn to some sanity checks to understand the significance of cluster E's observed profitability. We start our result validation from the systematic basket composition. For each customer, we calculate the $lift$ measure for the products in Tab. 8.1. We then count how many systematic customers have $lift>1$ for each of these products and we compare the three customer classes: systematic, standard and casual. We see that, on average, for each product there are 16% (st.dev. 5%) more systematic customers with $lift>1$ than casual ones, and 9% (st. dev. 4%) more systematic than standard.

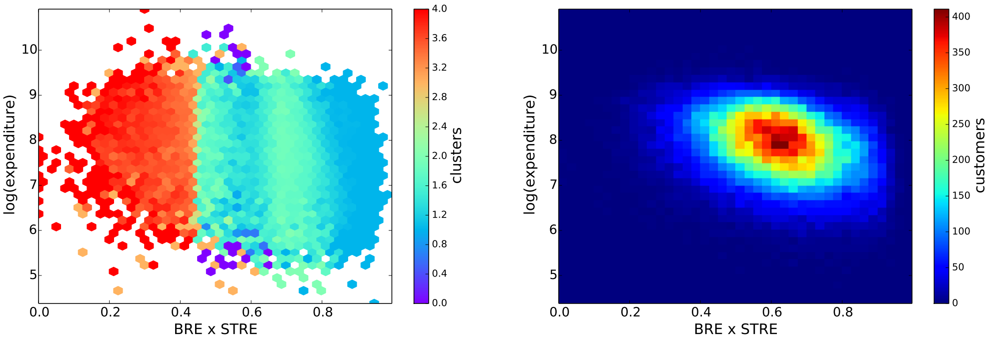


Figure 8.7: Heatmaps depicting the relationship between the combined behavioral entropy (x axis) and the expenditure level (y axis, in log scale). We have both the average cluster composition of the cell (left, from $0 = A$, to $4 = E$) and the simple count of the number of customers (right).

Now we focus on understanding if cluster E is really the most profitable per capita or if its expenditure level is not significantly different from a random occurrence. We perform two tests: a null model validation and a targeted model validation. Finally, we perform a last validation abstracting from the detected clusters and testing the direct connection between behavioral entropy and personal expenditure. In the null model validation we want to explain the expenditure level and the number of baskets of the customers belonging to cluster E. We create some random E clusters with different characteristics and we observe their expected characteristics. We define three models called “All”, “All except systematic” (“All - S”) and “All except E” (“All - E”). We run each model a thousand times and we plot the distribution of their expenditure levels and the number of baskets in Fig. 8.6. The red band in Fig. 8.6 is the observed E value. The “All” model constructs a purely random E cluster. We extract uniformly at random 7% of the customers in our data and we calculate their average expenditure and their average number of baskets. Fig. 8.6 reports that this model has an expected expenditure of 3,200 euros, that is slightly more than three quarters of the actual E expenditures. The “All except systematic” model constructs a random E cluster by (randomly) selecting customers outside the “systematic” cut, i.e., all customers that have BRE and STRE values higher than the 10th percentile. By restricting to these customers we attempt to counter the argument that it is the BRE and STRE values driving the expenditure and not other common factors of customers included in cluster E. However, we obtain again a lower expected expenditure: 3,400 euros or just 83% of the actual expenditure of cluster E. Finally, with the “All except E” we construct a random E cluster by selecting customers at random from the pool of customers that are not part of the original cluster E. In this model we investigate if it is likely to find a random composition of customers outside cluster E that are characterized by higher expenditure levels than the members of cluster E. This is the model that performs the worst, even worse than the “All” model, proving that “All” model’s performance was actually driven by E cluster members. In “All except E”, the expected expenditure is just below 3,000 euros. The number of baskets of cluster E members is impossible to match too. In this case, the “All” performs better than “All except systematic”, hinting that the number of baskets is more dependent on the behavioral entropy than the expenditure level.

Moving to targeted model validations, we define two: one based on expenditure and one based on the behavioral entropy. Differently from before, we are not composing a random E model, but we are sorting all customers in descending order of the chosen measure. Start-

	Dependent variable:					
	log(expenditure)			log(baskets)		
	(1)	(2)	(3)	(4)	(5)	(6)
BRE	-1.123^{***}			-1.124^{***}		
	(0.0021)			(0.0020)		
STRE		-1.152^{***}			-1.738^{***}	
		(0.0026)			(0.0024)	
BRE * STRE			-1.269^{***}			-1.492^{***}
			(0.0019)			(0.0018)
constant	8.738^{***}	8.754^{***}	8.635^{***}	5.360^{***}	5.833^{***}	5.394^{***}
	(0.017)	(0.020)	(0.012)	(0.016)	(0.019)	(0.011)
Observations	56,448	56,448	56,448	56,448	56,448	56,448
R^2	0.048	0.034	0.071	0.055	0.088	0.112
Adjusted R^2	0.048	0.034	0.071	0.055	0.088	0.112
Residual Std. Error	0.676	0.681	0.668	0.631	0.620	0.611
F Statistic	$2,851,457^{***}$	$1,970,123^{***}$	$4,322,123^{***}$	$3,277,555^{***}$	$5,419,480^{***}$	$7,131,251^{***}$
						Note: *p<0.1; **p<0.05; ***p<0.01

Table 8.4: BRE and STRE effect in predicting the total expenditure level (models 1 to 3) and the number of baskets (models 4 to 6) of a customer for the year 2012. This is a standard OLS model. The R^2 can be interpreted as the square of the correlation coefficient of the variables.

ing with expenditure, we collect the 7% top-spending customers and we count how many of them are classified in cluster E. The result is 14.66%, meaning that cluster E is represented in the top spending customers twice as much as its size would suggest. This confirms the strong relation between E and high expenditure levels. If we include also the cluster characterized by the second most regular customers, cluster D, the share goes up to 38.7%. The second targeted model involves selecting the customers in the “systematic" cut regardless the cluster in which they were classified. Their expenditure levels are very high, higher than the members of cluster E. This hypothetical super-systematic cluster has an expected expenditure level of almost 4,600 euros, as the orange band depicts it in Fig. 8.6.

For our last validation step we abstract from the cluster division, to observe the direct relationship between a customer's behavioral entropy and her profitability for the retail company. This is done by plotting the behavioral entropy of a single customer against her expenditure level. Fig. 8.7 shows two variants of this plot. In both cases we have a heatmap that groups the customers in a given interval of expenditures and of entropy. The x axis combines BRE and STRE by multiplying them. The y axis reports the logarithm of the expenditure level. On the left of Fig. 8.7, we have the average cluster composition of the cell. To calculate this average each cluster is mapped to an integer. The heatmap has a left to right gradient, where the lowest values on the x axis correspond to highest clusters (D and E). The heatmap contains a negative relationship between combined entropy and expenditure. To better highlight this negative relationship, on the right of Fig. 8.7 we use a different coloring logic for the heatmap. Instead of reporting the cluster composition, we color the cell according to its number of customers. Blue means few or no customer, red means a high concentration of customers. We can see that now the negative relationship is more clear: the densely populated cells show a downward pattern. To quantify objectively the size of the effect depicted in Fig. 8.7, we set up a model where we attempt to predict the logarithm of the customer's expenditure (or baskets) by using her BRE and STRE level. First we test the two measures separately, then we create a global measure by multiplying them. Tab. 8.4 reports the result of this regression. Both BRE and STRE have significant effects, with comparable levels. We are using a log-linear space, thus a coefficient of -1.123 means that increasing the entropy level by 1 is associated with an expected expenditure drop of almost a third ($e^{-1.123} \sim 0.325$)[2]. An hypothetical perfectly predictable customer (entropy = 0) would make three times as many profits for the company than a hypothetical completely unpredictable customer (entropy = 1). Combining BRE and STRE together, the effect almost reaches a fourfold increase ($e^{-1.269} \sim 0.281$). The effect is stronger if we predict the number of baskets instead of the expenditure level. The unit decrease in combined entropy is associated with almost a fivefold increase in the number of baskets purchased ($e^{-1.492} \sim 0.225$).

8.1.4 Conclusion

We have proposed novel personal indicators of systematic shopping behavior for enriching the model in the Personal Data Store and we have investigated the effects of customer predictability in the retail market scenario. We have estimated how much the behavior of a customer is predictable along two dimensions: basket composition and spatio-temporal, i.e., where and when a customer purchases the products she needs. We have shown that it is possible to divide the customers of the PDE at collective level into systematic and non-systematic, and even define five distinct classes. The systematic customers have been showed to be a minority, but their per capita expenditure and the expected number of baskets is much higher than average. Our individual measures have proved to be significant predictors of the value of a customer for the supermarket and point out that nudging customers to be regular could be an interesting strategy to increase revenues.

8.2 A Personal Data Model for Musical Preferences

Since music is a pervasive dimension of our life, and due to the abundance of online data sources like Spotify, iTunes and Last.Fm, we propose a Personal Data Model able to capture the characteristics and the systematic patterns which are present in our musical listening behavior. We call this model *Personal Listening Data Model* (*PLDM*) [134]. The PLDM is built on a set of personal listening represented by an abstract data type taken as input. A listening is formed by the song listened, the artist of the song, the album, the genre and the listening time-stamp. According to Personal Data Analytics, the PLDM is a particular type of the Personal Data Model designed for musical listening data.

The PLDM contains some *indicators* extracted from the listening features that summarize the listener and explain her level of repetitiveness in the listening. Moreover, the PLDM is formed by some listening *patterns* extracted from the listening *frequencies*. These patterns are the top listened genre, artist, album etc. and the most representative preferences. In addition, the PLDM contains the frequent listening *sequences*. Those are the typical repetitions followed by the user during a listening session.

8.2.1 Personal Listening Data Model

In this section we formally describe the *Personal Listening Data Model*. By applying the following definitions and functions of Personal Data Analytics it is possible to build for each user a listening profile P_u giving a picture of her habits in terms of listening.

[2] See http://goo.gl/rD6YTy for an explanation of our coefficient.

Figure 8.8: A listening $l = \{\langle \mathit{time\text{-}stamp}, \mathit{song}, \mathit{artist}, \mathit{album}, \mathit{genre} \rangle\}$ is a tuple formed by the *time-stamp* indicating when the listening occurred, the *song* listened, the *artist* which sings the song, the *album* the song belongs to, and the *genre* of the artist.

Definition 9 (Listening)**.** *Given a user* u*, we define* $L_u = \{\langle$*time-stamp*, *song*, *artist*, *album*, *genre*$\rangle\}$ *as the set of listening performed by* u*.*

L_u corresponds to H_u w.r.t. listening data. Since a song can belong to more than a genre and can be played by more than an artist, each listening l (Fig. 8.8) is an abstraction of a real listening. However, we can assume this abstraction without losing in generality.

From the set of listening L_u, for each user we can extract the set of songs S_u, artists A_u, albums B_u and genres G_u. For example, $A_u{=}\{\mathit{artist} | \langle \cdot, \cdot, \mathit{artist}, \cdot, \cdot \rangle \in L_u\}$, $G_u{=}\{\mathit{genre} | \langle \cdot, \cdot, \cdot, \cdot, \mathit{genre} \rangle \in L_u\}$, etc. Their sizes ($| \cdot |$) are valuable *indicators*.

The user behavior can be summarized through frequency dictionaries indicating the support (i.e. relative number of occurrences) of the listening features.

Definition 10 (Support)**.** *The* support *function returns the frequency dictionary as a set of couples (item, support) where the* support *of an* item *is obtained as the number of occurring items on the number of listening.*

$$\mathit{sup}(X, L) = \{(x, y) | y = |Y|/|L| \wedge x \in X \wedge Y \subseteq L s.t. \forall l \in Y, x \in l\} \tag{8.1}$$

We define the following frequency dictionaries: $s_u{=}\mathit{sup}(S_u, L_u)$, $a_u{=}\mathit{sup}(A_u, L_u)$, $b_u{=}\mathit{sup}(B_u, L_u)$, $g_u{=}\mathit{sup}(G_u, L_u)$, $d_u{=}\mathit{sup}(D, L_u)$ and $t_u{=}\mathit{sup}(T, L_u)$ where $D{=}\{\mathit{mon}, \mathit{tue}, \mathit{wed}, \mathit{thu}, \mathit{fri}, \mathit{sat}, \mathit{sun}\}$ contains the days of weeks, and $T{=}\{(2\text{-}8], (8\text{-}12], (12\text{-}15], (15\text{-}18], (18\text{-}22], (22\text{-}2]\}$ contains the time slots of the day.

These dictionaries can be exploited to extract indicators and patterns.

Definition 11 (Entropy)**.** *Given dictionary* $X = \{(x_1, y_1), \ldots, (x_n, y_n)\}$*, the* entropy *function returns the normalized entropy defined as*

$$\mathit{entropy}(X) = \frac{-\sum_{i=1}^{n} \mathrm{P}(y_i) \log_2 \mathrm{P}(y_i)}{\log_2 n} \quad \in [0, 1] \tag{8.2}$$

The entropy tends to 0 when the user behavior is systematic, tends to 1 when the behavior is not predictable. These indicators are similar to those related with shopping behavior described in the previous Section and in [129]. We define the entropy for songs, artists, albums, genres, days and time-slots as $e_{s_u}{=}\mathit{entropy}(s_u)$, $e_{a_u}{=}\mathit{entropy}(a_u)$, $e_{b_u}{=}\mathit{entropy}(b_u)$, $e_{g_u}{=}\mathit{entropy}(g_u)$, $e_{d_u}{=}\mathit{entropy}(d_u)$ and $e_{t_u}{=}\mathit{entropy}(t_u)$.

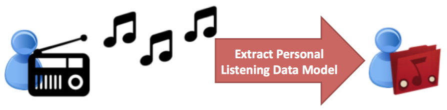


Figure 8.9: The raw listening of a user L_u can be turned into a Personal Listening Data Store P_u extracting the songs S_u, artists A_u, albums B_u and genres G_u and by applying to them the functions *sup*, *top*, *repr*, *entropy*, *getseq* and *freqseq*.

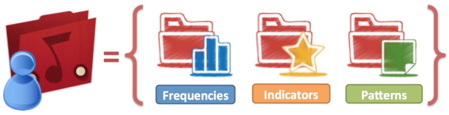


Figure 8.10: The PLDM is formed by *indicators* ($|L_u|$, $|S_u|$, $|A_u|$, $|B_u|$, $|G_u|$, and entropy values), by *frequencies* (the support dictionaries) and by *patterns* (most listened preference, most representative preferences).

The simplest pattern we consider is the most listened song, artist, genre, etc.

Definition 12 (Top)**.** *Given dictionary $X = \{(x_1, y_1), \dots, (x_n, y_n)\}$, the* top *function returns the most supported item. It is defined as:*

$$top(X) = \underset{(x,y)\in X}{argmax}(y) \tag{8.3}$$

We define the most listened songs, artists, albums and genres as $\hat{s_u}{=}top(s_u)$, $\hat{a_u}{=}top(a_u)$, $\hat{b_u}{=}top(b_u)$ and $\hat{g_u}{=}top(g_u)$, respectively.

Moreover, we want to consider for each user the set of most representative, i.e. significantly most listened, subsets of artists, albums and genres.

Definition 13 (Repr)**.** *Given dictionary $X = \{(x_1, y_1), \dots, (x_n, y_n)\}$, the* repr *function returns the most representative supported items. It is defined as:*

$$repr(X) = \underset{(x,y)\in X}{knee}(y) = \underset{(x,y)\in X^*, y'\in X'}{argmax}(|y - y'|) \tag{8.4}$$

where X^* is X sorted with respect to the supports y, $X' = \{y' | y' = mx' + n\}$ with $m{=}(max(sup(X)) - min(sup(X)))/|X|$ and $n{=}min(sup(X))$.

The method $repr(X)$ returns a set of preferences with a support higher than the support of most of the other listening. For example if $g_u{=}\{(rock, 0.4), (pop, 0.3), (folk, 0.1), (classic, 0.1), (house, 0.1)\}$, $repr(g_u)$ returns $\{(rock, 0.4), (pop, 0.3)\}$.

This result is achieved by employing the *knee* method [275]. Given a dictionary X of pairs composed by item x and support y, the *knee* method sorts the pairs (x_i, y_i) according to the supports generating X^*. Then, it selects the point x^*_k on the support curve X^* which has the maximum distance $|y^*_k - y'_k|$ with the correspondent point x'_k in X', where X' is the straight line passing through the minimum and the maximum point of the curve described by X^*. In this way the *knee* x^*_k is different for each user because it is driven by personal data. Finally, the method returns the pairs with a support greater or equal than the support y_k of the knee x_k. We define the most representative songs, artists, albums and genres as $\tilde{s_u}{=}repr(s_u)$, $\tilde{a_u}{=}repr(a_u)$, $\tilde{b_u}{=}repr(b_u)$ and $\tilde{g_u}{=}repr(g_u)$, respectively. Obviously, we have $\hat{g_u}{\subseteq}\tilde{g_u}{\subseteq}g_u$ that holds also for songs, albums and artists.

By applying the definitions described above on the user listening L_u we can turn the raw listening data of a user into a complex personal data structure (see Fig. 8.9) that we call *Personal Listening Data Model* (PLDM). The PLDM characterizes the listening behavior of a user by means of its *indicators*, *frequencies*, and *patterns* (see Fig. 8.10).

Definition 14 (Personal Listening Data Model)**.** *Given the listening L_u of a user u we define the user* personal listening data model *as*

$$\begin{aligned} P_u = \langle |L_u|, |S_u|, |A_u|, |B_u|, |G_u|, \quad & e_{s_u}, e_{a_u}, e_{b_u}, e_{g_u}, e_{d_u}, e_{t_u}, && \mathit{indicators} \\ s_u, a_u, b_u, g_u, d_u, t_u, & && \mathit{frequencies} \\ \hat{s_u}, \hat{a_u}, \hat{b_u}, \hat{g_u}, \quad & \tilde{s_u}, \tilde{a_u}, \tilde{b_u}, \tilde{g_u} \rangle && \mathit{patterns} \end{aligned}$$

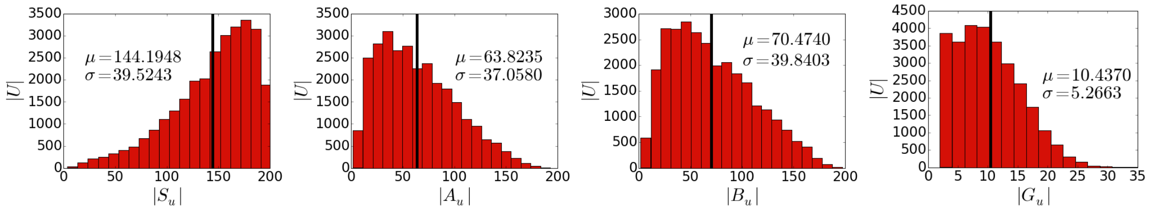


Figure 8.11: Distributions of the number of songs $|S_u|$, artists $|A_u|$, albums $|B_u|$ and genres $|G_u|$ respectively. The black vertical lines highlight the means.

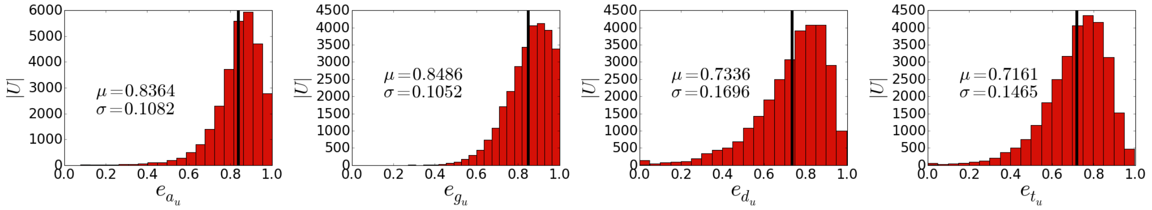


Figure 8.12: Distributions of entropy for artists e_{a_u}, genre e_{g_u}, day of week e_{d_u} and time of day e_{t_u} respectively. The black vertical lines highlight the means.

8.2.2 Case Study

In this section we show the analytical benefits of the application of the PLDMs on the data extracted from *Last.Fm* (see Section 6.4). Given the listening L_u, we calculated the PLDM P_u for each user $u \in U$. The obtained PLDMs allowed us to estimate how the Last.Fm audience is segmented. Another finding is that the musical profile of each user is best outlined using a limited set of distinct musical preferences, but not by a unique liking.

Indicators Analysis

The first analysis we report is related to the *indicators* of the PLDMs $\{P_u\}$ extracted. In Fig. 8.11 are reported the distributions of the number of users which have listened a certain number of songs $|S_u|$, artists $|A_u|$, albums $|B_u|$ and genres $|G_u|$. The first distribution is right-skewed, i.e., most of the users have listened to about 140 songs. This implies that some tracks were listened more than once. On the other hand, the other distributions are left-skewed: a typical user listens about 60 artists, 70 albums and 10 genres.

Fig. 8.12 depicts the distributions of the entropy.It emerges that users are much more systematic with respect to the listening time (day of week and time of the day) than with respect to what they listen. This behavior is in opposition to what happens in shopping [129]. Since the artist and genre entropy are right-skewed, it seems that most of the users are not very predictable with respect to the genre or to the artist. This is a first clue that is very unlikely that exists a unique prevalence towards a unique artist or genre.

Fig. 8.13 (left) shows the heat-map of the correlations among the indicators. Some of them like $|A_u|$, $|B_u|$ and $|G_u|$ are highly correlated[3] ($cor(|A_u|, |B_u|)$=0.86, $cor(|G_u|, |B_u|$= 0.64)): the higher the number of artists or genres, the higher the number of albums listened. Other interesting correlations are $cor(|B_u|, e_{g_u})= -0.33$ and $cor(|B_u|, e_{a_u})$=0.55. Their

[3] The *p-value* is zero (or smaller than 0.000001) for all the correlations reported.

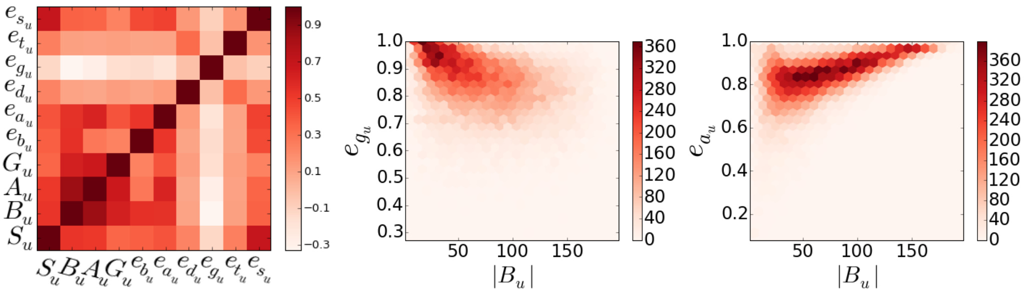


Figure 8.13: Correlation matrix (*left*)): the darker the more positively correlated, the lighter the more negatively correlated. Scatter density plots of number of albums $|B_u|$ and genre entropy e_{g_u} (*center*) and number of albums $|B_u|$ and artists entropy e_{a_u} (*right*).

	e_{t_u}	e_{d_u}	e_{s_u}	e_{a_u}	e_{b_u}	e_{g_u}	size
A	0.8067	0.8442	0.9744	0.8591	0.8794	0.8461	0.44
B	0.7092	0.7234	0.9305	0.7001	0.6732	0.8862	0.13
C	0.4672	0.3366	0.9254	0.7438	0.7717	0.8751	0.06
D	0.5568	0.7687	0.9748	0.8666	0.8855	0.8383	0.19
E	0.7484	0.5624	0.9775	0.8739	0.8918	0.8306	0.19

Table 8.5: Centroids for the entropy and size of the clusters extracted.

density scatter plots are reported in Fig. 8.13 *(center, right)*. They tell us that the higher the number of albums listened, the lower the variability with respect to the genre and the higher the variability with respect to the artists. From this result we understand that a user listening to many different albums narrows her musical preferences toward a restricted set of genres, and that she explores these genres by listening various artists of this genre and not having a clear preference among these artists.

Segmentation Analysis

The second analysis we propose investigates the existence of different groups of listeners with respect to their *indicators* in the PLDMs $\{P_u\}$. We applied the clustering algorithm K-Means [275] by varying the number of clusters $k \in [2, 30]$. By observing the trend of the sum of squared error [97] we decided to select 5 as the number of clusters. In Fig. 8.14 are described the radar charts representing the centroids while in Table 8.5 are reported the value of the centroids and the size of the clusters.

The biggest cluster is A. It contains the majority of the listeners. It seems that these listeners use the web service without a specific listening schema, i.e., they reproduce the tracks using the random function. However, a peculiarity of these users, is that they are more repetitive than users in the other clusters with respect to the genres.

In opposition with A, users in clusters B and C do not have a set of genres which is clearly preferred on top of the others, but they are the most systematic users in terms of albums and artists listened. This means that they like a concise set of artists regardless of their genre and they keep listening only them. The main difference between B and C is that users of cluster B are the most systematic in terms of albums and artists, while those of clusters C are the most regular with respect listening in specific days and time slots.

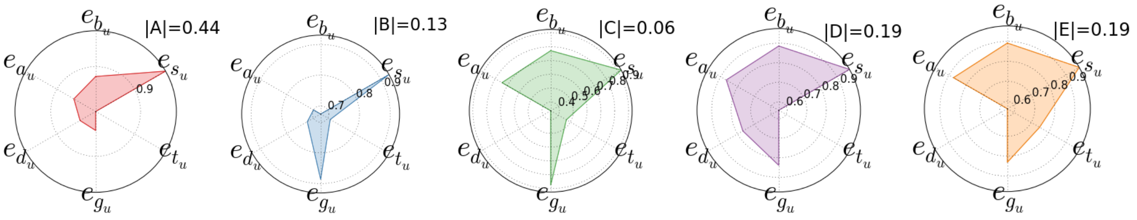


Figure 8.14: Radar charts for the centroids of the clusters extracted on the PMDLs.

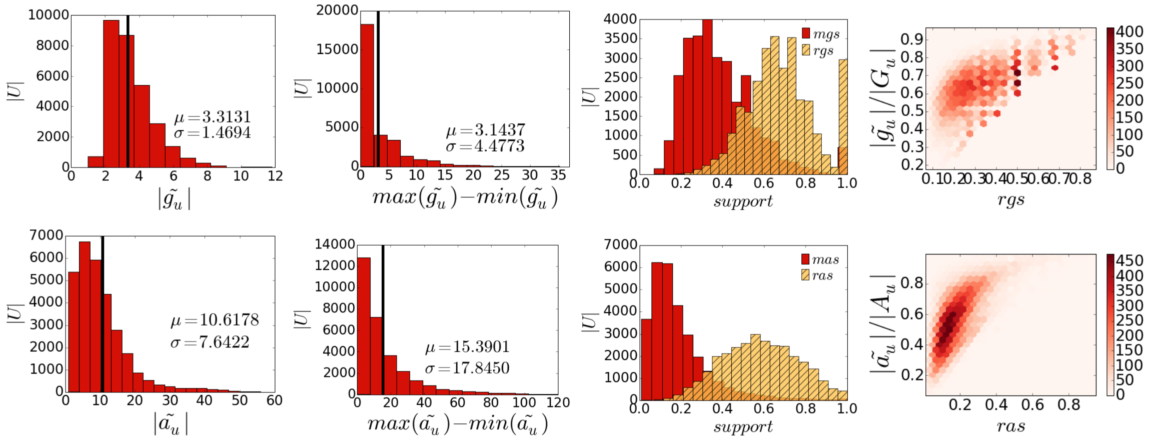


Figure 8.15: Frequencies analysis for genre (top row) and artist (bottom row). *First column*: distribution of number of users w.r.t the number of representative preferences. *Second column*: distribution of number of users w.r.t the maximum difference in frequencies between the listening preference. *Third column*: distribution of number of users w.r.t the support given by the representative preferences. *Last column*: density scatter plot between the representative preferences support and the ratio of their number on the number of all the possible artists or genres.

Finally, users in clusters D and E are similar to those in cluster A with respect to the level of repetitiveness of listening of genres, artists, and albums. On the other hand, how is highlighted by the last two radars in Fig. 8.14, they are complementary with respect to the day of the week and to time of listening. Users in cluster D do not have a specific day of the week but use the service constantly at the same time (e.g. during gym session or during specific working areas). Conversely, users in cluster E do not have a specific time slot but use the service periodically in specific days of the week (e.g. during the weekend).

We can conclude that exists a clear distinction among different groups of listeners. From the clustering information originated from the Personal Listening Data Model, a user could learn that is focusing too much on a certain genre or on certain artists and that is not exploring what is outside her “musical confidence zone”.

Frequency Analysis

In this section we exploit the knowledge of the frequency vectors to demonstrate that the most listened genre, album, and artist considered alone do not represent properly the preferences of the users. To this aim we look at the frequency vectors a_u, g_u, the top listened $\hat{a_u}$, $\hat{g_u}$, and the most representative $\tilde{a_u}$, $\tilde{g_u}$. In the following discussion we will refer $\tilde{a_u}$ and $\tilde{g_u}$ equivalently as $\tilde{x}$ and to the artists and genres contained in such sets as *preferences*.

	$\{\hat{g_u}\}$	sup	$\{\hat{a_u}\}$	sup	$\{\tilde{g_u}\}$	sup	$\{\tilde{a_u}\}$	**sup**
1	Rock	53.86	The Beatles	0.75	Rock	13.41	David Bowie	0.29
2	Pop	19.64	David Bowie	0.72	Pop	9.73	Arctic Monkeys	0.26
3	Hip Hop	5.05	Kanye West	0.56	Hip Hop	5.16	Radiohead	0.24
4	Electronic	2.21	Arctic Monkeys	0.54	Inide Rock	4.39	Rihanna	0.24
5	Folk	2.03	Rihanna	0.51	Folk	4.31	Coldplay	0.23
6	Punk	1.74	Lady Gaga	0.48	Electronic	4.26	The Beatles	0.22
7	Inide Rock	1.65	Taylor Swift	0.47	Punk	4.07	Kanye West	0.21
8	Dubstep	0.90	Radiohead	0.43	House	2.63	Muse	0.19
9	House	0.85	Muse	0.38	R&B	2.53	Florence	0.19
10	Metal	0.84	Daft Punk	0.37	Emo	2.11	Lady Gaga	0.19

Table 8.6: Top ten of top listened ($\{\hat{g_u}\}$, $\{\hat{a_u}\}$) and most representative ($\{\tilde{g_u}\}$, $\{\tilde{a_u}\}$).

Fig. 8.15 reports the results for genre (top row) and artist (bottom row).The first column shows the distribution of the number of users with respect to the number of representative genres $|\tilde{g_u}|$ and artists $|\tilde{a_u}|$. In both cases the smallest value is larger than 1 indicating that each user has more than a preference. On the other hand, a large part of all the genres and artists listened are removed when passing from x to $\tilde{x}$. Indeed, the mean for the genres decreases from 10 to 3, the mean for the artist diminishes from 60 to 10.

The second column (Fig. 8.15) illustrates the distribution of the number of users with respect to the maximum difference in frequencies between the listening preference obtained as $max(\tilde{x}){-}min(\tilde{x})$. For both features the mode of this value is close to zero. This proofs that the highest preferences are similar in terms of listening for the majority of the users.

The third column shows the distributions of the users with respect to the most listened artist support, $mas{=}v$ s.t. $(a,v){=}\hat{a_u}$, and most listened genre support, $mgs{=}v$ s.t. (g,v) $=\hat{g_u}$, and the representative artist support, $ras{=}sum(v|(a,v){\in}\tilde{a_u})$, and representative genre support, $rgs{=}sum(v|(g,v){\in}\tilde{g_u})$. It is evident the increase of the support when not only the top but also all the representative preferences are considered.

The last column reports a density scatter plot of the representative preferences support (rgs and ras) and the ratio of their size, i.e., $|\tilde{a_u}|/|A_u|$ and $|\tilde{g_u}|/|G_u|$ respectively. Since the higher concentration of points tends to be ∼0.2 with respect to the x-axis and ∼0.5 with respect to the y-axis, we have that for most of the users it is sufficient a limited number of preferences (but more than one) to reach a very high level of support. This concludes that each user can be described by few preferences that highly characterize her.

Finally, it is interesting to observe how the total support of the users and consequently the ranks of the top ten artists and genres change when the preferences in $|\tilde{g_u}|$ and $|\tilde{a_u}|$ are considered instead of those in $|\hat{g_u}|$ and $|\hat{a_u}|$. We report in Table 8.6 the top ten of the top listened genres and artists and the top ten of the most representative genres and artists with the users support, i.e., the percentage of users having that genre or artist as $\hat{g_u}$ or $\hat{a_u}$, and $\tilde{g_u}$ or $\tilde{a_u}$. We can notice how for the two most listened genres (rock and pop) there is a significant drop in the total support, vice-versa the other genres gain levels of support. The overall rank in the genre top ten is not modified very much. On the other hand, a complete new rank appears for the artists with a clear redistribution of the support out of the top ten. This last result is another proof that user's preferences are systematic but they are not towards a unique genre or artist, while they are towards groups of preferences.

Storage Analysis

To enhance the portability of the PLDM, we report in Fig. 8.16 the boxplots of the storage occupancy of the data model PLDMs (left) and for the raw listening (right). The storage

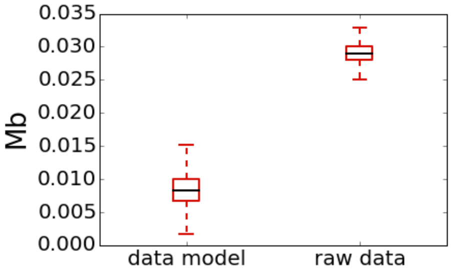


Figure 8.16: Boxplot of the data storage in MegaByte for the data model and for the raw data.

required by the data model is typically one third of the storage required by the raw data. Moreover, the storage space of the data model will not grow very much when storing more listening because the number of possible genres, artists, albums, songs is limited, while the number of listening grows continuously. Thus, an average storage of $0.01Mb$ together with a computational time of max 5 sec per user, guarantees that the PLDM could be calculated and stored individually without the need of a central service.

8.2.3 Conclusion

We have presented the Personal Listening Data Model (PLDM). The PLDM is a Personal Data Model specifically designed to deal with musical preferences. It is formed by *indicators* of the musical behavior, listening *patterns* and vectors containing the listening *frequencies*. By employing the PLDM on a set of 30k Last.Fm users we have shown how the indicators of PLDM can be exploited to produce a users segmentation able to discriminate between different groups of listeners. Moreover, the patterns and frequency vectors of the PLDM have been used to prove that information like the most listened genre or artist are not enough to represent the musical preferences of a user.

8.3 Towards a Personal Automatic Spatio-Temporal Agenda

As reported in Section 2, in the literature exists a various and large set of patterns and indicators to describe the personal mobility of a user. However, none of these models entirely consider all the dimension of human mobility. For instance, large attention is generally provided to movements, but also the spatio-temporal presence in certain places provides a worth piece of information. In this section we provide a preliminary description of a Personal Data Model for a mobility profile P_u which is able to summarize and characterize entirely the typical day of a user in terms of mobility. We call this model for Personal Data Analytics *Personal Mobility Data Model (PMDM)* or *Personal Agenda*. The Personal Agenda is a sort of container for novel and existing behavioral models and indicators, where all the patterns are in line with the personal perspective and are extracted by employing personal parameter-free and autofocus algorithms. Parts of this model are used in the various work described in the next part of this thesis.

8.3.1 Personal Mobility Data Model

The aim of the Personal Data Model we propose is to capture the personal mobility *agenda* of a user. Therefore, the model does not consider only the moments or the places visited, but it captures the systematic presences of the user during the day. We start by defining the basic data type of the model and from which the mobility patterns are extracted.

Definition 15 (Trajectory). *A* trajectory *is a sequence of spatio-temporal points* $a = \{(x_0, y_0, t_0), \dots, (x_{n-1}, y_{n-1}, t_{n-1})\}$ *where the spatial points* (x_i, y_i) *are sorted by increasing time* t_i*, i.e.,* $\forall\ 0 \leq i < n$ *we have that* $t_i < t_{i+1}$.

Given a point $p = (x, y, t)$, we refer to its components with $p.x$, $p.y$ and $p.t$ respectively. Each point (x, y) represents GPS coordinates expressed as longitude and latitude, while t represents the time-stamp. Given a trajectory a we refer to a particular point i in a with $a(i)$. Consequently, $a(0)$ refers to the start point, while $a(n-1)$ refers to the end point. Moreover we define $start(a) = a(0)$ and $end(a) = a(n-1)$ as the functions that given a trajectory, returns the first and last point respectively.

Definition 16 (Individual Mobility History). *Given a user u, her* individual mobility history $H_u^{t,t'} = \{a_0, \dots,\ a_{n-1}\}$ *is the set of the trajectories traveled in the time window* $[t, t')$*, i.e.,* $\forall\ a \in H_u$ *we have that* $t \leq start(a).t < end(a).t < t'$.

Definition 17 (Stops Observations). *Given a user u, her* stops $S_u^{t,t'} = \{s_0, \dots, s_{n-1}\}$ *is the set of points corresponding to the start or end of a trajectory, i.e.,* $\forall\ s \in S_u^{t,t'}$ *we have that* $\exists a \in H_u^{t,t'}$ *s.t.* $s = start(a)\ \vee s = end(s)$.

We define with $getStops(H_u) = S_u$ the function that takes as input the individual mobility history and returns the set of stops. The locations associated with a set of stops basically group the latter into partitions that define the places (or areas) they cover. E.g. a location can be home or work and the stops belonging to it are the parking lots used.

Definition 18 (Locations). *Given a user u and her stops* S_u*, her* locations $\mathcal{L}_u = \{L_0, \dots, L_{k-1}\}$ *is a partitioning of* S_u *into disjoint sets:* $\forall\ L \in \mathcal{L}_u, L \subset S_u$, $\bigcup_{L \in \mathcal{L}_u} = S_u$ *and* $L, L' \in \mathcal{L}_u \wedge L \neq L' \Rightarrow L \cap L' = \emptyset$*, where* $\forall\ L \in \mathcal{L}_u, L = \{s_0, \dots, s_m\}$*, s.t.* $\sum_{i=0}^{k-1} |L_i| = |S_u|$.

We define with $getLoc(S_u) = \mathcal{L}_u$ the function that takes as input the stops and returns the locations. Autofocus properties and the fact that is especially designed exactly for this task make TOSCA (see Section 7.1) a good candidate for implementing the function *getLoc*. It is worth to notice that $|\mathcal{L}_u| < |S_u|$.

Definition 19 (Regular Locations). *Given a user u and her locations* $\mathcal{L}_u$*, her* regular locations $\mathcal{L}_u^{\Re} \subseteq \mathcal{L}_u$ *is a subset of* $\mathcal{L}_u$ *containing only the locations visited most frequently.*

In order to consider all the trajectories and all the stops, we define a dummy location containing all the stops which do not belong to a regular location, i.e., the union of the stops contained into the locations which are not regularly visited as $L_u^{\Im} = \{s | \exists\ l \in \mathcal{L}_u - \mathcal{L}_u^{\Re}$ s.t. $s \in L\}$. Hence we have $|\mathcal{L}_u^{\Re}| \leq |\mathcal{L}_u| < |S_u|$ and $|\mathcal{L}_u^{\Re}| + |L_u^{\Im}| = |\mathcal{L}_u|$. The irregular location $L_u^{\Im}$ is an abstract location from the mobility point of view but is formed by real stops to not frequent places. Knowing that the user is in $L_u^{\Im}$ it is a worth information because means that she is not present in a frequent location. We define with $getReg(\mathcal{L}_u) = (\mathcal{L}_u^{\Re}, L_u^{\Im})$ the function that takes as input the locations and returns the regular locations and the not irregular one $L_u^{\Im}$. An effective technique for distinguishing between regular and occasional locations is the *knee* method detailed in Section 8.2.1.

Each regular location can be interpreted as a *subjective point of interest*, a place around which the mobility of that individual gravitates. This allows to study and analyze the locations which are meaningful only for the individual, like her home, work place, gym, favorite shop etc. On these locations we define the following *indicators* and *probabilities*:

- $\omega_L : L \to N$ returns the number of times u was observed in location L
- $\tau_L : L \to N$ returns the time spent by u in location L
- $\rho_{L,t} : L \times T \to [0,1]$ estimates the probability to find u in L at time t
- $cm_L = (\frac{1}{|S_L|} \sum\limits_{s \in S_L} s.x, \frac{1}{|S_L|}) \sum\limits_{s \in S_L} s.y)$ center of mass of location L
- $n_L = |\mathcal{L}_u|$ number of locations
- $n_{\widetilde{L}} = |\mathcal{L}_u^{\Re}|$ number of regular locations
- ms_L minimum number of stops in a locations to be considered regular
- $cm_L = (\frac{1}{|\mathcal{L}|} \sum\limits_{L \in \mathcal{L}} cm_L.x, \frac{1}{|\mathcal{L}|} \sum\limits_{L \in \mathcal{L}} cm_L.y)$ center of mass of u
- $cm_{\widetilde{L}} = (\frac{1}{|\mathcal{L}^{\Re}|} \sum\limits_{L \in \mathcal{L}^{\Re}} cm_L.x, \frac{1}{|\mathcal{L}^{\Re}|} \sum\limits_{L \in \mathcal{L}^{\Re}} cm_L.y)$ center of mass of u w.r.t $\mathcal{L}^{\Re}$
- $rg_L = \frac{1}{|\mathcal{L}|} \sum\limits_{L \in \mathcal{L}} (cm_L - cm)^2$ radius of gyration of u
- $rg_{\widetilde{L}} = \frac{1}{|\mathcal{L}^{\Re}|} \sum\limits_{L \in \mathcal{L}^{\Re}} (cm_L - cm^{\Re})^2$ radius of gyration of u w.r.t $\mathcal{L}^{\Re}$
- $st_L = \sum\limits_{L \in \mathcal{L}} \tau_L$ total stay time in locations
- $st_{\widetilde{L}} = \sum\limits_{L \in \mathcal{L}^{\Re}} \tau_L$ total stay time in regular locations

Besides locations, the mobility of a user is obviously characterized by her *movements*, i.e., similar trajectories that start and end in the user's locations:

Definition 20 (Movements)**.** *Given a user u, her mobility history H_u, her regular locations $\mathcal{L}_u^{\Re}$ and the irregular location $L_u^{\Im}$, her* movements *$\mathcal{M}_u {=} \{M_0, \ldots, M_q\}$ is a partitioning of H_u into disjoint sets: $\forall\ M {\in} \mathcal{M}_u, M {\subset} H_u$, $\bigcup_{M \in \mathcal{M}_u} {=} H_u$ and $M, M' {\in} \mathcal{M}_u \wedge M {\neq} M' \Rightarrow M {\cap} M' {=} \emptyset$, where $\forall\ m {\in} \mathcal{M}_u, m {=} \{a_0, \ldots a_m\}$, s.t. $\sum_{i=0}^{k-1} |m_i| {=} |H_u|$ and $\forall\ M {\in} \mathcal{M}_u, \exists L, L' \in \mathcal{L}_u^{\Re} \cup L_u^{\Im} \wedge l {\neq} L' \wedge \exists a \in M$ s.t. $start(a) \in L \wedge end(a) \in L' \vee start(a) {\in} L' \wedge end(a) {\in} L$.*

In summary, a movement is a set of trajectories which start from a location L_1 and arrive in a location L_2. Each trajectory belongs only to a movement. In other words, a movement is an abstract trajectory for which the sequence of spatio-temporal points is not specified. We define with $getMov(H_u, \mathcal{L}_u^{\Re}, L_u^{\Im}) = \mathcal{M}_u$ the function that takes as input the mobility history, the regular locations and the irregular one and returns the set of movements. This function can be realized through a simple "group-by" considering the start and end locations. Like for the locations, also for the movements we define the following *indicators* and *probabilities*:

- $\omega_M : M \to N$ returns the number of times u followed movement M
- $\tau_M : M \to N$ returns the time spent by u traveling along M
- $\lambda_M : M \to N$ returns the distance traveled by u along M
- $\rho_{M,t} : M \times T \to [0,1]$ estimates the probability to find u along M at time t
- $n_M = |\mathcal{M}_u|$ number of movements
- $n_{\widetilde{M}} = |\mathcal{M}_u^{\Re}|$ number of movements among regular locations
- ms_M minimum number of trajectories in a movement to be considered frequent
- $tt_M = \sum\limits_{M \in \mathcal{M}} \tau_M$ total traveling time
- $tt_{\widetilde{M}} = \sum\limits_{M \in \mathcal{M}^{\Re}} \tau_M$ total traveling time along frequent movements
- $td_M = \sum\limits_{M \in \mathcal{M}} \lambda_M$ total distance traveled
- $td_{\widetilde{M}} = \sum\limits_{M \in \mathcal{M}^{\Re}} \lambda_M$ total distance traveled along frequent movements

However, a movement as it is defined is an abstraction revealing only the start location and the end location. Therefore, we exploit the concept of *routine* defined in [285].

Definition 21 (Routines). *Given a user u and her movements $\mathcal{M}_u$, her* routines *$\mathcal{R}_u$ is a subset of $\mathcal{M}_u$ containing for each movement $M \in \mathcal{M}_u$ the trajectories $R \subseteq M$ that better approximate the movement M and which are frequent for the user u, i.e., s.t. $|M| \geq ms_M$.*

We define with $getRep(\mathcal{M}_u) = \mathcal{R}_u$ the function that takes as input the set of movements and returns the set of routines representing each movement. It is possible to extract a routine representing a set of movements by applying clustering techniques like those in [285]. Note that if a movement does not contain enough trajectories then it cannot have a routine representing it. Moreover, a movement M can be represented by more than one routine. For example, a user could systematically move from L_1 to L_2 in two or three different ways to deal with traffic conditions depending on the time of the day or the day of the week. The personal threshold to determine when a movement can be represented through a routine because it is followed by a sufficient number of trajectories, can be calculated once again by using the *knee* method. It is worth to specify that, according to [285], the set of routines R_u of user u can also be extracted directly from the individual mobility history H_u through an appropriate clustering method with an ad-hoc distance function between trajectories. This approach enables also to specify a minimum level of support required by each routine in order to be considered as representative for that user.

All these elements – regular and irregular locations, movements, routines – can be combined using a network-like data structure. Similar to [249], from a mobility point of view this data structure links the elements in a natural way, and can be a fundamental component for the mobility of the Personal Data Store.

Definition 22 (Individual Mobility Network). *Given a user u, her regular locations $\mathcal{L}_u^{\Re}$, irregular location $L_u^{\Im}$, and movements $\mathcal{M}_u$, her* individual mobility network *is a directed graph G_u=(V, E), where V=$\mathcal{L}_u^{\Re} \cup L_u^{\Im}$ is the set of nodes and E=$\mathcal{M}_u$ is the set of edges.*

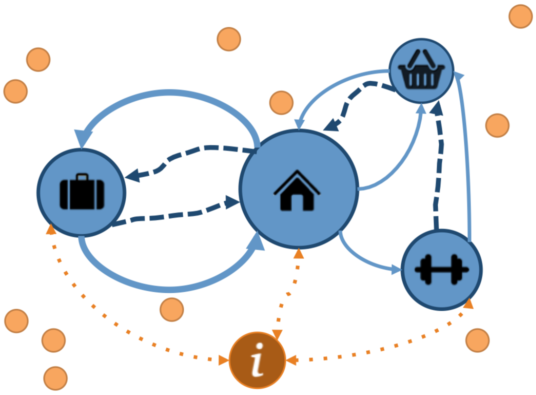


Figure 8.17: Example of *Personal Agenda*. This user visited 15 distinct locations. Four of them are regular locations: blue big circles (Home, Work, Shop, Gym). The remaining locations (small orange circles) are conceptually grouped into the irregular location. Six movements are detected: Home-Work and vice-versa, Home-Shop and vice-versa, Home-Gym and Gym-Shop. They are represented by the light blue straight lines. Finally, four routines – dotted dark blue lines – are the most frequent and systematic trajectories. The individual trajectory patterns are not highlighted.

Nodes represent locations and edges represent movements between locations. By using the indicators and functions previously described it is possible to accurately describe both nodes and edges by means of structural annotations. The individual mobility network of an individual is an abstraction of her mobility behavior at different layer. The irregular location is an abstract entity without any reference to the geographic space, while the regular locations have a well-defined shape and positioning. Similarly, the movements are an abstraction, while the routines are real trajectories.

Up to now, we have considered only elements and patterns without memory (e.g. locations and routines), or, according to a Markov models [160], components with one step of memory (e.g. the probabilities for locations and movements). However, in everyday mobility, certain behaviors are a consequence of the sequence of movements and locations that we have followed in the past. This is the reason why it is crucial for a personal mobility model to capture and extract also the frequent sequences considering both the locations together with their arrival and leaving time. We define the *individual trajectory patterns* of a user, or *t-patterns*, as follows:

Definition 23 (Individual Trajectory Pattern)**.** *Given a user u, her regular $\mathcal{L}_u^{\Re}$ and irregular $L_u^{\Im}$ locations, her* individual trajectory pattern $\mathcal{T}_u = \{T_0, \dots, T_n\}$ *is a set formed by couples defined as $T{=}(\bar{L},\bar{\alpha})$, where $\bar{L}{=}\langle L_0,\dots,L_{n-1}\rangle$ is an ordered sequence of locations, and $\bar{\alpha}{=}\langle \alpha_0,\dots,\alpha_{n-2}\rangle$ is the temporal annotation of the sequence such that $\forall_{\leq i<n-1}\ \alpha_i{<}\alpha_{i+1}$*

The temporal patterns can be represented also as:

$$T = (\bar{L},\bar{\alpha}) = L_0 \overset{\alpha_0}{\to} L_1 \overset{\alpha_1}{\to} \dots \overset{\alpha_{n-2}}{\to} L_{n-1}$$

Each pattern has a support sup_T indicating the number of occurrences of that particular sequence. It is worth to notice that, when a user stops in a location and then move again, in order to capture the stop and the permanence in that location, the pattern contains a repetition of the location. Examples of individual trajectory patterns are the following:

$$H \xrightarrow{\text{[8.00,8.15]-[8.20,8.35]}} W \xrightarrow{\text{[8.20,8.35]-[17.50,18.00]}} W \xrightarrow{\text{[18.20,18.40]-[8.20,8.35]}} H \quad (8.5)$$

$$\begin{aligned} H \xrightarrow{\text{[8.00,8.10]-[8.25,8.35]}} W \xrightarrow{\text{[8.25,8.35]-[16.30,16.50]}} W \xrightarrow{\text{[16.50,17.10]-[17.30,17.50]}} \ldots \\ \ldots \to S \xrightarrow{\text{[17.30,17.50]-[18.30,18.35]}} S \xrightarrow{\text{[18.30,18.35]-[19.00,19.30]}} H \quad (8.6) \end{aligned}$$

The t-pattern (8.5) represents the pattern Home-Work-Home capturing the behavior of a typical working day, while t-pattern (8.6) Home-Work-Shop-Home captures the sequences of the days when the user leave the working place a bit earlier to go to the shop.

In line with the model described in Section 8.2, the applications of various extraction functions on raw mobility data or on mobility data models lead to capture diversified and complementary aspects of the user mobility behavior. Hence, we can structure all the patterns defined for the personal mobility to form the *Personal Agenda*:

Definition 24 (Personal Agenda). *Given the locations $\mathcal{L}_u^{\Re}$ and $L_u^{\Im}$, the movements $\mathcal{M}_u$, the routines $\mathcal{R}_u$ the mobility network G_u and the trajectory patterns $\mathcal{T}_u$ we define the* Personal Agenda *of user u as the tuple:*

$$\begin{aligned} \mathcal{P}_u = \langle \mathcal{L}_u^{\Re}, L_u^{\Im}, & \qquad \texttt{locations} \\ \mathcal{M}_u, & \qquad \texttt{movements} \\ \mathcal{R}_u & \qquad \texttt{routines} \\ G_u, & \qquad \texttt{mobility network} \\ \mathcal{T}_u, & \qquad \texttt{t-patterns} \\ \mathcal{I}_u & \qquad \texttt{indicators} \\ \mathcal{E}_u \rangle & \qquad \texttt{probabilities} \end{aligned}$$

where $\mathcal{I}_u = \{n_L, n_{\widetilde{L}}, ms_L, cm_L, cm_{\widetilde{L}}, rg_L, rg_{\widetilde{L}}, st_L, st_{\widetilde{L}}, n_M, n_{\widetilde{M}}, ms_M, tt_M, tt_{\widetilde{M}}, td_M, td_{\widetilde{M}}\}$ and $\mathcal{E}_u = \{\omega_L, \tau_L, \rho_{L,t}, \omega_M, \tau_M, \lambda_M, \rho_{M,t}\}$

The whole data model can be recalculated from the raw data by using a moving window and discarding old information. On the other hand, it can be incrementally updated by assigning each trajectory to the movement that better represents it by consequently refreshing the various supports in the network and in the t-patterns. Figure 8.17 summarizes and clarifies the concepts and patterns forming the Personal Agenda.

8.3.2 Conclusion

We have presented the *Personal Agenda* a Personal Mobility Data Model structuring and organizing different and various mobility patterns and indicators in a unique model. This model is being entirely employed in an ongoing work to extract a detailed picture capturing the mobility agenda of each user, and also to build an "agenda planner" that can be exploited to predict the user movements and activities and to recommend alternative time schedule and route planning. In the rest of this thesis we employ some mobility patterns and indicators that we have introduced in this section.

Part IV

Personal Data Analytics for Individual and Collective Services

Chapter 9

Improving Personal Mobility

Mobility is a central dimension of our society and it crosses the choices we make in our everyday life. Smart Cities applications are fostering research in many fields including physics, computer science, engineering, but also sociology and psychology. Data Mining is used to support applications such as optimization of a public urban transit network [33], carpooling [285], smart traffic monitoring [88], urban event detection [32], and many more.

Personalization is the turning point for improving mobility services. Through Personal Data Analytics we can exploit Personal Data Models to overcome the limitation of the classical approaches used for location-based services. Indeed, the "wisdom of the crowd" can better emerge from the users' profiles and can be better exploited to satisfy individual needs and preferences. In this thesis we show how the Personal Mobility Data Models defined in the previous sections can be exploited in real-world mobility services aimed at improving personal mobility. In the following, we describe how Personal Data Models are used to build a trajectory prediction service that could provide a driver information about activities she may perform in the future locations, or traffic problems that may occur along the route predicted. Moreover, we demonstrate how the Personal Data Store and its models can become a fundamental component for a route planner service based on personal experience, and how the routes suggested differ from those of classical planners.

9.1 Trajectory Prediction through Mobility Profiling

Predicting the future locations of a mobile user is a flourishing research area that is powered by the increasing diffusion of location-based services. The knowledge of mobile user positions fosters applications which need to know this information to operate efficiently. Examples of such services are traffic management, navigational services, mobile phone control, etc. Many *location-based services* are based on the current or on future locations of a user. By using the knowledge about the locations it is possible to fetch relevant information such as nearby points of interest and available services. Moreover, predicting the future positions can inform a driver about services like restaurants, banks, shops which are present in the future locations, or traffic problems that may occur along her route.

Nowadays, every moving device periodically informs the positioning system of its current location. Due to the unreliable nature of mobile devices and to the limitations of the positioning systems, the location of a mobile object can be often unknown for a long period of time. In such cases, a method to predict the next position of a moving object is required in order to anticipate or to pre-fetch possible services. The strong interest in this kind of applications led to the study of several approaches in the literature addressing the location prediction problem. Some of them base the prediction on single users' movement history, while others extract common behaviors from the histories of all the users in the system.

In line with Personal Data Analytics, we propose *MyWay* [284], a system to forecast the *exact* future positions. *MyWay* predictors exploit the individual systematic behaviors of a single user, the individual systematic behaviors of all the users in the system (called collective behavior), and a combination of them. To predict the future positions of a user, *MyWay* first uses her systematic behaviors and, if they are not sufficient, it exploits the systematic behaviors of the crowd. This idea is based on the conviction that typically any user systematically visits a small set of locations and regularly moves between them by choosing the best movements learned by the daily experience [123, 269]. *MyWay* requires that each individual computes an abstract representation of her systematic behavior, i.e., the routines which are a component of the Personal Mobility Data Model (PMDM) described in Section 8.3. In this section, we refer to them with the expression of *individual mobility profiles*. In particular, we consider as *individual mobility profile* the paths that are regularly followed by the user, i.e., the *routines* R_u [285]. Then, at collective level *MyWay* requires that the *individual mobility profiles* are shared among the users of the PDE.

The following prediction strategies are developed. The *individual strategy* predicts the future positions using only the *routines* part of the user's *individual mobility profile*. The *collective strategy* considers the routines of all users exploiting the possibility that a user could follow a path which is atypical for her but systematic for another user. The *hybrid strategy* uses the collective strategy when the individual one fails. As theorized in Chapter 5, *MyWay* exploits the possibility to use two levels of knowledge (individual and collective), obtaining advantages from the previous strategies. In addition, a great novelty introduced by *MyWay* is that it does not apply any apriori spatial discretization. In fact, most of the works proposed so far in the literature apply a spatial discretization such a fixed grid on the space [215, 216] or a territory tessellation obtained by clustering spatial points [18, 156]. The spatial discretization often affects the precision of the prediction that instead of returning spatio-temporal points returns regions with higher granularity.

Our claim is that the prediction strategy which uses only individual mobility profiles is comparable with a prediction strategy based on raw movement data. If confirmed, this approach has two important advantages: *(i)* it dramatically minimizes the quantity of information required since a mobility profile is a concise representation of the information in the user PMDM; and, *(ii)* it can help to reduce the privacy risks: the mobility profile represents a systematic behavior, i.e., paths that are regularly followed by the user, but does not reveal all the details of her past spatio-temporal positions. Moreover, in a mobility profile the spatial information is a representation of a group of similar raw trajectories, while the temporal information expresses the relative time and not the absolute one. This means that given a profile we can know that the user typically visits the place A at 11AM but, we are not sure if on a specific day he visited that location. Clearly, using the mobility profile we cannot guarantee a specific privacy protection, but surely the risk is lower than the one that can derive from the use of raw data. In order to have a formal privacy guarantee we should adapt one of the possible privacy technologies developed in the last years for trajectory data [3, 220, 278] or we should design a new one by following the privacy-by-design approach [212]. However, this aspect is out of the scope of this thesis.

9.1.1 MyWay Prediction System

Problem Definition

The problem we face consists of predicting the future positions visited by a user at specific time instants by exploiting the typical mobility behavior of users in the system. The different formulations of the problem and the possible solutions are determined by the type of the object, the area in which it is moving, the kind of prediction returned and how the notion of future is defined. The main challenge of this problem is due to the complexity and fine granularity of GPS data. Often, most of the works in the literature apply a spatial discretization by using clustering techniques on spatial points or simply a grid on the space to reduce the complexity of the problem. Clearly, on one hand, this makes easier finding frequent or interesting locations and patterns to be exploited in the prediction; on the other hand, it affects the precision of the prediction that often returns *regions* with a granularity imposed by the apriori discretization.

The prediction method proposed does not use any apriori spatial or temporal discretization, i.e., it is parameter-free, and, given a user u and her current trajectory m, aims at forecasting the future *exact* position visited by the user u at a specific time instant t. This task is composed of two main steps: *(i)* learning a prediction model by observing historical movement data, and *(ii)* applying the prediction model to forecast the future positions.

MyWay is a system of prediction strategies able to solve this challenging task. It exploits the users' systematic mobility stored in their PMDM, i.e., the user individual mobility profile, and the knowledge coming from the PDE in the form of a collective profile. To build such models we refer to the mobility profiles presented in Section 8.3 and to the *routines* R_u. In the following we define a new distance function between trajectories which is more efficient and gives a better result in terms of profile quality with respect to the one used in [285]. Then, we define the prediction method: two basic *individual* and *collective* prediction strategies, and a third strategy that combines the basic ones, called *hybrid*.

Distance Functions

The mobility profile extraction process uses a distance function during the clustering step to identify *similar* trajectories. In practice, the distance function defined between two trajectories determines if they are representing a similar movement. There are many possibilities in defining such distance. Some examples are described in [13] (i.e., the one used in [285]) and in [116], and each one analyzes a different perspective and is used for a particular objective. Another important aspect to consider is the complexity of such distance function which greatly affects the performance of the whole process.

We adopt a different distance function from the *Route Similarity* used in [285] due to the fact that it assumes that the two trajectories have the same sampling rate. In general,

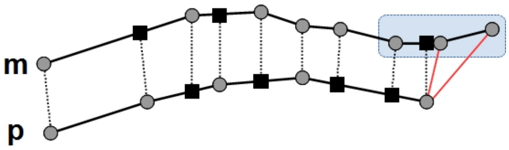


Figure 9.1: Computation of *Interpolated Route Distance*. The circular gray points are the real points, the black squares are the interpolated ones. The dotted lines are the spatial distances.

Algorithm 9: *distanceFunctionIRD(Trajectory t_1, Trajectory t_2).*

```
Input  : $t_1, t_2$ - trajectories
Output: $d$ - distance between $t_1$ and $t_2$
$d \leftarrow 0$;
$i_1 \leftarrow 1$, $i_2 \leftarrow 1$;
$p_1 \leftarrow getPoint(t_1, i_1)$, $p_2 \leftarrow getPoint(t_2, i_2)$;
while $\neg(i_1 = getSize(t_1) \wedge i_2 = getSize(t_2))$ do
    $d \leftarrow d + sphericalDistance(p_1, p_2)$;
    $len_1 \leftarrow \infty$;
    $len_2 \leftarrow \infty$;
    if $(i_1 < getSize(t_1))$ then
        $len_1 \leftarrow sphericalDistance(p_1, getPoint(t_1, i_1 + 1))$;
    end
    if $(i_2 < getSize(t_2))$ then
        $len_2 \leftarrow sphericalDistance(p_2, getPoint(t_2, i_2 + 1))$;
    end
    if $(len_1 < len_2)$ then
        $i_1 \leftarrow i_1 + 1$;
        $p_1 \leftarrow getPoint(t_1, i_1)$;
        $p_2 \leftarrow getNearestPoint(t_2, p_1)$;
    else
        $i_2 \leftarrow i_2 + 1$;
        $p_2 \leftarrow getPoint(t_2, i_2)$;
        $p_1 \leftarrow getNearestPoint(t_1, p_2)$;
    end
end
return $d$;
```

this is not true in real-world datasets and the bias introduced by this assumption may produce anomalous effects. Moreover, a misleading distance value is produced when the sampling rate and the frequency of the observations differ in the two trajectories. This is due to the comparison of the existing raw points (i.e. no interpolation is used) and the usage of a heuristic applying a penalty when there is a difference in the number of points.

Similar limitations characterize the distance function introduced in [9]. To overcome these limitations we define a new distance function having the following advantages:

Interpolation: Each point of a trajectory is compared to the closest point over the segments of the other one. This avoids the asynchronous sampling rate and the different frequency of points.

Efficiency: *Route Similarity* is highly inefficient comparing several times the same point to others in order to find the closest one; in our case the comparisons are equal to the number of points in the trajectories (in the worst case).

Symmetric: *Route Similarity* uses a heuristic to give penalty when a point does not find a correspondence in the other trajectory. This process is not symmetric and this is not acceptable for a distance function.

For all these reasons we define a distance function suitable for our purposes called *Interpolated Route Distance* (IRD). The function temporally aligns the first points of the two trajectories m and p using the initial time, then for each point in m, it interpolates a point in p – if it does not exist – and vice-versa. Finally, it computes the spatial distance (spherical) between each pair of aligned points (real or interpolated). When one of the two trajectories is longer, then the exceeding part is compared with the last point of the short trajectory (i.e. we consider as if the user stops at the last point when a trajectory ends). The average of those distances is the result of IRD. Fig. 9.1 and Alg. 9 show how IRD is computed. In Alg. 9 the *getPoint* returns the i^{th} point in the trajectory (e.g. $i = 1$ indicates the first point), *sphericalDistance* returns the spherical distance between two points, and $getNearestPoint$ given a point and a trajectory find the nearest interpolated point to the segments of the second (i.e. the square points in Fig. 9.1).

Moreover, for our purposes we also define a slight variation of *IRD.* We call this distance function *Constrained IRD* (CIRD) since, besides the two trajectories, it takes as input also two parameters (γ, σ) called respectively *tail percentage* and *prediction threshold.* They are used to verify if in the last $\gamma\%$ of the trajectory m exists a point which is further than σ meters from the trajectory p. If this happens the distance function returns an *infinite distance*, i.e., it considers the two trajectories not comparable. An example of the portion of trajectory influenced by the constraint is depicted in Fig. 9.1 as a blue box.

The computational complexity of the Alg. 9 depends on the number of spherical distances and interpolation to be computed. It's easy to see that in the worst case, i.e. when none of the points are aligned, the number of interpolation are exactly the same of the distances to be computed. This number at the maximum is the sum of the points of each trajectory, therefore removing the constant factor, the complexity is $O(l_1 + l_2)$ where l_1 and l_2 are respectively the numbers of point in t_1 and t_2.

Method

We define the prediction method as a function over a mobility profile which, given the current trajectory m, returns the *exact* future position s of the user after a time period $\hat{t}$. More in detail, the prediction method is composed of two functions: *Match* which finds in the profile the routine most similar to the current trajectory, and *LookAhead* which predicts the future position having a routine and the current user position.

Definition 25 (Match)**.** *Let γ and σ the CIRD parameters. Given a trajectory m, and the routines R=$\{r_1, \dots, r_n\}$ of the mobility profile, the routine r is selected if:*

$$r = Match(m, R, \gamma, \sigma) = arg\,min_{r_i \in R} CIRD(Cut(r, m), m, \gamma, \sigma)$$

In the above definition, $Cut(r, m)$ selects the sub-trajectory of the routine defined between two points: q that is the closest point (real or interpolated) to the last point of the trajectory m, and b that is the temporally antecedent point which makes the length of the sub-trajectory equal to m. If the routine is not long enough b is the first point.

The usage of CIRD (and therefore γ and σ parameters) represents our interest in eliminating possible false positive matches given by common initial parts of the trajectory m and of the routines in R. We say that a match $Match(m, R, \gamma, \sigma)$ is *undefined* if R is empty or CIRD returns an infinite distance for each $r \in R$, i.e. the process ends without any routine for the match. The process of matching the trajectory m with a routine r is

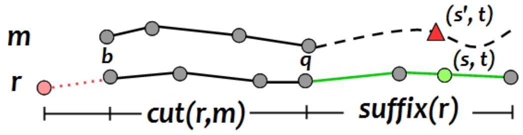


Figure 9.2: Match example between the current movement m and the routine r.

shown in Fig.9.2 where in solid black we represent the portions that are compared, while in dotted red and green the parts ignored for the matching.

Once r is obtained, i.e., the most similar routine to the current trajectory m, then we can use it to predict the position within a time period as follows:

Definition 26 (LookAhead)**.** *Let m be the current trajectory, r a routine, $\hat{t}$ a time period and $q{=}(x,y,t)$ as the closest point in r to the current position, i.e. the last point of m. Then, we define $LookAhead(r,m,\hat{t}){=}s$, where s is the predicted point in r at time $t{+}\hat{t}$. If the routine is shorter than $t{+}\hat{t}$ in time, then as point s is returned the last point of r.*

The combination of *Match* and *LookAhead* realizes the prediction methods used for all the strategies in *MyWay* system. More formally we can define a predictor as:

$$s = Pred(m, R, \hat{t}, \gamma, \sigma) = LookAhead(Match(m, R, \gamma, \sigma), m, \hat{t})$$

where m is the current trajectory, R the routines, $\hat{t}$ a time period, and s is the resulting prediction point. The difference between the three strategies is how the method is used. We must notice that if the result of $Match$ is *undefined*, then also the result of $Pred$, that is s (the predicted point), is *undefined.* Note how in this service the Personal Data Model consists of the individual mobility profile and, in particular, on the user's routines R.

The *individual strategy* predicts the future positions of a user by exploiting only the systematic behavior of the user herself. Therefore, it is particularly suitable for users having a high degree of systematic mobility. More formally, we define the *individual predictor* for a user u as: $s = Pred(m, R_u, \hat{t}, \gamma, \sigma)$.

The *collective strategy* considers the routines of all users for the prediction, thus exploits the possibility that a user could follow an atypical path for her but systematic for another user. More formally, we define the *collective predictor* as: $s = Pred(m, R_C, \hat{t}, \gamma, \sigma)$.

Finally, we define the *hybrid strategy* as a composition of the individual and of the collective ones:

$$s = \begin{cases} Pred(m, R_u, \hat{t}, \gamma, \sigma) & \text{if } not\ undefined \\ Pred(m, R_C, \hat{t}, \gamma, \sigma) & otherwise \end{cases}$$

The hybrid strategy uses the individual predictor when is possible. If an individual prediction is not found, i.e. it is *undefined*, then the hybrid strategy uses the collective predictor. The idea behind the hybrid strategy is to recognize the specificity of the individual profile compared to the collective profile. Indeed, the collective strategy mixes up all the user's routines with the routines of the crowd and loses the added value of knowing the individual mobility profile of a specific user which enables very accurate predictions.

The resulting three predictors are shown in Fig.9.3, here for each predictor a different color is used: individual history, the individual profile and the *individual predictor* (red) are inside the user PMDM, while the *collective predictor* (blue) is outside and therefore handled by the PDE through a distributed protocol for sharing the users' information as

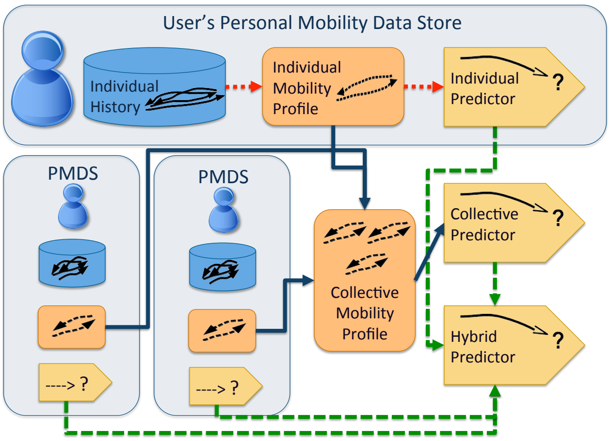


Figure 9.3: MyWay prediction strategies schema.

well as the *hybrid predictor* (green). In the *hybrid* strategy, thanks to the PDE the models and predictors are stored in the network, every user can use her own PDS, and the query for the prediction is distributed only in case the individual predictor of a specific user fails. Our experiments will show how the hybrid strategy achieves the best performances.

9.1.2 Experiments

In the following we evaluate *MyWay*'s prediction strategies performances. First of all we present the measures used to evaluate the predictions, then we describe the dataset used and the parameter setting, and finally the various prediction experiments against the competitors and considering different degrees of profile sharing.

Evaluation Measures

It is important to note how *MyWay* is challenging a very hard prediction problem due the following considerations: *(i)* users do not move every time in the same period of the day (at least not exactly); *(ii)* movement speed is not constant during the travel, even following the same trajectory; *(iii)* possible errors deriving from spatial sampling of the data could deeply influence the predicted position both in time and space. Consequently, it is reasonable to consider a set of tolerances to fairly evaluate the results. We use $spat_{tol}$ and $temp_{tol}$ to describe the spatial and temporal tolerances which generate a spatio-temporal area around the real point. This area contains all the values considered correct for the prediction problem. An example of usage of these tolerances is shown in Fig.9.4.

Definition 27 (Spatio-Temporal Tolerance)**.** *Given the predicted position s at time t, the real position s' at time t', and the position s'' at time t'' that is the closest real position to s such that $|t'-t''| \leq temp_{tol}$, then the prediction is considered correct iff $\|s-s''\| \leq spat_{tol}$.*

It is worth to underline that if $temp_{tol}=0$ then $s'=s''$ and thus we are predicting exactly the point where the user will transit in future without any temporal tolerance. To enhance the importance of $spat_{tol}$ we can consider two different environments for applying prediction: taking into account an academic campus, it is meaningless to adopt $spat_{tol}$

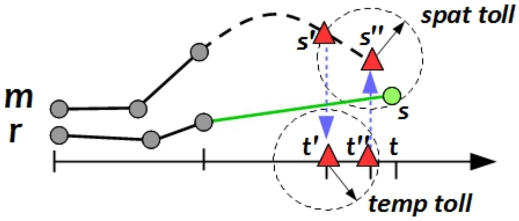


Figure 9.4: Spatial and temporal tolerances example: the red triangles are the real points s' and s'' such that $|t' - t''| \leq temp_{tol}$, the green circle is the predicted point such that $\|s - s''\| \leq spat_{tol}$.

greater than kilometers because nearly every prediction would be classified as correct; on the other hand, if we are considering toll roads then low $spat_{tol}$ would be inadequate.

Furthermore, let $\mathcal{TS}$ be the set of trajectories for which we want a prediction, $\mathcal{TP}$ the set of trajectories for which a prediction is provided, and $\mathcal{TPC}$ the set of trajectories for which the future spatio-temporal position is correctly predicted, then the following validation measures are defined and considered in experiments:

- *Prediction rate* $= \frac{|\mathcal{TP}|}{|\mathcal{TS}|}$ allows to estimate the predictive ability and corresponds to the percentage of trajectories for which a prediction is supplied;
- *Accuracy rate* $= \frac{|\mathcal{TPC}|}{|\mathcal{TP}|}$ allows to estimate the prediction goodness and corresponds to the percentage of future spatio-temporal positions correctly predicted;
- *Spatial Error* $= \frac{\sum_{\forall (s,s'')} \|s - s''\|}{|\mathcal{TP}|}$ allows estimating the error of the predictions (both correct and incorrect).

Dataset

We perform our experiments on the real GPS dataset named *Octo* described in Section 6.1. From this dataset we selected the users traveling through Pisa province with at least 20 travels considering only weekdays. Considering that in Pisa province there are about 476, 260 trajectories, this led to a dataset with 30% of all the users and 80% of the all trajectories, that is about 5, 000 users and 326, 000 trajectories. We consider as training set the first 3 weeks, and as test set the remaining last week. We test *MyWay* using two different test sets: one by considering only the first 33% of each trajectory ($test_{33}$), and one by considering the first 66% ($test_{66}$). These two test sets represent two levels of knowledge of the current movements and we will show how they affect accuracy and prediction rate. From an analysis of the PMDM extracted from this dataset results that for each user the routines have a minimum support of five trajectories. The percentage of total trajectories covered by all the routines $R_C = \bigcup_{u \in U} R_u$ is 47.91%, i.e., nearly half of the movements can be classified as systematic.

Parameter Setting

The parameter setting adopted in the experiments is reported in Tab. 9.1. With respect to CIRD, from some test on a sample of users we found that $\gamma = 10\%$ and $\sigma = 500$ meters are able to reduce the error in prediction, i.e., the number of false positives (or false matches). On the other hand, since they are quite restrictive the prediction rate is negatively affected.

Symbol	Description	Values
γ	tail percentage	$\{5, \mathbf{10}, 15, 20, 25\}$ %
σ	prediction threshold	$\{0.2, \mathbf{0.5}, 1.0, 1.5, 2.0, 5.0, 10.0\}$ km
t	lookahead	$\{0, 1, 2, 5, \mathbf{10}, 12, 15, 17, 20\}$ min
$spat_{tol}$	spatial tolerance	$\{0.05, 0.10, \mathbf{0.25}, 0.50, 0.75, 1.00\}$ km
$temp_{tol}$	temporal tolerance	$\{0.0, \mathbf{0.5}, 1, 2, 5\}$ min

Table 9.1: Set of evaluation parameters. The values in bold represents the default setting.

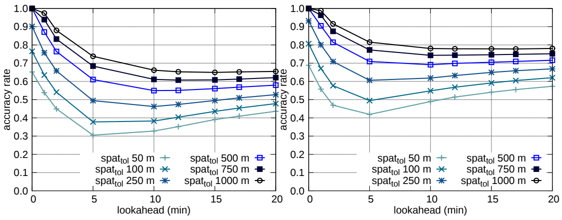


Figure 9.5: (Individual) Accuracy rate on $test_{33}$(left) and $test_{66}$ (right) using a $temp_{tol}$ of 30 sec, different $spat_{tol}$ values and varying the look ahead.

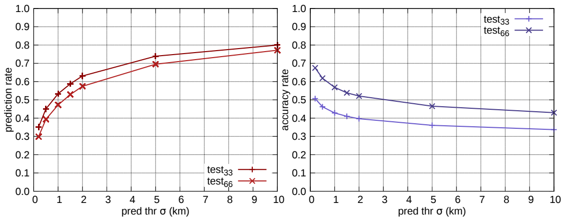


Figure 9.6: (Individual) How the prediction threshold affects the prediction and accuracy rate using a $spat_{tol}$ of 250 m and a $temp_{tol}$ of 30 seconds.

Finally, we show in the following the performances for the various look ahead and spatial tolerances and we present results for temporal tolerance equals to 30 seconds even though for higher values we obtain better performances.

Prediction Evaluation

Individual Strategy. The individual prediction consists in using the mobility profile of a single user to predict her future positions. In Fig.9.5 the accuracy obtained over the two test sets $test_{33}$(left) and $test_{66}$(right) is shown. Here, different levels of spatial tolerance $spat_{tol}$ are used (from 50 m to 1 km) with a temporal tolerance $temp_{tol}$ of 30 seconds. The first aspect to notice is how the accuracy varies for different time periods $\hat{t}$ used for the look ahead: the prediction for very short-term (1-5 minutes) is lower than the mid-term (5-20 minutes). This is due to the fact that in the short-term predictions the speed of the current movement may be very different from that in the routine, e.g., an extemporary acceleration, deceleration or a traffic light may affect the prediction accuracy. On the other hand, for the mid-term prediction the speed tends to be similar to the average speed and the prediction becomes more precise. For example, considering a variation of speed of $30km/h$ in one minute we have a spatial difference of 500 meters. Clearly, using a higher temporal tolerance this effect disappears, but this strongly depends on the application in which the prediction is used. The second aspect to notice is the higher accuracy rate in $test_{66}$ w.r.t. $test_{33}$. This happens because in $test_{66}$ the knowledge on the current movement is higher and therefore our method is able to better understand which is the best routine to use. The third aspect, shown in Fig.9.6(left), is the prediction rate that is higher in $test_{33}$ than in $test_{66}$. The limited knowledge of the current movement allows the predictor to match more routines even though they are not the exact future trajectory. In details, passing from $test_{33}$ to $test_{66}$, we have an increasing of $10-15\%$ for the accuracy rate and a decrease of $5-8\%$ for the prediction rate. This behavior is really interesting because highlights how *MyWay* reacts to the information gained from the query or, in other words, how it can tune the prediction in a real scenario as the user proceeds along her travel.

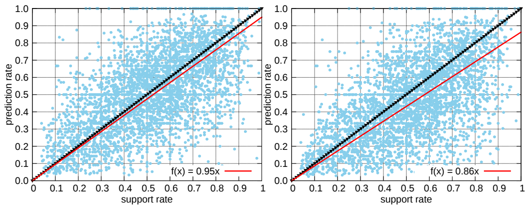


Figure 9.7: (Individual) Predictability of users vs. prediction rate. Respectively the Pearson coefficients are 0.6372 and 0.6771.

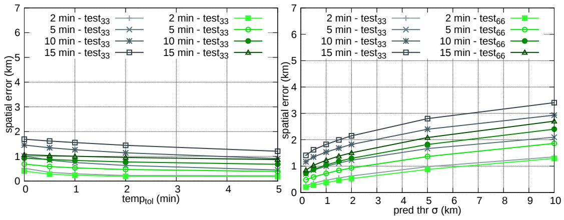


Figure 9.8: (Individual) The spatial error using $test_{33}$ and $test_{66}$ varying the $temp_{tol}$(left) and the prediction threshold (right).

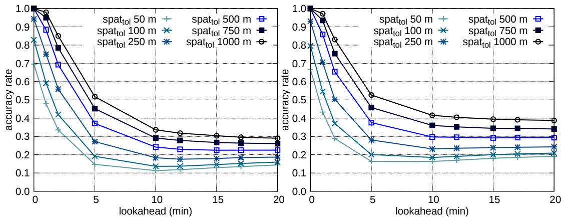


Figure 9.9: (Collective) Accuracy rate on $test_{33}$(left) and $test_{66}$ (right) using a $temp_{tol}$ of 30 sec, different $spat_{tol}$ values and varying the look ahead.

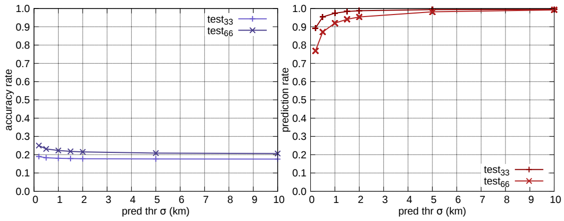


Figure 9.10: (Collective) How the prediction threshold affects the prediction and accuracy rate using a $spat_{tol}$ of 250 m and a $temp_{tol}$ of 30 seconds.

In Fig.9.6 the prediction rate (left) and the accuracy rate (right) are studied varying the prediction threshold. We observe that relaxing this threshold the two measures respectively increase and decrease. Allowing a more loose matching in the end part of the current trajectory more predictions are produced (due to the constraint in CIRD); on the other hand, the accuracy rate decreases but it is important to note how this is not proportional. In other words, it is possible to tune the system according to the application needs in order to be more *conservative* - i.e. if the errors in prediction are considered critic fails - or *speculative* -i.e. having a prediction is better even if we introduce errors. Note that the *prediction tail* γ parameter is not shown due to the lack of space but extensive tests revealed that it enhances the prediction threshold effect.

To better understand the quality of the prediction, we study the relation between the *users' predictability* and the prediction rate obtained with this strategy. For this analysis we consider: the prediction rate and the *support rate*, defined as the ratio between the number of trajectories represented by the routines and the number of trajectories in the individual history. The result is shown in Fig.9.7 where each point refers to a user and in red we represent the linear regression of those points. The dotted black line represents the performance of a theoretically perfect system which matches exactly all the movements to the proper routine. If the user's routines cover $k\%$ of her movements, the theoretical system is able to predict a maximum of $k\%$ of the trajectories because the rest is composed by not systematic movements that are unpredictable using the user's mobility profile. Comparing the two lines we can notice how our system is close to the theoretical one.

Finally, we analyze the spatial error of the predictions shown in Fig.9.8. We observe on the left how the spatial error increases considering higher look-ahead values, and it slightly decreases with higher value of $temp_{tol}$, while on the right we can see the effect of a higher prediction threshold which makes *CIRD* more *permissive* increasing the spatial error.

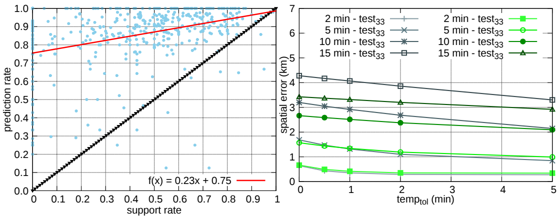


Figure 9.11: (Collective) Predictability of users vs. prediction rate (Pearson coefficient 0,5165) (left) and the spatial error varying the $temp_{tol}$ (right).

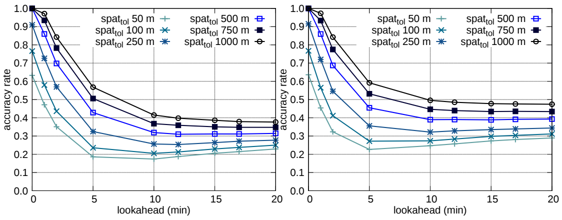


Figure 9.12: (Hybrid) Accuracy rate on $test_{33}$(left) and $test_{66}$(right) using a $temp_{tol}$ of 30 sec.

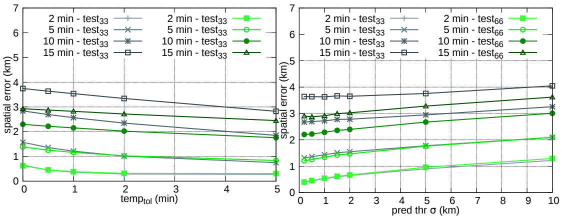


Figure 9.13: (Hybrid) The spatial error using $test_{33}$ and $test_{66}$ varying the $temp_{tol}$(left) and the prediction threshold (right).

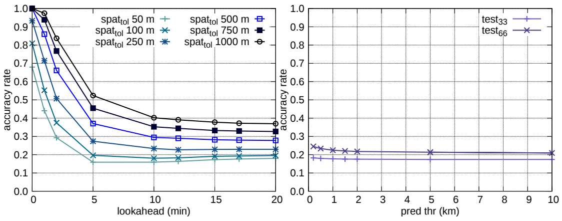


Figure 9.14: (Global predictor) Accuracy rate on $test_{66}$ using a $temp_{tol}$ of 30 seconds (left) and comparison between $test_{33}$ and $test_{66}$ varying the prediction threshold σ (right).

Collective Strategy. The collective strategy uses the collective mobility profile composed by the union of all the routines R_C. In Fig.9.9 the accuracy and the prediction rate over the two test sets are reported. We notice how the collective strategy presents a decrease of $15-20\%$ in accuracy, while the prediction rate is increased by a $30-45\%$ obtaining values greater than 85% as shown in Fig.9.10. The effect of the collective knowledge strongly increases the performances, indeed almost all the queries have a prediction even if their quality decreases. This is due to the fact that we are using strangers behaviors to predict the user's movements. Moreover, Fig.9.11(left) shows how this strategy overcomes the predictability limitation: the red line is over the black dotted one representing the fact that the prediction rate for most of the users is over the support of their profiles. Looking at the spatial error, Fig.9.11(right), and comparing it to the individual strategy we note that it increases following the lower accuracy rate provided by this strategy.

Hybrid Strategy. The idea behind the hybrid strategies is to recognize the specificity of the individual profile compared to the collective profile, in other words, it consists in using the user's individual profile and, in the case it fails, in using the collective profile. Obviously, this strategy achieves the same prediction rate of the collective one because in the worst case this last strategy is used while, as expected, the evaluation results show an increasing of accuracy equal to 10% as shown in Fig.9.12. Therefore, the hybrid strategy outperforms the basic versions realizing the best trade-off between accuracy and prediction rate. Analyzing Fig.9.13 we can notice that also the spatial error is mitigated by the two levels of prediction showing a decreasing in every combination of the parameter values.

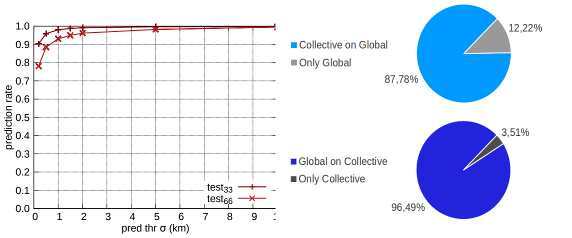


Figure 9.15: (Global predictor) Prediction rate (left), collective and global coverage (right).

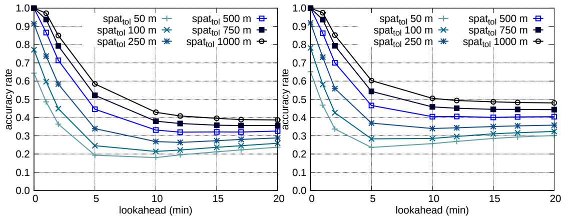


Figure 9.16: (Hybrid approach with global predictor) Accuracy rate on $test_{33}$(left) and $test_{66}$($right$) using a $temp_{tol}$ of 30 seconds.

Data Sharing vs Profile Sharing

Now, we want to compare *MyWay* with a *global predictor* extracting the routines directly from raw data. In other words, the profiling process considers all the trajectories of all the users. In this way we obtain *global routines* and a *global mobility profile*. Note that, a global routine, instead of representing the systematic movement of an individual, represents a *common behavior* of the crowd. In Fig.9.14 and 9.15(left) we report the prediction performances of the *global predictor*. We observe that its prediction rate is essentially the same of our collective strategy, and its accuracy rate increases of less than 3%. This means that for the prediction task the global profile does not increase significantly the level of knowledge. In other words, compared with the collective profile some routines are missing due to the higher level of abstraction. Moreover, some new routines, composed by a *common but not systematic behavior*, are created but the overall prediction power remains similar. This is also confirmed in Fig.9.15(right) showing that the collective profile covers the global profile and viceversa. The containment between them highlights that they substantially represent the same set of behaviors. Fig. 9.16 shows as the hybrid approach applied in the global context (individual/global combination) improves the accuracy rate leading to similar performances of our hybrid strategy.

Furthermore, we observe some advantages of *MyWay* w.r.t. the global predictor.

Data disclosure. A *global predictor* requires that the user shares with the coordinator her individual history that describes in detail all her movements; on the contrary, *MyWay*, in the worst case, requires to disclose only the *routines*, a model that surely reveals the user mobility behavior with fewer details. This aspect is very important because nowadays, people are often reluctant to share personal information because in the current systems users have a limited capability to control and exploit it. Therefore, in order to enable applications that require the active participation of people, it is necessary to encourage individuals in contributing with their self-knowledge to improve the quality of services offered by those applications. The opportunity of sharing models instead of detailed trajectories without causing deterioration of the performances is a good advantage of our system.

Communications. The need of sharing raw data also raises a problem in the communications cost needed to transfer all the data from all the users to the coordinator. With *MyWay* we can transmit only the information which is really needed for the prediction leading to a reduction of more than 97% of the data, i.e., spatio-temporal points. This is essential for an application which wants to gather information from a wide number of users.

Computational Cost. The main difference between the hybrid and global predictor is the fact that the model of the first one is composed by the union of the routines computed individually by each user, while the model of the second one is extracted from the whole data computed by a coordinator. The extraction is performed with a clustering algorithm with a complexity of $O(n\ log(n))$, thus it requires $O(\sum_{u \in U} |M_u| log(|M_u|))$ for the hybrid predictor, and $O(m\ log(m))$ for the global one, where $m{=}|\bigcup_{u \in U} M_u|$. Finally, the matching phase consists in finding the model routine minimizing the distance with the current trajectory, hence the complexity is $O(|R_C|)$, i.e. the number of routines in the model. Considering that the number of trajectories per user is significantly lower (of orders of magnitude) than the entire dataset, we can appreciate the great advantage of our system in terms of computational cost. We obtained an average runtime of 10 seconds for the individual routines, and more than 8 hours for the construction of a global profile. The experiments are executed on a single machine with 4 processors at 4.2 GHz each. The running time for building all the individual profiles R_u for each $u{\in}U$ is 7.24 hours, the collective predictor is simply the union of the individual profile and therefore the building time is not relevant and finally the time needed to build the global predictor is 8.43 hours.

Model Update. Models extracted cannot last forever: the mobility of the users may change significantly during different periods, thus it is reasonable to consider a method to update the profiles in a running system. In the collective strategy we can suppose to have at individual level a method to check if the last profile is still valid or not - e.g., considering the profile coverage over the most recent user's trajectories. In the case of a variation, the user recomputes a new model, sends it to the coordinator updating the collective profile by substituting the old user's profile with the new one. In the global scenario this is not possible, in fact, the user must send continuously her data to the coordinator which periodically recomputes the overall profile to remain up-to-date.

Comparing with State-Of-The-Art

We compare the performances of *MyWay* with individual and global competitors.

Individual Competitor. We compare our prediction system with the machine learning based individual predictor presented in [10]. Since this method uses an apriori spatial discretization, for a fair comparison we decided to use a grid that strongly affects our results. We perform the comparison constructing for each spatial tolerance (250, 500, 1000 meters) two different kinds of grid. The first one has a cell side equals to the square inscribed in a circle with radius equals to our spatial tolerance (lower bound x^l); while the second one has a cell side equals to the square inscribing this circle (upper bound x^u).

Note that, this approach does not use any notion of lookahead, i.e., it cannot predict the future position after a specified time interval from the current time, but it can just predict the next cell. It deals with trajectories represented as sequences of cells in a grid. We reimplemented and tested this method on our individual routines showing the performances in Fig. 9.17. As in [10], we discretized the trajectories in sequences of length h and we studied the goodness of a prediction varying this value. Note that, we used our individual routines instead of the starting dataset of trajectories because in [10] the authors state that they use systematic movements. Comparing the performances of our *individual strategy* (Fig. 9.5 & Fig. 9.6) with this competitor, we can see that our method provides more accurate predictions. This is true even if we consider for the competitor the sequence length that gets the best results. However, the machine learning predictor gets a higher prediction rate w.r.t. our individual strategy. Nevertheless, as shown above, we can overtake this lack by using our hybrid strategy. Moreover, we also test our individual

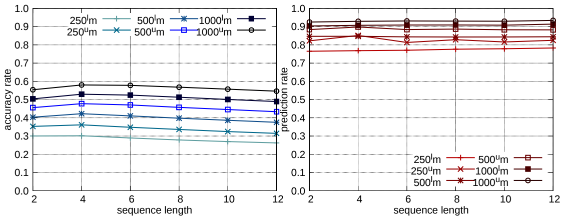


Figure 9.17: Individual competitor performances using different values of sequence length and grid side.

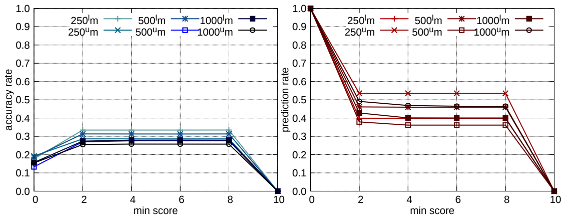


Figure 9.18: WhereNext performances using different value of minimum score and grid size.

predictor using an infinite $temp_{tol}$ in order to exclude the time dimension (not considered by the competitor) and using $spat_{tol}$=500 we obtain a prediction rate of 87% and an accuracy over 70% which are clearly higher than the competitor performances for any value of h.

Global Competitors. We also compare our proposal with method presented in [211], called WhereNext, that uses a pattern based methodology to predict the next cell of a movement. This is a global approach and considers all the trajectories to generate *trajectory patterns* that contain the information on the travel time between two consecutive cells. The global method differs completely from the collective one, which combines the set of individual profiles because only the behavior followed by the crowd will survive to the process of extraction. WhereNext is just able to predict the next cell and the time spent on average for moving from the current cell to the next one. Since even this method applies a spatial discretization, we use the same grid defined above. The goodness of predictions got by WhereNext depends on the quality of trajectory patterns and on the *minimum score* used to consider admissible a prediction. In Fig.9.18 the results of this competitor are shown for the different grids. Comparing them with *MyWay*, we can see that our *individual strategy*, in general, performs better than WhereNext in terms of both accuracy and prediction rate. While considering the *collective strategy* we pay the increasing of the prediction rate - 40% greater than WhereNext - with decreasing of accuracy which let the competitor win using the 250 grid with an advantage of 5%. This disadvantage disappears if we compare WhereNext with the *hybrid strategy*.

Comparison Summary. To better appreciate the encouraging performances of *MyWay*, we compare the various methods in Fig. 9.19. To this end, we select a set of parameters producing a fair evaluation. In particular, we analyze the performances for 2 and 5 minutes of look ahead by observing a unique value obtained as composition of the prediction rate and accuracy. Thus, this value is very high when both of these factors are high while is low when just one between prediction rate and accuracy is low. We observe the values obtained using a $spat_{tol}$=250 (left) and $spat_{tol}$=500 (right) for *MyWay* while for the competitor we use the upper bound cells 250^u and 500^u. For a small look ahead like 2, *MyWay* performs dramatically better than the competitors. On the other hand, looking 5 minutes in the future is more difficult and leads to a decrease of the performances. However, with the exception of the individual strategy performing few points worse than [10] when considering $spat_{tol}$ = 250, the performances of *MyWay* remain higher than the others and the hybrid approach performs clearly better than everything else.

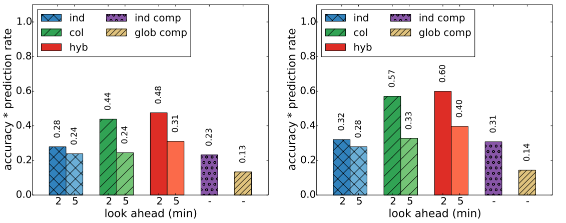


Figure 9.19: Performance comparison for *prediction rate* times *accuracy* for $spat_{tol}$=250 (left) and $spat_{tol}$=500 (right) in $test_{66}$. Parameters: *MyWay* γ=500 m, $temp_{tol}$=30 sec, *Individual competitor sequence length*=4, *Global competitor min score*=4.

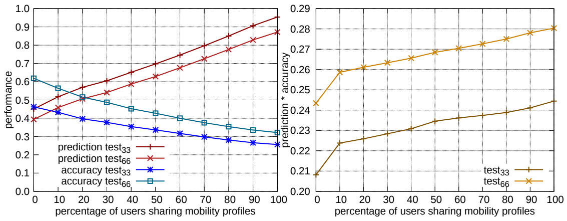


Figure 9.20: Increasing the participation of the users the prediction rate increase loosing some accuracy (left), but the overall performance rises (right).

9.1.3 Participation Analysis

In the experiments we showed before we considered a complete participation of the users, but in reality the users of the PDE may choose: *(i)* to contribute to the *MyWay* service by sharing their profiles with the collectivity in order to obtaining a better service when using the hybrid strategy or *(ii)* to maintain their profile private using only the individual strategy. Therefore, we study how the participation of the user effects the overall performances by analyzing the prediction rate and the accuracy varying the percentage of users sharing their profiles. Fig.9.20 shows the two measures and the overall performances in the two test cases. This result fills the gap between the individual strategy and the hybrid one which are represented by 0% and 100%, respectively. We observe how a greater sharing of routines enables better performances. The prediction rate dramatically increases at each step, while the accuracy slightly decreases. This happens because a larger number of trajectories become predictable allowing more errors, but the overall performances clearly improve.

From the complexity point of view, more users of the PDE participate, more routines will be part of the model. As discussed in the computational analysis paragraph for the collective model the routines are related only to user's trajectory and computed locally, therefore the increase will be limited. This is not true for the global one where adding a new user the complexity increase due to the fact that each new trajectory must be compared with the entire set (theoretically). Another aspect to be considered is the storage of this information, even in this case the collective model must store only the new routines while the global one needs all the trajectories for further recomputation.

9.1.4 Conclusion

We have presented *MyWay*, a Personal Data Analytics system for predicting future positions of mobile users at specific time instants. It is based on three strategies that exploit the individual systematic behaviors of users in the daily mobility, described by their *individual mobility profiles*. The *individual strategy* takes advantage of the single user's regularity; the *collective strategy* exploits individual systematic behaviors of all users, and the *hybrid strategy* combines both of them using two levels of knowledge (individual and collective). We have evaluated our prediction strategies on large real-world trajectory data. Our experiments show that the best prediction strategy is the hybrid one, i.e., the collaboration of the users in the PDE improves the level of individual awareness with a collective awareness, and this raises the performance of the prediction system.

9.2 Towards an Experienced Route Planner

Route planners help users to select a route between two locations. When providing directions, online mapping services generally suggest the shortest route. Popular route planners such as Google Maps, Open Street Maps etc. generate diverging directions using powerful libraries of roads and road attributes [318]. However, they often ignore the preferences of the users they serve and the paths followed by users living in the area of interest. Since cities are becoming more and more crowded, smart route planners are gathering an increasing interest. In such a context, a route planner which takes into account personal users' preferences [185], and which exploits the crowd expertise in order to identify the best route, can be more desirable and helpful than an ordinary route planner [117].

A route planner service which exploits Personal Mobility Data Models to improve the planning can have a real advantage if commuting users do not follow the shortest path in their systematic movements but deviate from them. Consequently, we tried to understand and to estimate how much the systematic movements of a user are different from the shortest paths between the origin and destination locations [128]. The intuition is that a user which lives and acts in a certain territory do not automatically select the shortest path. This can be due to the user's experience of traffic conditions and roads quality, for passing close to the cheapest petrol station, for avoiding roads with control of speed, etc. However, independently from the reasons, if there is a divergence between the systematic route with origin point o and destination point d, and the shortest route from o to d suggested by a route planner, then also other users could benefit from this kind of knowledge which comes from individual expertise in a certain area. Therefore, we present an *experienced route planner* considering PMDM that can propose as alternatives to the shortest path a route frequently followed by a user among those in the PDE which participate in the route planning service.

9.2.1 Trajectory Map Matching Background

Given a trajectory m defined as in Section 8.3, generally does not contain the relative traversed road network segments. This lack of information can be restored by means of some *map matching* techniques. We adopted the *gravity model* [78] as method to match each single trajectory point to the road segment it belongs to:

Definition 28 (Gravity Force Attraction). *Given a point p_i and a set of road segments describing the road network $S = \{s_1, \ldots, s_r\}$ where $s_j = \{p_{start}, p_{end}\}$, we define the* gravity force attraction *of a segment s_j for a point p_i as:*

$$GFA(p_i, s_j) = w_{(p_i,s_j)} = w^d_{(p_i,s_j)} \cdot w^\theta_{(p_i,s_j)}$$

where $w^d_{(p_i,s_j)} = 1 - \frac{\mathrm{dist}(p_i,s_j)}{\sum\limits_{s_k \in S} \mathrm{dist}(p_i,s_k)}$, $w^\theta_{(p_i,s_j)} = 1 - \frac{\mathrm{ang}(p_i,r_j)}{\sum\limits_{s_k \in S} \mathrm{ang}(p_i,s_k)}$, dist *is the euclidean distance between a point and a segment, and* ang *is the absolute difference between the direction of the point and the direction of the segment.*

This gravity model can be applied over the whole road network segments S. However, S is generally very large. Hence, it is possible to use a *nearest neighbor* approach and consider only a subset $S_k \subset S$ containing the k segments closest to a given point.

Given a GPS trajectory $m = \{p_1, \ldots, p_n\}$ and a set of road segments S, it is possible to assign each point p_i to the segment with the most powerful force $\bar{s_j} = \sigma(p_i, S, k) = argmax_{s_j \in S_k}(GFA(p_i, s_j))$. The Gravity Model adopted can also be used to estimate the travel time of each matched road segment; once every trajectory point has been matched, the typical travel time of a segment s, given P the set of points matched to s, is defined as

$$\frac{\sum p_i \in Pspeed(p_i) * GFA(p_i, s)}{\sum p_i \in PGFA(p_i, s)}$$

Definition 29 (Trajectory Map Matching). *Given a trajectory m and a set of road segments S, we refer to m^* as the trajectory m on the road segment network S, i.e. the points of m^* belong to the segments in S:*

$$m^* = mapmatch(m, S, k)$$

where $m^* = [p_1^*, \ldots, p_n^*] = [\bar{s_1}, \ldots, \bar{s_{n-1}}]$ and $[p_i^*, p_{i+1}^*] = \bar{s_j} = \sigma(p_i, S, k)$

Thus, trajectory m can be transformed into the map matched version $m^* = [p_1^*, \ldots, p_n^*]$ containing points which belong to the road segments S, where $p_1^*, \ldots, p_n^*$ maximize the attractions with $p_1, \ldots, p_n$, i.e., m^* is the best representation of m on S (with $k > 0$). A refinement is needed to obtain the map matched trajectory m^*, i.e., a path must be added for each couple of points which are not directly connected. To find such path, we used a Time-Aware heuristic as described in [77]. This map-matching method takes the GPS travel time between the two consecutive GPS points as input and returns the path connecting the two points that better fit the input travel time. It is worth to consider that the road network is a directed graph, thus including only one-way segments.

9.2.2 Model

In the following we describe the analytic model adopted. Note how we exploit part of the Personal Mobility Data Model described in Section 8.3.1, i.e., the Personal Agenda. Given a set of users U and a set of road segments S, for each user $u \in U$, we calculate the individual mobility profile P_u. In this work we focus on the routines R_u [285]. Then for each routine $r_i \in R_u$, we map match the routine on the road network $r_i^* = mapmatch(r_i, S, k)$. We name *map matched routines* $R_u^* = \{r_1^*, \ldots, r_k^*\}$ the routines of a user u mapped on S.

Given an origin point o and a destination point d, we define a route planner $\bar{m} = routeplanner_{type}(o, d, S)$ as a function which returns the best path $\bar{m} = [o, \bar{p}_2, \ldots, \bar{p}_{n-1}, d]$ with respect to the type of search $type \in \{s, f\}$ (where s stands for *shortest* and f stands for *fastest*) on the road segments S. Then, for each routine $r_i^* = [o_i, \ldots, d_i] \in R_u^*$ we calculate the path returned by the route planner $\bar{r}_i = routeplanner_{type}(o_i, d_i, S)$ on the origin and destination. We indicate with $\bar{R}_u^{type} = \{\bar{r}_1, \ldots, \bar{r}_k\}$ the *shortest/fastest routes* of a user u, containing the paths returned by the route planner.

Summing up, given a set of users U, their individual history $H_u \forall\, u \in U$, and the road network segments set S we obtain:

1. $R_u\ \forall\, u \in U$ with the *routine* step extraction from the Personal Mobility Data Model P_u calculated on H_u for each $u \in U$;

2. $R_u^*\ \forall\, u \in U$ through the *map matching* step as result of the application of *mapmatch* for each $r_i \in R_u$, $\forall\, u \in U$;

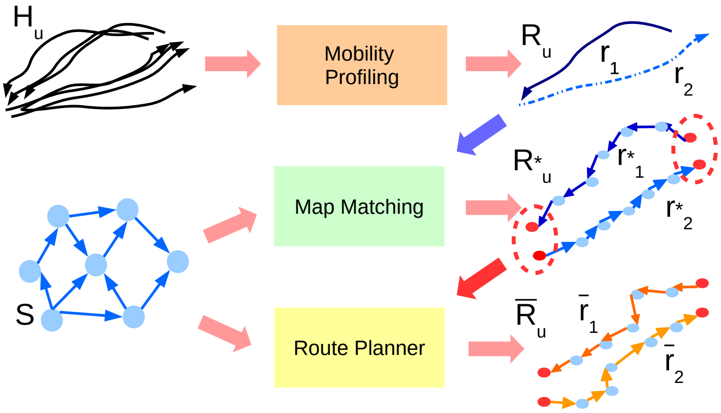


Figure 9.21: Steps of the analytic mobility model. Input: individual history H_u, road network segments set S. Output: personal map matched routines R^*_u, personal shortest/fastest routes $\bar{R}^{type}_u$. R_u is extracted from the Personal Mobility Data Model P_u with the *Mobility Profiling* module, through *Map Matching* we obtain R^*_u, and $\bar{R}^{type}_u$ is obtained by using the *Route Planner* on the origins and destinations (highlighted in the red dotted circles) of the routines in R^*_u.

3. $\bar{R}^{type}_u$ $\forall$ $u \in U$ by means of the *route planner* step as result of the application of *routeplanner* on the origin and destination points o_i, d_i for each $r^*_i \in R^*_u$, $\forall$ $u \in U$.

Fig. 9.21 shows the steps of the analytic mobility model. In the next section we will observe the differences between R^*_u and $\bar{R}^s_u$, $\bar{R}^f_u$. We remark that the *shortest path* is the path which minimizes the distance, the *fastest path* is the path which minimizes the travel time.

9.2.3 Case Study

In the following, we evaluate how much systematic users described by their map matched individual mobility profile R^*_u deviate from the shortest and fastest routes contained in the shortest mobility profile R^s_u and fastest mobility profile R^f_u. Moreover, we analyze which are the nodes on the road network S, the areas and the flows more affected by deviations. We accomplish these analyses by considering the trajectories passing through the provinces of Pisa and Florence on the *Octo* mobility dataset described in Section 6.1.

Deviation Analysis

We analyze the deviation in terms of space between the routines in R^*_u and the shortest routes in $\bar{R}^s_u$, and the deviation in terms of time between the routines in R^*_u and the fastest routes in $\bar{R}^f_u$. For each user $u \in U$, for each routine in $r^*_i = \{o_i, \ldots d_i\} \in R^*_u$, we calculate the difference with the corresponding route in $\bar{R}^{\{s,f\}}_u$, i.e. the route $\bar{r}_i$ which starts in o_i and ends in d_i. Note that the following results are biased by the route planner used.

In Fig. 9.22 we can observe the space and time differences distributions. With respect to the shortest path *(left)*, in both dataset there is a consistent set of routines with space difference equals to zero. This indicates that 30%-35% of the routines (for Pisa and Florence respectively) follow the shortest path suggested by the route planner. The remaining routines differentiate on average of 7 km (see Tab. 9.2). On the other hand, in Fig. 9.22 *(center)* none of the routines follows exactly the fastest path. Just a few routines, i.e. the 10%, follow the fastest routes with less than a minute of difference. All the others differentiate consistently (20 min on average Tab. 9.2). In addition, we can observe that

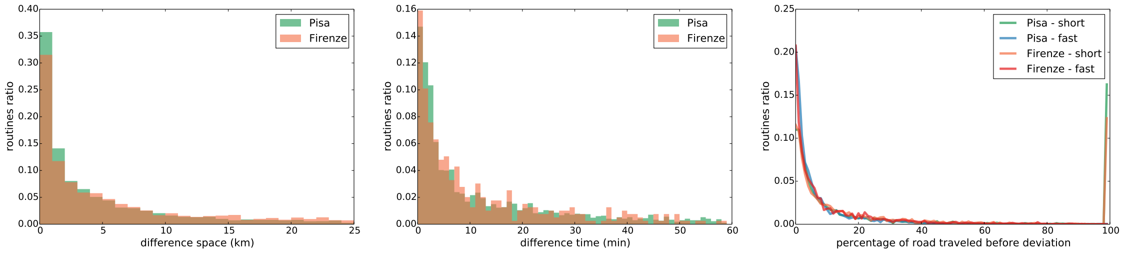


Figure 9.22: *(Left)* Space difference distribution in km between the routines in R^*_u and the corresponding routines in $\bar{R}^s_u$. *(Center)* Time difference distribution in minutes between the routines in R^*_u and the corresponding routines in $\bar{R}^f_u$. *(Right)* Distribution of the percentage of road traveled before the routine deviates from the shortest/fastest path.

	short - space diff			fast - time diff			short - pbd			fast - pbd		
	med	avg	std	med	avg	std	med	avg	std	med	avg	std
Pisa	02.31	07.16	13.56	07.42	26.92	58.13	07.07	25.14	35.52	07.96	23.19	32.33
Florence	03.64	10.22	18.45	07.31	19.06	29.90	02.97	07.58	13.54	01.05	01.58	21.58

Table 9.2: Median, average and standard deviation of the space difference (km), time difference (min) and relative percentage of road traveled before the deviation (*pbd*).

the routines of the 15% of the drivers in Pisa and 10% in Florence correspond to the shortest routes (R^*_u=$\bar{R}^s_u$), but none of the users has all the routines equal to the fastest path.

In Fig. 9.22 *(right)* is reported the percentage of road traveled before the deviation (*pbd*). It is obtained by observing after how much r^*_i deviates from $\bar{r}_i$ after the start point o_i. We can notice how 20% of the systematic movements deviate from the shortest/fastest routes at the very beginning. The distribution is a long-tailed power law with average percentage before deviation of 7% and 3% for Pisa and Florence respectively (see Tab. 9.2). Furthermore, how already observed, there is a consistent subset of routines (12-15%) which do not deviate from the shortest path. This does not occur for the fastest path.

Hence, systematic drivers generally deviate from the routes suggested by a route planner at the very beginning of their movements, and do not optimize their travel time but try to minimize the travel distance. However, even if the drivers deviate from the short/fast routes, these routes are in many cases very similar to the personal routines.

Towards an Experienced Route Planner

Before presenting the analysis of this section we remark that routines are frequent movements during the observation period. Thus, if drivers systematically deviate from what is supposed to be the shortest (or the fastest) path there should be a valid reason. Given a user moving for the first time in a certain area, it could be better for her to follow the routines described by "expert driver" instead of the routes suggested by a route planner.

A route planner could be boosted through the experience given by personal mobility models. Such a route planner should consider various information: *(i)* the road *intersections* where the systematic drivers deviate more, *(ii)* the *areas* where those intersections are concentrated, and *(iii)* the main *flows* of movement containing deviations. In the following we analyze these three factors to understand their impact and which are their possible uses. In particular, we focus on the deviation of the routines against the shortest path.

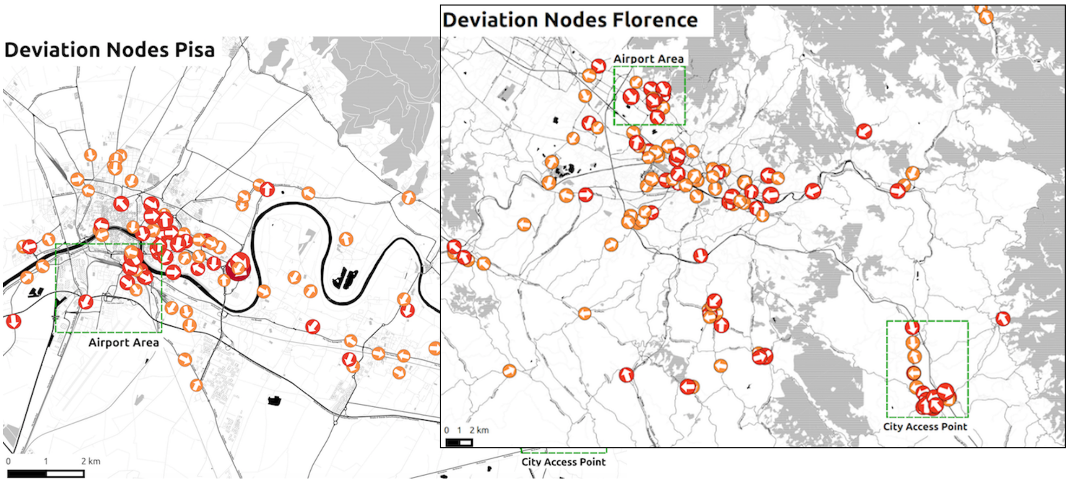


Figure 9.23: Deviation nodes supported by with at least 100 deviations.

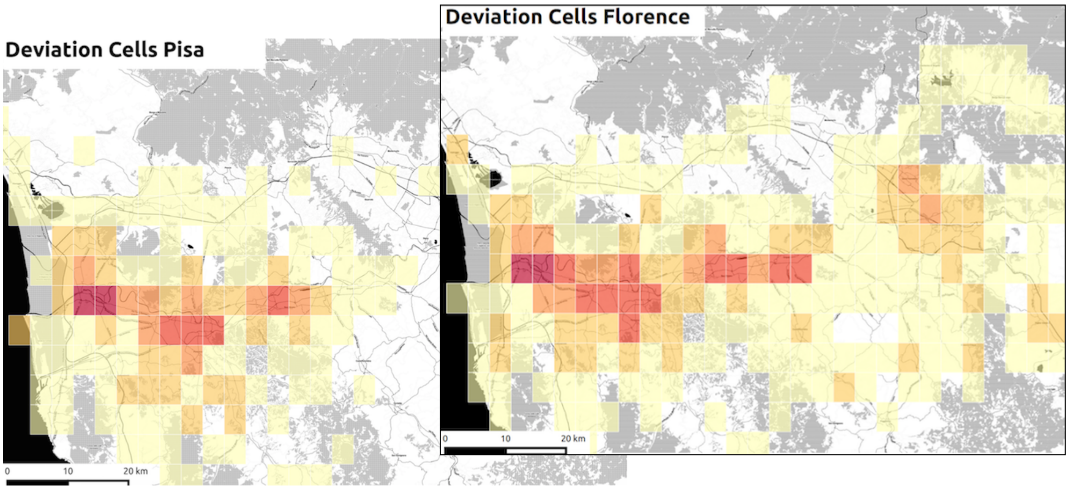


Figure 9.24: Heatmap of the deviation cells.

We refer to the road intersections as *deviation nodes*. They correspond to the first nodes in the set of road segments S from which the routines in R_u^* deviate from the route in $\bar{R}_u^s$. To count the number of deviations, instead of considering only the number of routines, we weight each routine $r_i^* \in R_u^*$ with the number of trajectories that support it. In Fig. 9.23 we can observe the deviation nodes in which there are at least 100 trajectories that deviate. The darker and the bigger is a marker, the higher is the number of deviations performed by the routines on that node. As expected, for both cities, the highest numbers of deviation nodes appear into the city center. This confirms the fact that in the city is very difficult to follow the shortest paths. Moreover, for both provinces we can observe some particular areas not in the city center (those highlighted in the green dotted squares) with a high number of deviations. They correspond *(i)* to the main access points to/from the city center, and *(ii)* to the roads close to the airports. This is a signal that these areas are probably affected by consistent traffic and the systematic users which have to pass through them prefer longer but less stressful routes.

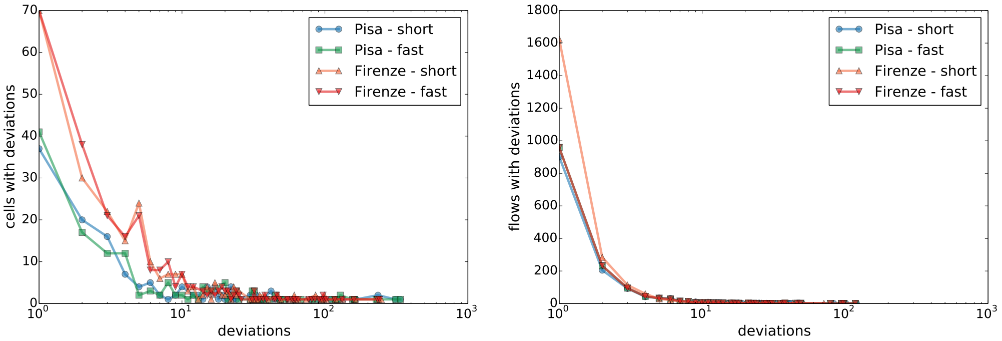


Figure 9.25: Number of cells with deviations (left), and number of flows with deviation (right).

To analyze the *deviations areas* we divide the territory using a grid with cells of 2.5 km of radius. The heatmap of the deviations is shown in Fig. 9.24. The darker is a cell, the higher is the number of trajectories which support the routines deviating there. For these images no filters are applied. The first insight is that the users acting in province of Florence have an active role even in the mobility of Pisa but the viceversa is not true. Indeed, most of the cells with more deviation in Pisa occur also in Florence. From the intersection of the two images emerges that most of the systematic deviations take place along the main road between Pisa and Florence (named SGC Fi-Pi-Li) with a concentration in the area around Empoli. This probably happens because most of the people living in Empoli, which is in province of Florence, go systematically to Pisa for working. For example, instead of following SGC Fi-Pi-Li that is a highway but has a lot of traffic, many drivers could prefer as alternative the road SS67 which runs along SGC Fi-Pi-Li but has much more turns and is not a highway. In Fig. 9.25 *(left)* we report the distribution of the number of cells per routines' deviations. It is a power low distribution indicating that there are few cells where most of the systematic users decide to take alternative routes. Those are the cells that more than the others the experienced root planner should consider when suggesting the routes.

We define a flow as a triple of cells $(origin, deviation, destination)$ where *origin* is the cell origin of the routine, *deviation* is the cell where r_i^* deviates from $\bar{r}_i$, and *destination* is the ending cell of the routine. In Fig. 9.26 we can observe the flows containing the routines supported by at least 100 trajectories. Through this approach we can observe the main flows along with most of the drivers deviate from the shortest paths. We can observe how in Pisa province there are various flows of entrance to and exit from the city center. The flows with more deviations (the purple biggest arrows) are just under the city center starting from the airport area up to the suburbs. They are surrounded by a large number of in-coming and out-coming flows. We remark that in many cases the deviation from the shortest path appears at the very beginning of the movement. Thus, the flows reported highlight the part of the movement after the deviation. Some deviation flows do not have a mutual reverse flow of the same importance. On the other hand, in province of Florence, the flows in the city center are on average shorter than those outside. In addition, the biggest flows are present in the airport area (big green arrow in the center) and close to the exit of the highways (big blue arrow bottom right and big aqua green arrow in the center). Fig. 9.25 (right) shows the distribution of the number of flows per routines' deviations. Similarly to the cells, the distribution is long-tailed indicating a small set of flows where many routines deviate from the shortest/fastest path.

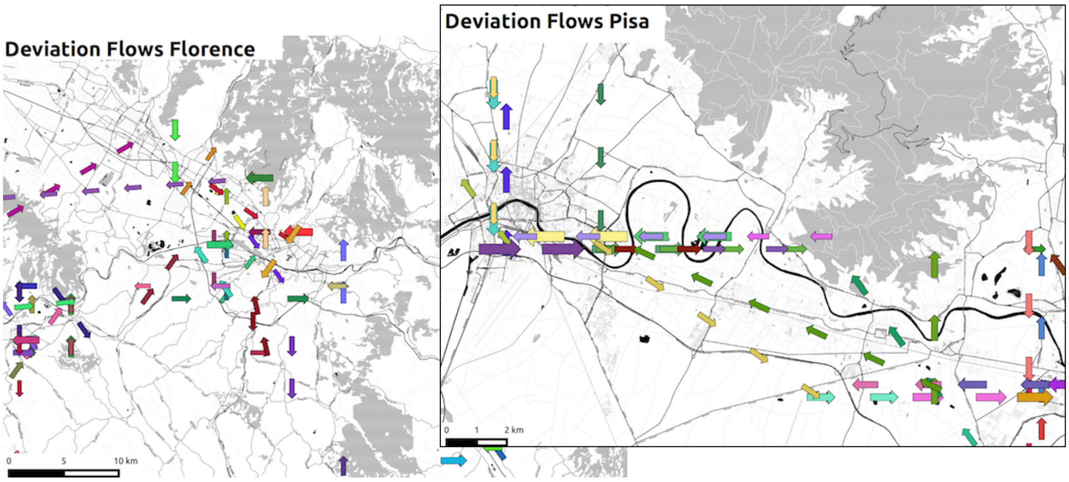


Figure 9.26: Deviation flows supported by at least 100 deviations.

Finally, we analyze the difference between the flows described above and the flows built using only origins and destinations. In other words, given an origin-destination flow (*origin*, *destination*) how many flows (*origin*, *deviation*, *destination*) pass through the same *deviation*? We name this indicator *flow similarity in deviation*. This value gives us a hint of how much a certain deviation is stable along a flow. A flow similarity in deviation of X% indicates the percentage of $(origin, deviation, destination)$ flow on the number of origin-destination flows $(origin, destination)$ which pass through the same *deviation* cell. E.g. given the following origin-destination flows $\{A \to B, X \to Y\}$ and the flows $\{A \to C \to B, A \to C \to B, A \to D \to B, X \to Z \to Y, X \to Z \to Y\}$, then the percentage of flow difference is 80%. We obtain the following results: *Pisa*: 83% (short), 78% (fast), *Florence*: 87% (short), 85% (fast). These high percentages are a clear signal that the deviations along the various flows are not a matter of individuals, but that are known and subscribed by the majority of the drivers. It is a sort of "common sense" which surprisingly emerges at collective level even though all the mobility models used in the proposed analysis are personal.

9.2.4 Conclusion

By exploiting Personal Data Analytics in the Personal Data Environment of shared knowledge of routines, we have analyzed the deviation of the systematic movements from the shortest and fastest paths suggested by a route planner on a set of drivers in Pisa and Florence provinces. We have found that systematic drivers deviate from the routes suggested by a route planner at the very beginning of their movements, and that they generally try to minimize the travel distance more than the travel time. Moreover, we have observed that the shortest paths are in many cases very similar to the systematic movements from which they deviate. Furthermore, through our model we have been able to select the areas and the flows with the highest number of systematic deviation and we have discovered that given a flow from an origin o to a destination d nearly all the users which systematically move from o to d deviate in the same area. Our analysis has shown that the traveled systematic movements give to the drivers a feeling that their route is better than the shortest or fastest paths suggested by a route planner.

Chapter 10

Social and Proactive Carpooling

There is no need to advocate why traffic and its consequences on the environment, our health and quality of life, and the economy is a major problem for our society. Carpooling, i.e., the act where two or more travelers share a vehicle in order to reach common or nearby destinations, is an old idea brought forward, among many others, to reduce traffic and its externalities. If a large proportion of travelers, especially daily commuters, would adopt carpooling, a substantial traffic reduction could indeed take place.

Despite its clear advantages in reducing costs, pollution, and time spent in finding a car park, there are still a few obstacles that prevent it from being the preferred way to move: safety of passengers, sub-optimal mobility matches, and time flexibility, among others. Indeed, it has been shown in the literature that it is extremely difficult to boost the adoption of carpooling to levels that significantly diminish traffic as a whole. There are many reasons why this happens: psychological, organizational, technological. As a matter of fact, we do not know much yet about the real carpooling potential that emerges from people's mobility. However, through Personal Data Analytics, we propose very preliminary steps towards designing the right mechanisms and incentives for successful carpooling systems.

Nevertheless, we now have access to the data to observe personal mobility at a microscopic level and for large populations of travelers, such as the digitized trajectories of vehicular travels recorded by GPS-enabled onboard devices. By exploiting the Personal Mobility Data Models of the users in the Personal Data Ecosystem that want to participate in a carpooling service, we developed different frameworks which try to reduce the number of circulating cars and, in particular, to reduce the systematic trips.

In this Chapter, in contrast with "on-demand" carpooling setting, where the user typically has to select origin, destination and departure time, in order to find matching drivers, we process data in temporal batches and focus on recurring trips by providing *proactive* suggestions which minimize the number of drivers driving alone. Moreover, since the obstacles for employing carpooling are related not only to mobility, but also to *social* aspects like the reticence to share the car with unknown people, we introduce in our system a measure to take into account these facts when we assign the passengers to the drivers.

10.1 Boosting Carpooling with Network Analysis

We analyze the potential impact of carpooling as a collective phenomenon emerging from people's mobility, by *network analytics*. By starting from the routines R_u of the PMDM, we construct the *network of potential carpooling*, where nodes correspond to users, and each link between user u and user v corresponds to the fact that u can take a lift from v, because there is a trip in v's profile that can serve u, i.e., u can be a passenger of driver v. In this carpooling scenario, the network is formed by the participating users of the PDE.

By analyzing the structural properties of this network, we can gain a deeper insight of the potential impact of carpooling [133]. We adapt network analysis tools such as community discovery and node ranking to the purpose of highlighting the subpopulations of travelers that have higher chances to create a carpooling community, and who are the users that show a higher propensity to be either a driver or a passenger in a shared car. Also, we can reason about the propensity of geographical units or cities to carpooling, as well as on the impact on externalities such as CO_2 emissions and costs that can be potentially reduced. Remarkably, our method explores the potentiality of carpooling in *systematic* travels, e.g., home-work commuting, as opposed to ride sharing in occasional trips, which is the approach of several popular works (see Section 3.2.1). Addressing the issue of sharing systematic trips is more challenging and can have a larger impact on traffic reduction.

An additional contribution of our study is the analysis of the potential aggregated outcome of a carpooling service in the networks considered, using several empirical simulations, in terms of expected number of single occupancy vehicles (SOV) that we observe as a result of carpooling matches that take place. We investigate several possible scenarios, and show how a carpooling assignment that exploits the mentioned network analytic concepts of communities and node rankings is the one with the best performance. Although much further work is needed to validate in the real world that mining carpooling networks can boost the adoption of ride sharing among communities of commuters, our study is a first in-depth analysis of the potential impact of the approach, which sheds a new, quantitative view of a mechanism that can only be explained in terms of a dynamic network of interacting actors exhibiting an often surprising aggregated behavior.

10.1.1 Complex Network Analysis Background

In the following we make use of three concepts belonging to complex networks analysis: *(i)* node degree, *(ii)* link analysis, *(iii)* community discovery. Let G be a directed graph and i a node of G, we define the *incoming degree* of i as the number k_i^{in} of links that point to i, and the *outgoing degree* as the number k_i^{out} of links that start from i and point to other nodes.

In network science, *link analysis* is a data-analysis technique used to evaluate relationships, i.e. connections, between nodes. In particular, we used Hyperlink-Induced Topic Search (*HITS*), also known as *hubs and authorities*, a link analysis algorithm that rates Web pages, developed in [100]. The algorithm assigns two scores to each page: its *authority* score, which estimates the value of the content of the page, and its *hub* score, which estimates the value of its links to other pages. Authority and hub values are defined in terms of one another in a mutual recursion: authority values are computed as the sum of the hub values that point to that page; hub values are the sum of the authority values of the pages it points to. These hub and authority scores are values that enable us to rank nodes according to some criteria. We define *HITS* as a *ranking function*:

Definition 30 (Ranking Measure)**.** *Given a direct graph $G = \langle N, E\rangle$, we define the ranking function* ranking(G) *as the algorithm* HITS*, taking as input G and returning two score vectors h and a, respectively for* hub *and* authority*.*

Community discovery is the problem of identifying communities hidden within the structure of a complex network [83]. A *community* is a set of entities that, in the network sense, are closer to the other entities of the community than with those outside it. Thus, communities are groups of entities that share some common properties and/or play similar roles. In literature, several popular community discovery algorithms exist [40, 84, 254]. Among them, in this work we choose to adopt *Demon* for its ability to deal with direct graphs and for the quality of the communities extracted.

Definition 31 (Community Discovery)**.** *Given a direct graph $G = \langle N, E\rangle$, we define the function* communities(G) *as the algorithm* Demon*, taking as input G and returning a set of communities $\mathcal{C} = \{C_1 \dots C_n\}$, where $C_i \subseteq N$ is a set of nodes.*

10.1.2 Method

In this section we describe an approach for realizing a proactive carpooling service based on the identification of pairs of users that could share their vehicle for one or more of their systematic trips. We propose a procedure for suggesting carpooling assignments among systematic users, i.e., recommending to drivers that frequently follow the same routes to offer a ride to other users who will become their passengers. The output of such procedure also provides the means for studying the potential of carpooling on the area of analysis. The procedure is composed of two main tasks. The first one regards the construction of the carpooling network, the calculus of the ranking scores and the extraction of the communities. The second one concerns the actual assignment of drivers and passengers among the users that form the carpooling network by exploiting the ranking scores and the communities discovered.

Carpooling Network Construction

We talk about *carpooling interaction* when a user can get or offer a ride to another one. The idea is to use complex networks to model the potential carpooling interactions, use the ranking measures to evaluate how much a user is suitable for being a driver or a passenger, and employ community detection to characterize groups of users highly related in terms of carpooling. The starting point of this analysis is the set of routines R_u part of the user Personal Mobility Data Model P_u. Since mobility profiles represent users' systematic behaviors, by comparing them it is possible to understand if a user can be served by another one. The system can keep reasonably up-to-date routines by executing the profiling process regularly, for instance every week, over the most recent mobility data.

A basic operation is to test whether a routine is *contained* in another. If r_1 is contained in r_2 then the user systematically following r_1 can potentially leave her car at home and travel with the user systematically following r_2. We define *routine containment* as:

Definition 32 (Routine Containment)**.** *Given two routines $r_1 = \{(x_1^{(1)}, y_1^{(1)}, t_1^{(1)}), \dots, (x_n^{(1)}, y_n^{(1)}, t_n^{(1)})\}$ and $r_2 = \{(x_1^{(2)}, y_1^{(2)}, t_1^{(2)}), \dots, (x_m^{(2)}, y_m^{(2)}, t_m^{(2)})\}$, a* spatial tolerance *$spat_{tol}$*

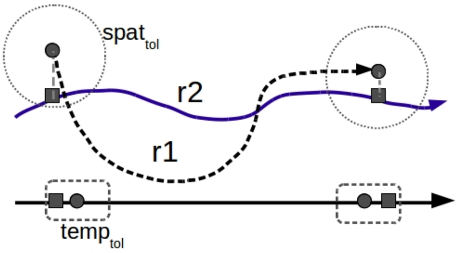


Figure 10.1: Example of routines containment: r_1 *is contained* in r_2 because the starting and ending points of r_1 (circles) are spatially and temporally close enough to some points of r_2 (squares).

and a temporal tolerance *temp$_{tol}$, we say that* r_1 is contained in r_2*, i.e.*

$$\mathit{contained}(r_1, r_2, \mathit{spat}_{tol}, \mathit{temp}_{tol}), \mathit{if}\ \exists\ i, j 1 \leq i < j \leq m\ \tilde{A}\check{e}\mathit{such\ that}$$
$$||(x_1^{(1)}, y_1^{(1)}) - (x_i^{(2)}, y_i^{(2)})|| + ||(x_n^{(1)}, y_n^{(1)}) - (x_j^{(2)}, y_j^{(2)})|| \leq \mathit{spat}_{tol} \wedge$$
$$|t_1^{(1)} - t_i^{(2)}| + |t_n^{(1)} - t_j^{(2)}| \leq \mathit{temp}_{tol}$$

where:

- *spat$_{tol}$* is the maximum total distance that the *served* user could walk to reach the pick-up point, and to reach her final destination from the get-off point;
- *temp$_{tol}$* is the maximum total amount of time that the *served* user is allowed to waste, as delay or anticipation of her systematic trip, considering departure and arrival time.

It is important to note that the *contained* relation is not symmetric, since a routine might include another one without having the vice versa holding. This can happen when the routines compared have different lengths, in which case the origin of the user which *serves* the other can be very far from the origin of the one who is *served*, and similarly for the destination point. Fig.10.1 provides a visual depiction of the containment relation over a simple example. This formulation assumes that the users *served*, i.e., the candidate passengers, are willing to walk and change their time schedule in exchange for the ride they get, while the users which *serve*, i.e., the candidate drivers, do not change their routine.

Using the *containment* relation we can build the *carpooling network* $G{=}\langle N, E\rangle$. Given the set of collective routines of all the users $R_C{=}\{R_1, \dots R_n\}$, for each pair of different users u and v, we check the routine containment between every routine $r_i^u{\in}R_u$ and every routine $r_j^v{\in}R_v$. If $\mathit{contained}(r_i^u, r_j^v, \mathit{spat}_{tol}, \mathit{temp}_{tol})$ holds, then $u{,}v{\in}N$ and $\{(u, v, r_i^u, r_j^v)\} \in E$.

Definition 33 (Carpooling Network)**.** *A carpooling network $G = \langle N, E\rangle$ is a multi-dimensional graph where N represents the set of all users taking part in at least a carpooling interaction, E is the set of all labeled edges (u, v, r_i^u, r_j^v), where r_i^u is a routine of $u \in N$, r_j^v is a routine of $v \in N$, and r_i^u is* contained *in r_j^v.*

The *carpooling network* guarantees that, since the links considered are routines, the movements they represent are systematically repeated. This ensures that a ride is most likely available or needed. In Fig. 10.2 *(left)* we have an example of carpooling network. In practice, we follow the basic idea of creating a network based on containment relationships [8] with some spatio-temporal variations. Given a carpooling network G we define the *possible passengers* and *possible drivers* as:

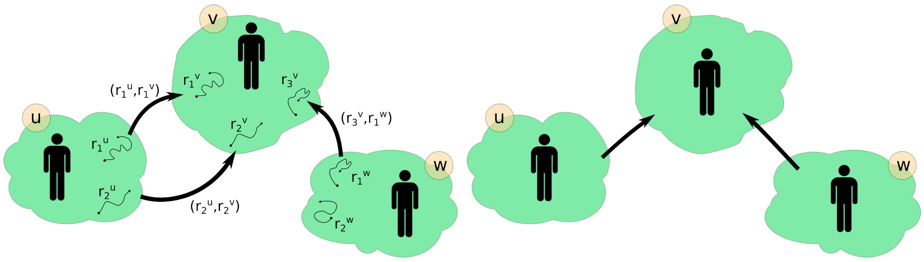


Figure 10.2: Carpooling Network (left), Carpooling User Network (right).

Definition 34 (Possible Passengers). *Given a carpooling network $G = \{N, E\}$, a user $u \in N$ is a* possible passenger *if she has at least an outgoing link, that is $k_u^{out} > 0$.*

Definition 35 (Possible Drivers). *Given a carpooling network $G = \{N, E\}$, a user $u \in N$ is a* possible driver *if she has at least an in-going link, that is $k_u^{in} > 0$.*

We denote with PP_G the set of all possible passengers and with PD_G the set of all possible drivers in G. Note that it is possible (and actually rather frequent) that $PP_G \cap PD_G \neq \emptyset$, thus some users can act both as possible passengers and as possible drivers.

Finally, it is worth to highlight that a carpooling network is in fact a multidimensional network: users u and v can share for example two routines; the going trip and the return trip because they take place at different times and also on different roads. However, in order to use some common network analytic tools we have to transform the carpooling network in a mono-dimensional network (see Fig. 10.2 *(right)*).

Definition 36 (Carpooling User Network). *Given a carpooling network $G = \langle N, E \rangle$, we define a* carpooling user network *as a direct mono-dimensional graph $G' = \langle N, E' \rangle$ obtained by collapsing all multi-dimensional edges between the same pair of users, i.e. $E' = \{(u, v) | (u, v, r_1^u, r_1^v) \in E\}$.*

Since G' is direct, then an arc (u, v) is directed from u to v, v is called *successor* of u.

Greedy Carpooling Assignment Suggestion

Using the carpooling network, we are able to extract potential assignments. The *carpooling assignment* method proposed in this section follows a simple heuristic and a greedy idea. The method takes as input a carpooling user graph G, i.e., multidimensional edges are not considered, assuming that each pair of users can share only one routine: the general case will be described later as an extension of the solution depicted here. This first procedure is applied to a relatively short time window within the day, where it is basically certain that each user will have at most one *active* routine, e.g. in a typical situation a time window covering the period from 8 a.m. to 8:15 a.m. might contain the home-to-work routine of a commuter, but not the symmetric one, which will likely appear in another time slot in the afternoon. In the following we describe the overall algorithm that iteratively applies the present one on different time slots. The output of the method is a classification of the users taking part in the carpooling network. In particular, the set D contains the drivers that host some passengers in their car, P contains the passengers that are hosted by some drivers, and S contains the single-occupant-vehicles (SOV) that drive alone. The three classes form a partitioning of the users, i.e. $N = D \cup P \cup S$ and $|N| = |D| + |P| + |S|$.

Algorithm 10: *calculateGeedyAssignment*$(G', f, m, c', c'', D, P, S)$

```
Input  : $G' = \langle N, E \rangle$ - carpooling user network, $c', c''$ - sorting criteria, $f$ - sorting
         function, $m$ - max number of free places, $D$ - set of sets of possible driver
         containing the assigned passengers (e.g. $D_v$ is the set of passengers assigned to
         driver $v$), $P$ - set of sets of possible passengers containing the assigned driver (e.g.
         if $v \in P_u$ it means that passenger $u$ is assigned to driver $v$, $|P_u| \leq 1$ always) $S$ -
         set of single occupant vehicle
Output: $D, P, S$
for $u \in f(N, c')$ do
    if $D_u \nsubseteq D \wedge P_u \nsubseteq P$ then
        for $v \in f(successors(u), c'')$ do
            if $|D_v| \leq m$ then
                $D_v \leftarrow D_v \cup \{u\}$;
                $P_u \leftarrow \{v\}$;
                break;
            end
        end
    end
end
for $u \in N$ do
    if $D_u \nsubseteq D \wedge P_u \nsubseteq P$ then
        $S \leftarrow S \cup \{u\}$;
    end
end
return $D, P, S$;
```

The procedure uses a sorting function f to order the *possible passengers* according to some criteria c'. It takes the first possible passenger u from the sorted list, and it orders her *possible drivers* using f according to another criteria c''. Then, it takes the first driver v that still has free places in her car, and assigns u to v. The procedure is repeated until every user is assigned, or there are no free places left. The *greedy assignment* method is illustrated in Alg. 10 where the function $successors(u)$ returns the set of successors of u.

We remark that the algorithm is intended to be applied iteratively on successive time windows, therefore it takes as input also the output sets obtained from previous iterations, in order to consider in the matching process all users that are not already and completely assigned. For example, if a driver has already used all her free places for an active routine, then she cannot take other passengers, and therefore she is not considered in the matching at the present iteration. On the other hand, a user that was classified as SOV for an active routine can still be considered both as a possible passenger and as a possible driver.

The main purpose of this procedure is to reduce the number $|S|$ of systematic cars in which the driver is driving alone and, in second instance, the total number of systematic cars in circulation given by $|D| + |S|$, thus increasing the number of systematic cars that are not needed anymore – corresponding to the number of users that turned into passengers, $|P|$. The most important component is represented by $|S|$, since SOVs do not play an active role in carpooling although they could potentially share at least one routine with another user. The algorithm is parametric with respect to the sorting criteria used.

Although the algorithm has a quadratic complexity, in practical cases it is essentially linear in the number of nodes analyzed, $O(|N|)$. This happens because if a node has already been visited, then it cannot be re-analyzed. Also the presence of an inner loop does not lead to quadratic complexity because this would mean that every possible driver could offer a lift to all (or a large part of) possible passengers, which is highly improbable. Moreover, we have to consider the cost of the sorting functions f, which is $\Theta(NlogN)$ in the worst case. The cost of the innermost sorting function could be at worst $\Theta(N^2logN)$ but, as above, this would happen if every node links to all the others. In practice, the innermost sorting function f cost is $O(k_u^{out}logk_u^{out})$ each time it is repeated, i.e. $O(Nk_u^{out}logk_u^{out})$. Since the average k_u^{out} is very low in this kind of networks, we have that $O(k_u^{out}logk_u^{out})$ can be approximated to a constant c. Thus, the dominant cost remains $\Theta(NlogN)$.

The problem analyzed is NP-complete [171], and an optimal approach to solve it is exponential in the number of edges. Indeed, such an approach should take into account the fact that every assignment might inhibit any of the others, since each node in the network can either be a driver or a passenger and once the choice is made it cannot be reversed, then virtually all combinations must be tried in order to find the best one. Finally, we note that, in spite of its resemblance with bipartite matching, our formulation of the carpooling problem cannot be solved just using a maximal matching over the bipartite graph among possible drivers and possible passengers, because the intersection between possible drivers and possible passengers is not empty. Thus, in order to reduce it to the bipartite case, we should evaluate the matching over all its possible bipartite projections, i.e. by assigning all users to one fixed role, trying all possible combinations. That is computationally equivalent to the exhaustive, brute force approach mentioned above. For these reasons, the solution we propose is a heuristics, which trades optimality for scalability.

Ranking Criteria and Problem Partitioning

In order to find the best assignments among the users taking part in the carpooling scenario, it is useful to discover the best passengers and the best drivers among the candidate ones. We say that a user is a "good passenger" if she can accept a lift from many "good drivers", and mutually, a user is a "good driver" if she can offer a ride to many "good passengers". Thus, we analyze the carpooling network to rank a user as a "good passenger" or as a "good driver". The idea to reach this goal is to consider the carpooling user graph and the apply the *HITS* algorithm [168]. Indeed, the HITS task of extracting hub and authority scores to estimate the value of a web page can be directly mapped to the carpooling scenario for measuring how much a user is suitable for being a good passenger or a good driver. In the context of carpooling networks, we define the hub score as *passengerness*, i.e. the attitude of u for being a good passenger, and the authority score as *driverness*, i.e. the attitude of u for being a good driver.

Definition 37 (Passengerness and Driverness)**.** *Given the carpooling user network* $G = \langle N, E \rangle$ *and its adjacency matrix* A*, for each user* $u \in N$*, we define* passengerness p_u *and* driverness d_u *respectively as the hub and authority scores of* u *in* G*. Formally, vectors* p *and* d *are eigenvectors such that* $p = AA^Tp$ *and* $d = A^TAd$*.*

Even though the *passengerness* and the *driverness* are indicators of how much a user can be a good driver or a good passenger, they do not provide information about which groups of users could more easily travel together, or which geographical areas could be

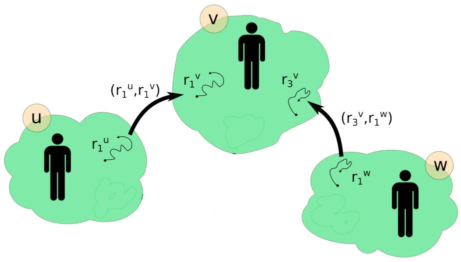


Figure 10.3: Carpooling Temporal Network.

more promising for a carpooling service. Consequently, we extract groups of users sharing common routines, which have then been analyzed to characterize each group geographically (to understand whether such groups are localized or dispersed over large areas), and with respect to their *passengerness* and *driverness*.

Definition 38 (Carpooling Community)**.** *Given a carpooling user network* $G' = \langle N,\ E' \rangle$ *we define a* carpooling community $C \subseteq N$ *as a group of users who share more routines with the users inside the community rather than with the users outside the community.*

To extract the carpooling communities and to perform the carpooling suggestions without discarding the temporal knowledge we introduce carpooling temporal networks:

Definition 39 (Carpooling Temporal Network)**.** *Given a carpooling network* $G = \langle N, E \rangle$*, a timestamp ts and a temporal duration dur, we define a* carpooling temporal network *as a direct graph* $G' = \langle N', E' \rangle$ *such that* $E' = \{e_{uv} \in E \mid isActive(e_{uv}, ts, dur)\}$ *and* $N' \subseteq N$ *is the set of all nodes comparing in* E'*. The isActive operator is defined as*

$$isActive(e_{uv}, ts, dur) \equiv (ts \leq t_1^{r_i} < ts + dur) \ \wedge \ (ts \leq t_n^{r_i} < ts + dur)$$

where $t_1^{r_i}$ *and* $t_n^{r_i}$ *are the timestamps of the first and last point of* r_i*, respectively.*

An edge e_{uv} is active if the contained routine is not finished in a certain time window. Note that a *carpooling temporal network* is a mono-dimensional direct graph if the used time window is short enough (i.e., *dur* is relatively small) and there are not two users u and v that systematically follow two different pairs of matching routines in the same time window – usually a rather extreme phenomena for reasonable values of *dur*. A *carpooling network* can be seen as a particular *carpooling temporal network* where every edge is active. Finally, we highlight that a *carpooling temporal network* is different from a *carpooling user network*, since the second considers every carpooling interaction.

Never Drive Alone Method

In the following, we describe the *Never Drive Alone (NDA)* method using the measures and concepts defined up to now. NDA performs the following steps: *(i)* extracts the systematic movements; *(ii)* builds the carpooling network; *(iii)* calculates the *passengerness* and *driverness* ranking scores; *(iv)* extracts the carpooling communities; *(v)* makes the assignments and classify the users as drivers, passengers or SOVs. The detailed procedure is described in Alg. 11 and 12. The main difference between these two versions is that the second one uses the community information, while the first one does not.

```
Algorithm 11: $NeverDriveAlone(\mathcal{M}, dur, f, m)$
  Input  : $\mathcal{M}$ - dataset of user movements, $ts$ - start of time window, $dur$ - temporal
           duration, $f$ - sorting function, $m$ - max number of free places,
  Output: $D$ - set of drivers, $P$ - set of passengers, $S$ - set of SOVs
$D \leftarrow \emptyset$; $P \leftarrow \emptyset$; $S \leftarrow \emptyset$;
$\mathcal{P} \leftarrow \emptyset$; /* set of PMDM */
for $M_u \in \mathcal{M}$ do
    $Pr_u \leftarrow extractProfile(M_u)$;
    $\mathcal{P} \leftarrow \mathcal{P} \cup Pr_u$;
end
$G \leftarrow buildCarpoolingNetwork(\mathcal{P}, contained(*))$;
$G' \leftarrow extractCarpoolingUserNetwork(G)$;
$k^{out}, k^{in} \leftarrow getDegrees(G')$; /* calculates out-degree and in-degree values */
$p, d \leftarrow HITS(G')$;/* calculates passangerness and driverness ranking scores */
$c' \leftarrow createSortingCriteria(k^{out}, p)$;/* creates the first sorting criteria */
$c'' \leftarrow createSortingCriteria(k^{in}, d)$; /* creates the second sorting criteria */
$D' \leftarrow \emptyset$; $P' \leftarrow \emptyset$; $S' \leftarrow \emptyset$;
for selected $ts$ do
    $G^{ts,ts+dur} \leftarrow extractCarpoolingTemporalNetwork(G, ts, dur)$;
    $D', P', S' \leftarrow calculateGeedyAssignment(G^{ts,ts+dur}, f, m, c', c'', D', P', S')$;
    $D, P, S \leftarrow updateAssignments(D, P, S, D', P', D')$;
    $D', P', S' \leftarrow removeFinishedInteractions(G^{ts,ts+dur}, D', P', S', ts, dur)$;
end
return $D, P, S$;
```

Given a time window defined by the parameters ts and dur discussed in the previous section, function $removeFinishedInteractions$ removes from D', P', S' the assignments that will not be active in the next time window because they end in the current one. In this way, a driver can offer a lift to more then m (max number of free places) users, because she might drop-off a passenger and later take another one, also multiple times. The returned sets classify the user according to their role in the carpooling scenario. That is, a user will be in S if and only if she is left out from every carpooling interaction in every time window. If a user can physically act either as a driver or as a passenger then she is counted as a driver because for at least a systematic trip she offered a ride and thus used her car. This happens when a user offers a ride to someone in the morning, then returns to the starting point and finally in the afternoon takes a lift to go somewhere else.

When the procedure is performed taking into account the carpooling communities (see Alg. 12), for each timestamp considered the communities are extracted and analyzed in a certain order which can depend on the size of the community. The purpose is to reduce the focus assignment problem on sets of users that are similar in the carpooling sense, that is, we give to the edges of nodes belonging to different communities a lower importance, because they are expected to offer a ride or get a lift with lower probability – typically because different communities often correspond to different geographical areas. On the contrary, users in the same communities are similar each other, thus their links are evaluated with a high importance in suggesting assignments.

```
Algorithm 12: $NeverDriveAloneCommunities(\mathcal{M}, dur, f, m)$
Input  : $\mathcal{M}$ - dataset of user movements, $ts$ - start of time window, $dur$ - temporal
         duration, $f$ - sorting function, $m$ - max number of free places,
Output: $D$ - set of drivers, $P$ - set of passengers, $S$ - set of SOVs
$D \leftarrow \emptyset$; $P \leftarrow \emptyset$; $S \leftarrow \emptyset$;
$\mathcal{P} \leftarrow \emptyset$; /* set of PMDM */
for $M_u \in \mathcal{M}$ do
    $Pr_u \leftarrow extractProfile(M_u)$;
    $\mathcal{P} \leftarrow \mathcal{P} \cup Pr_u$;
end
$G \leftarrow buildCarpoolingNetwork(\mathcal{P}, contained(*))$;
$G' \leftarrow extractCarpoolingUserNetwork(G)$;
$k^{out}, k^{in} \leftarrow getDegrees(G')$; /* calculates out-degree and in-degree values */
$p, d \leftarrow HITS(G')$;/* calculates passangerness and driverness ranking scores */
$\mathcal{C} \leftarrow extractCommunities(G')$; /* extracts the users' communities */
$c' \leftarrow createSortingCriteria(k^{out}, p)$;/* creates the first sorting criteria */
$c'' \leftarrow createSortingCriteria(k^{in}, d)$; /* creates the second sorting criteria */
$D' \leftarrow \emptyset$; $P' \leftarrow \emptyset$; $S' \leftarrow \emptyset$;
for selected $ts$ do
    $G^{ts,ts+dur} \leftarrow extractCarpoolingTemporalNetwork(G, ts, dur)$;
    for $C \in \mathcal{C}$ do
        $G_C^{ts,ts+dur} \leftarrow extractSubGraph(G^{ts,ts+dur}, C)$;
        $D', P', S' \leftarrow calculateGeedyAssignment(G_C^{ts,ts+dur}, f, m, c', c'', D', P', S')$;
    end
    $D, P, S \leftarrow updateAssignments(D, P, S, D', P', D')$;
    $D', P', S' \leftarrow removeFinishedInteractions(G^{ts,ts+dur}, D', P', S', ts, dur)$;
end
return $D, P, S$;
```

Sorting and Matching Strategies

Both Alg. 11 and 12 rely on the greedy procedure reported in Alg. 10. It is worth to underline that this procedure is based on the knowledge extracted form data. Indeed, the structure of the greedy assignment exploits the fact that the carpooling networks show a power low distribution of the nodes' degree (see the detailed study provided in the following). By using smart sorting criteria, our purpose is to lead the algorithm to consider first the least "promising" passengers (i.e. the most difficult ones to match), and then by ordering their drivers, to assign the worst passengers with their least promising drivers. This way, passengers with fewer possibilities to be matched are assigned first, while passengers which have more opportunities are assigned to the remaining drivers. We can instantiate this reasoning both using the in/out degrees and using the passengerness/driverness ranking criteria.

We consider the following criteria, in order of complexity:

- *(r) random criteria* ($c' = \{random$ order$\}, c'' = \{random$ order$\}$): users are sorted randomly both if they are drivers or passengers;
- *(g_1) degree criteria* ($c' = \{k^{out}$ ascending order$\}, c'' = \{k^{in}$ ascending order$\}$): users are sorted according to the carpooling user network *out-degree* k^{out} and *in-degree* k^{in}, i.e. by increasing k^{out} and than, their neighbors are ordered by increasing k^{in};

- *(g_2) degree - ranking scores criteria* ($c' = \{(k^{out}, p)$ order$\}, c'' = \{(k^{in}, d)$ order$\}$): users are sorted according to *passengerness* p and *driverness* d in addition to k^{out} and k^{in}, that is, the nodes are sorted in a lexicographical order by increasing (k^{out}, p) and then, their neighbors are sorted in a lexicographical order by increasing (k^{in}, d).

The methodology described could be applied also switching passengers with drivers, i.e. by enumerating drivers first, and then matching each of them with her possible passengers. Yet, preliminary experiments proved that this order is largely less successful.

Another information that can be exploited to guide NDA is the community membership. Therefore, we consider two further variants of the method: a basic one, which is agnostic of the communities; and a community-driven one, where the matches between intra-community individuals have priority over all the others:

- *(w) plain version*, Alg. 11, considering every edge with the same importance;
- *(c) prioritized version*, Alg. 12, that suggests an assignment to the users inside the same community and then, if that fails, among users of different communities.

Finally, we adopted two strategies w.r.t. the temporal dimension. The routines linking any pair of profiles make the carpooling network a summary of a typical day. We can decompose this day in a series of time slots with a predefined duration (*dur*), obtaining a series of carpooling temporal networks. The way the sequence of time slots is produced is a parameter of the method. We consider two variants of time slots:

- *(discrete)*: they start at discrete time instants, e.g. one every 5 minutes from midnight. This produces a sliding window of length *dur* that moves of step 5 minutes;
- *(continuous)*: they start in correspondence of the last carpooling interaction, i.e. the time of the last matched routines becomes the next starting time.

In the following we evaluate experimentally each combination of the three parameters discussed, i.e. sorting criterion, usage of communities, choice of time slots.

10.1.3 Case Study

In this section we illustrate an instantiation of the overall approach proposed on a real case study, and we show the results obtained. The section is divided into two main parts: construction of the carpooling networks, and selection of carpooling recommendations.

Dataset

As a proxy of human mobility, we used the real GPS traces dataset *Octo* described in Section 6.1. In particular, we focus on the area of Pisa and Florence provinces. Moreover, since it is commonly observed that during Saturday and Sunday most people leave their working mobility routines and adopt other more erratic behaviors, we consider only working days, i.e. from Monday to Friday. Finally, we remove potential noise trajectories by not considering too short trips (less than 1km). In conclusion, we employ our analytical method on a total of $\sim$50,000 users and $\sim$1.400.000 trajectories. After the mobility profile exaction phase, in total we obtain $R_C = \sim 17,200$ routines. On average we have 2.14 routines per user, i.e. home to work and work to home. Note that routines are not available for all

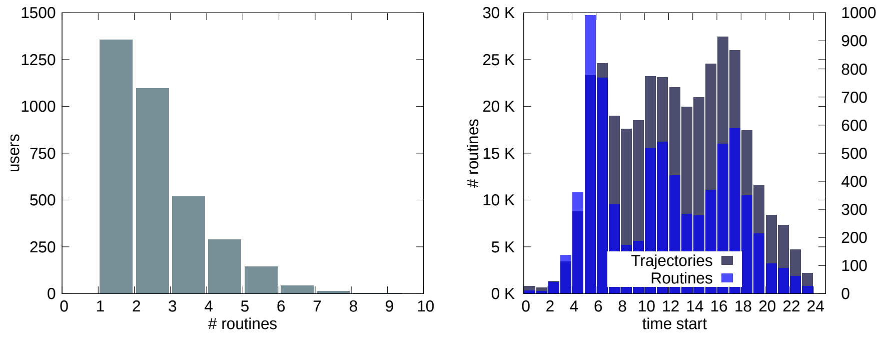


Figure 10.4: Distribution of routines per user (left), trajectories and routines start time (right).

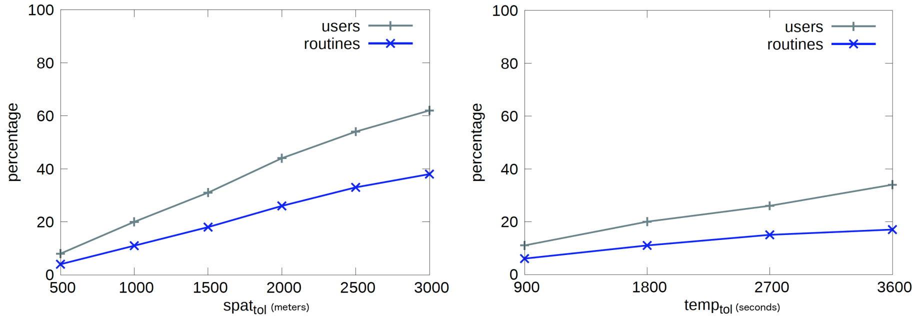


Figure 10.5: Network construction, *contained* parameters test: (left) $spat_{tol}$, (right) $temp_{tol}$.

the users analyzed because for this study we required that a routine must be supported by at least 8 trajectories in order to consider only very repetitive movements such that the results are statistically meaningful. Fig.10.4 *(left)* shows the number of routines per users in Pisa province, with almost every user having one or two routines, which most likely correspond to commuting trips between home and work. Fig. 10.4 *(right)* reports the temporal distribution of the trajectories and routines. We can see that the profiles follow the timing of typical working days, highlighting the three peeks during early morning (5–6), lunchtime (11–12), and late afternoon (17–18).

Carpooling Network

In this section we instantiate the network construction step of the methodology proposed, and analyze the characteristics of the resulting carpooling networks. First, we focus on the information that can be inferred from the network, trying to obtain preliminary estimations of the potential reduction of traffic. Then, we study the topological properties of the network, computing ranking measures and extracting communities.

Network Construction. The carpooling network is derived by the application of the function *contained*. Therefore, the resulting network directly depends on the value used for its parameters $spat_{tol}$ and $temp_{tol}$. In order to find good values for these parameters and to obtain a sound network made of reliable carpooling interactions, we performed a network construction test on a sample of $1,000$ mobility profiles. Fig. 10.5 shows how the containment is affected, in percentage, in terms of routines and mobility profiles that have

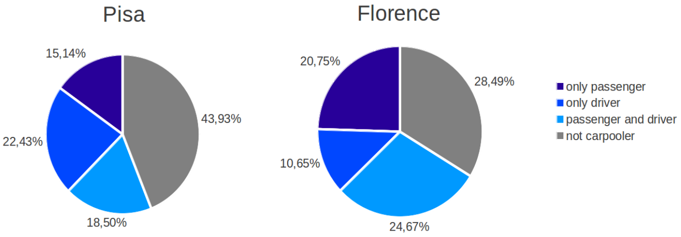


Figure 10.6: Carpoolers classification pie chart for Pisa and Florence.

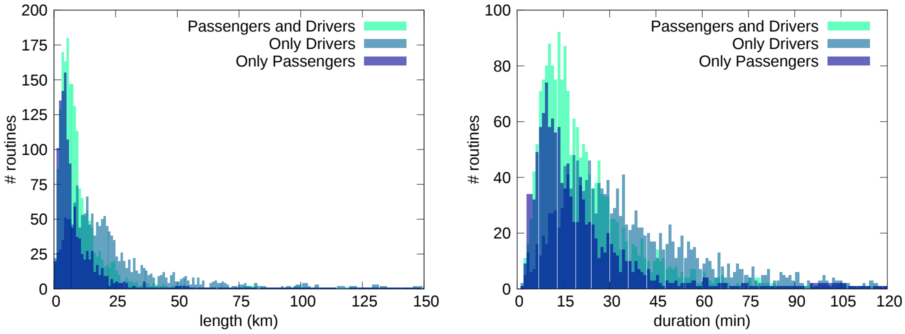


Figure 10.7: Routines distribution: length (left), duration (center), time start (right).

at least one match. The default values of $spat_{tol}$ and $temp_{tol}$ are, respectively, 1 km and 30 minutes. It is worth to notice that by allowing a walking distance ($spat_{tol}$) of 3 km and a wasting time ($temp_{tol}$) of 30 minutes, about 60% of the profiled users have at least one match, which decreases to 10% if the walking distance becomes 500 meters. Similarly, by allowing a walking distance of 1 km and a wasting time of 60 minutes, 30% of the profiled users have at least one match, which decreases to 10% if the wasting time becomes 15 minutes. This suggests that an increase in the walking distance has a larger impact than an increase in the wasting time, in terms of number of carpooling matches. Based on these observations, we built the carpooling networks for Pisa and Florence using a maximum walking distance of 1 km and a maximum wasting time of 30 minutes.

Network Analysis. By observing the *users* appearing in the carpooling networks (among those which have a mobility profile), we can distinguish those that can join others as passengers or drivers, and those that cannot. In particular, we can classify them into four categories, based on their in- and out-degree in the network:

- *only passengers*: they can only get rides, i.e., k^{in}=0 and k^{out}>0.
- *only drivers*: they can only offer rides, i.e., k^{out}=0 and k^{in}>0;
- *passengers and drivers*: they can act as passengers and drivers: k^{out}>0 and k^{in}>0;
- *no carpoolers*: they have systematic movements but do not share any routines with other users: $k^{out} = 0$ and $k^{in} = 0$.

With respect to the definitions introduced in the previous section, users which are only passengers belong to PP, those which are only drivers belong to PD, and the users which are passengers and drivers belong to both PP and PD. Fig. 10.6 depicts the pie chart with the

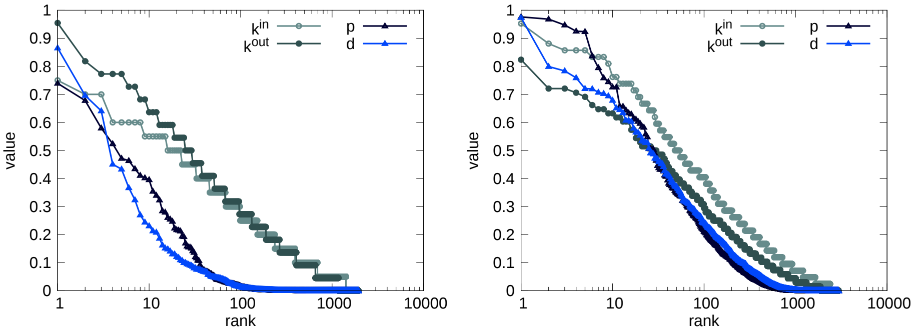


Figure 10.8: Degree and ranking scores distribution: (left) Pisa, (right) Florence

percentages of different types of users in the carpooling user networks of Pisa and Florence. We can observe how the carpooling potentiality is different in the two cities, with Florence showing larger percentages of carpoolers, especially of the *driver and passenger* type.

In Pisa we obtain around 7,400 mobility routines, each representing at least 8 single trips of the user (indeed, *minsize*=8 in these experiments), for a total of around 59,200 systematic trips. Also, we discover that around 1,720 of the routines are actually contained in at least one other routine, i.e. the user could carpool with another driver, which means a potential reduction of systematic mobility of about 23%. We finally analyze the spatio-temporal features of the routines extracted. Fig. 10.7 (left) shows the length distribution of routines for the categories we described above on the Pisa dataset. We notice that users who are *only passengers* mainly have a routine length between 0 and 10 km, while the *only drivers* have longer routines, between 5 and 25 km. This fact, confirms by the distribution of trip durations in Fig. 10.7 (right), meets the intuition that users traveling for longer distances can more easily offer lifts to others, while short-distance travelers can more easily be taken as passengers.

The following analysis is focused on some *topological features* of the carpooling user networks. In particular, the degree (in-degree k^{in} and out-degree k^{out}) of nodes and their ranking scores (driverness d and the passengerness p). The ranking scores are calculated running the *HITS* algorithm on the carpooling user networks[1]. Fig. 10.8 shows both the degrees and the ranking scores distribution for Pisa and Florence, with values rescaled to the $[0, 1]$ interval in order to make the two plots comparable. Both distributions are long-tailed, meaning that there few users have high values and many users have low values. As highlighted in the previous section, some users are *only passenger* or *only driver*, and therefore their corresponding nodes in the network have k^{in}=0 or k^{out}=0. We can notice that in Pisa, many users also have a zero driverness d, and, the same happens for passengerness p. This emphasizes the significant difference that exists between the degree and the ranking scores, at least in the carpooling user network of Pisa. The conclusion is that, despite the obvious correlation between k^{out} and p, and between k^{in} and d, they can behave in a significant different way, and users that can be drivers for many passengers might possibly be not *good drivers*, and vice-versa. On the other hand, the carpooling user network of Florence is denser, and the correlation between degree and ranking scores is higher.

[1] For this task we adopted the Python implementation of HITS provided by the NetworkX library (http:networkx.github.io), with a tolerance threshold of 1.0e-8.

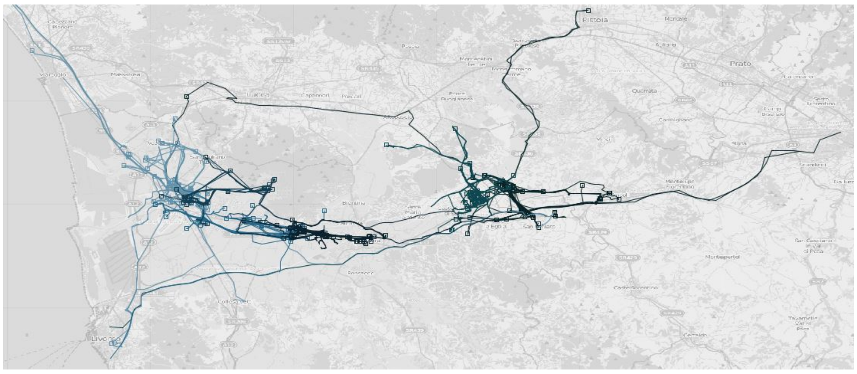


Figure 10.9: Geographical view of some carpooling communities in Pisa province.

The main differences between the two provinces are in the p and d ranking scores. In Pisa the driverness d rapidly falls down getting close to zero within the first one hundred users, while in Florence it decreases much more gradually. A similar consideration can be done by looking at p. Moreover, in Pisa there are few drivers with a high d, suggesting that only a few of them can serve good passengers, while Florence has more good drivers. Most of the nodes in the networks considered here have very low degrees, between 3 and 8. This is probably due to the strict parameters that we adopted in building the carpooling networks to have reliable interactions. The effect is that the carpooling users networks are very sparse, which turns to be an advantage for the task of suggesting assignments since each user has only a small number choices to consider. Finally, both datasets show a standard deviation of the all features (k^{in}, k^{out}, d and p) larger than their mean, suggesting that our users are rather heterogeneous. Also, passengerness and driverness appear to be poorly correlated, resulting in a Kendall's Tau coefficient 0.134.

Communities. The HITS algorithm returns an indicator of how much a user can be a good driver or a good passenger. However, these ranking scores do not help in grouping similar users, that is, users that with a high probability would like to share their travels. For this purpose, we use carpooling communities, i.e., groups of users who share more routines with other users inside the group than with users outside the group. Various state-of-art community discovery algorithms were tested for this purpose, including Infohiermap [254], Louvain [40] and Demon [84]. Finally, the Demon algorithm was selected, due to its better performances both in terms of runtimes and quality of the result. Fig. 10.9 shows a sample of carpooling communities in Pisa province. It is interesting to notice that the carpooling communities are geographically well localized. Every community acts on a specified area that contains the systematic movements of its users. This means, for instance, that a user who is active in the northern area of Pisa can generally disregard the mobility of any user that is moving in different areas.

The topology of the communities emerging from the network results to be very similar to the topology of the original carpooling user network. That is, every community, from a topological point of view, behaves as the overall network. The average size of the communities is $30 - 40$ nodes and the average degree inside a community is around 4 with a low standard deviation (1.32 on average). Observing the distribution of the driverness and passengerness scores within each community, shown in Fig. 10.10 for Pisa and Florence province, we discover that the carpooling communities can be classified into two

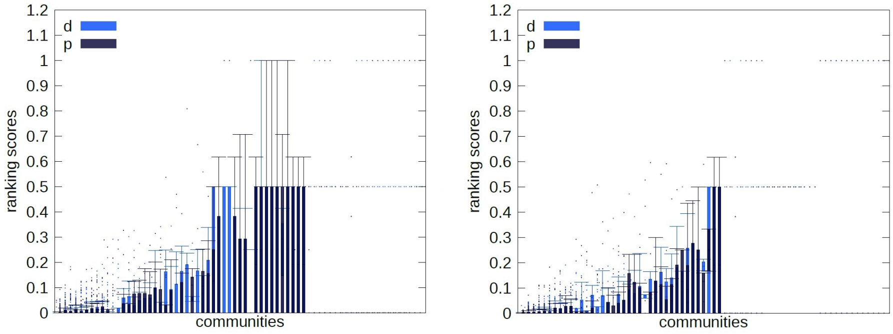
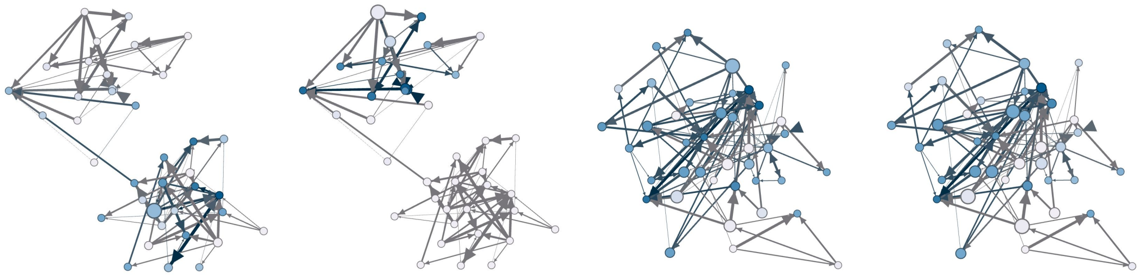


Figure 10.10: Carpooling ranking scores box-plot for Pisa (left) and Florence (right).

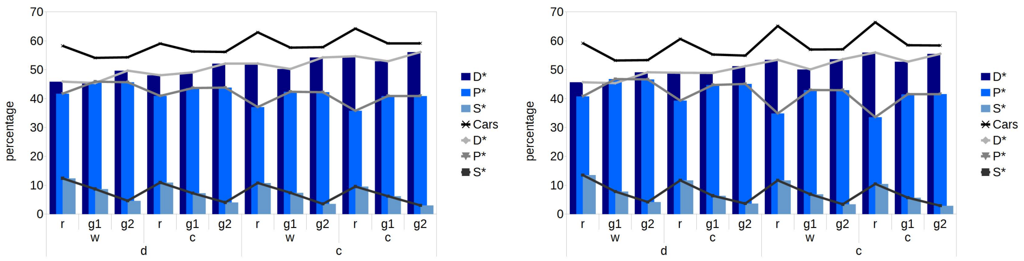


(a) Not Autonomous - Global (b) Not Autonomous - Local (c) Autonomous - Global (d) Autonomous - Local

Figure 10.11: Pisa, examples of a *not autonomous* and an *autonomous* community, showing *global* and *local* ranking scores. Size of nodes represents driverness. Darkness represents passengerness.

categories. Indeed, we can see from the box-plots that the distributions on the different communities have a high variability, showing a group of communities having consistently very low values, while the others are made of nodes with (on average) high ranking scores.

Then, we evaluated how much the ranking scores d and p of a node change if they computed considering only the community it belongs to, i.e. running the HITS algorithm locally to the sub-network formed by each community. We call the new scores *local driverness* and *local passengerness*, to distinguish them from the *global* values. By analyzing the Kendall's tau correlation between the global and local ranking scores for each community we found that, in the Pisa dataset, there are about 30 communities with a correlation close to one, while the remaining circa 20 communities have correlations lower than 0.4. That means that the first group of communities in the PDE are basically *autonomous*, since they are very weakly influenced by the nodes outside the community, and therefore could rely on finding possible assignments without considering inter-community links. On the other hand, the other communities are *not-autonomous*, since they can be influenced by inter-community links and their users could find potential best matches with users belonging to a different community. Fig. 10.11 shows real examples of a *not-autonomous* (left) and a *autonomous* community (right), depicting both the global ranking scores (left column) and the local ones (right column). The size of nodes represents the driverness score, while its darkness represents passengerness. We remark that virtually nothing changes for the *autonomous* community, whereas completely different scores emerge for the *non-autonomous* community, confirming the observations discussed above.



Pisa

Florence

Figure 10.12: Assignment results for all strategies and criteria adopted.

Carpooling Suggestions Performances

In the following we describe the results obtained by performing the Never Drive Alone procedure on Pisa and Florence datasets. The assignment performance evaluation is done by measuring the number of resulting SOVs, the number of systematic cars travelling, as well as evaluating the impact of NDA in economic and environmental terms.

Experiments Setup. The NDA procedure has been tested considering all the variants previously discussed. Moreover, the vehicle capacity of each user has been fixed to $m = 4$, i.e. each vehicle can host four passengers in addition to the driver, which fits quite closely the local standards of the area under study. Also, the time slot duration for the creation of temporal networks was fixed to $dur = 1$ $hour$, meaning that trips longer than one hour might be prevented from being matched to others even if the *contain* relation holds – an extremely unlikely event in our dataset, since 1-hour routines are very rare.

Results. Fig. 10.12 shows the percentage of passengers P^*, drivers with passengers on-board D^* and SOVs S^* obtained over Pisa and Florence by applying each combination of the criteria adopted (abbreviations (r), (g_1), etc. are those provided in Section 10.1.2). In addition, it shows the corresponding number of (systematic) cars on the road (see the dark line on the top of both pictures). As first evaluation, we see that there are always more than one third of users that become passengers, in most cases around half of the users become drivers with passengers, and only a small percentage remains a single-occupant vehicle.

We notice also that, while there are significant differences of performances among the algorithm variants, the simplest (random) variant already reaches very good results, with a SOV around 12%. Such result suggests that the networks considered constrain significantly the assignment phase, leaving few alternative opportunities to explore, although smarter assignment methods are able to improve the results. More tolerant settings in the construction of the carpooling network (such as admitting matched with longer distances to walk to take a lift) are expected to yield networks with more alternatives to explore, and therefore make the improvement margins over the random solution much larger.

The plots show that the knowledge extracted from the mobility data and refined with network analysis progressively leads to improvements regarding the minimization of the number of SOVs. Indeed, we observe that the sorting criterion *(g_2)*, gets better results than the sorting criterion *(g_1)*, which in turns outperforms *(r)*. Moreover, Fig. 10.13 also depicts how the strategy considering the community information *(c)* slightly reduces the number of SOVs with respect to the strategy that considers the whole network *(w)*. This suggests that the carpooling service might be organized in a local way, i.e. it might be convenient to

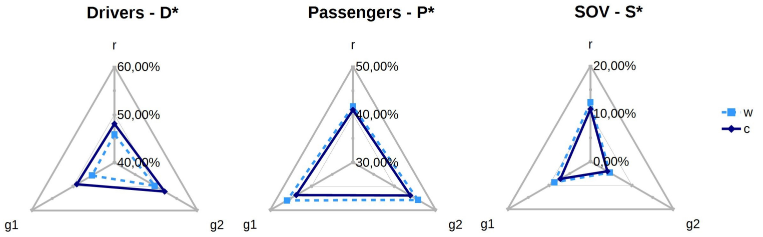


Figure 10.13: Assignment results for the strategies *(w)* and *(c)*, and the three sorting criteria.

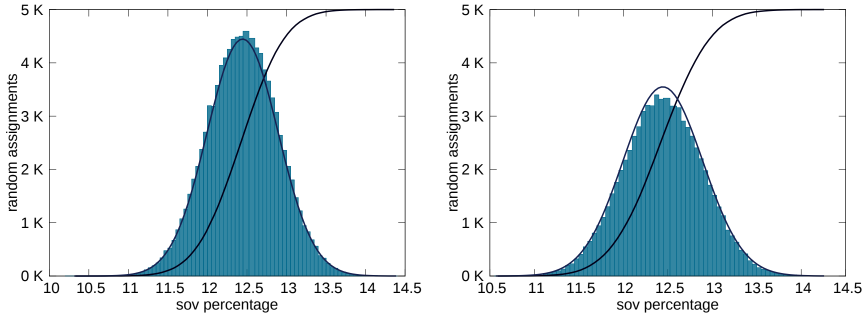


Figure 10.14: SOVs percentage distribution (PDF and CDF) of random assignment tests ran 100,000 times for discrete time strategies. Communities not considered *(left)* and considered *(right)*.

focus the proactive suggestions mainly among users within the same community, basically disregarding the others. Also the temporal information contributes with useful suggestions: considering dynamically each change in the carpooling interactions *(d)* to compute the assignments procures a little advantage with respect to the one obtained using fixed time slots *(s)*. Yet, the calculus with *(d)* is computationally more expensive, especially in periods where carpooling interactions are frequent (morning, midday, evening).

So far, our considerations were focused on minimizing the number of SOVs. Anyway, if we want primarily to minimize the number of systematic cars traveling, and only secondarily the number of SOVs, we discover that the best approach still uses the *(g_2)* criteria, yet this time considering the whole network *(w)* and static (discrete) time slots. Finally, Fig. 10.12 also shows that, although Florence has more good drivers and passengers than Pisa, the two carpooling networks yield comparable results in terms of suggestions.

NDA vs. Random Assignment Approach. In order to better verify that the provided solution is consistently better than those found by a random exploration of choices, we report in Fig. 10.14 the results obtained by running 100,000 times NDA with random sorting criteria *(r)* on the Pisa carpooling network, considering the whole network without assignment priorities (left *(w)*) and prioritizing the assignments between nodes in the same community (right *(c)*). What we obtain in both cases is a normal distribution. Regarding *(w)* the mean value of SOVs, obtained nearly five thousand times, is 12.44 and the standard deviation is 1.48. On the other hand, considering *(c)*, the mean value is 12.28, a bit lower than the previous, but obtained no more than three thousand times and a half, and with a larger standard deviation of 1.97. The solution provided by NDA considering both carpooling ranking measures and community knowledge provides a SOVs percentage slightly smaller than 4.63%, which is largely better than anyone found by the 100,000 random runs. Indeed, according to the distributions shown in the figure, the expected probability of finding a SOVs percentage lower than that is around $6.56 \cdot 10^{-8}$, therefore very close to zero.

Comparison with Existing Approaches. As described in the related works, most of the literature on carpooling is focused either on the simulation of very specific aspects, such as the impact of high occupancy vehicle lanes on traffic, or on the realization of a real-time service. On the opposite, our work aims to provide a solution for carpooling matching and study its impact in a real context. The main works that tackle problems close to ours, are [135] and [76], which we considered for a comparison of performances. Both works are based on data sources significantly different from those adopted in our paper: [135] is tailored around (geo-localized) Twitter data, and exploits the topics of the text messages posted and the social network of users; [76], instead, is based on a mix of mobile phone data (CDR traces) and social media (geo-localized Twitter posts and Foursquare check-ins). That makes a direct (and fair) comparison over a common benchmark very difficult.

Another important difference between our approach and the two competitors considered, is that the latter aim to maximize the number of users involved in the carpooling, yet not considering explicitly the overall coherence of the carpooling assignment, i.e. a passenger for a home-to-work trip needs to be passenger also for the return trip. In the following summary of results, we call this incomplete form of assignment *partial passengers*, in contrast to the complete one, called *total passengers*. As described in Section 10.1.2, our approach is focused on the more realistic scenario of *total passengers*, which is ensured by requiring that the status of the user (passenger or driver) is kept for the whole day.

Below we provide an indirect comparison of the three methods, summarizing the performance results obtained by each of them over its own datasets:

- **CAR-O [135]**: 71.95% of users in the Rome dataset and 74.82% of users in San Francisco become *partial passengers*. Impact on single trips saved not provided.
- **EN-ROUTE [76]**: 65% of users in Madrid and 68% in New York become *partial passengers*. Impact on single trips saved not provided.
- **NDA:** 43.83% of users in Pisa and 45.10% in Florence become *total passengers*. Impact on single trips is 77.52% in Pisa, and 77.03% in Florence.

These results suggest that the matching strategies provided by our solution can reach an impact over car traffic that is apparently similar to those obtained by other approaches in similar contexts, yet providing a more realistic application scenario.

Evaluating the Economic and Environmental Impact of Carpooling

In order to evaluate the practical importance of the carpooling matching discussed in the previous section, we consider here the best configuration setting for the system and study its results from several viewpoints. The first one is simply the impact of the carpooling in terms of reduction of cars on road. Tab. 10.1 summarizes the number of routines with details on the number of routines that might potentially be served by other drivers (*# can ride*), those that might give a lift to other passengers (*# can drive*) and their union (*# linked*). Finally, the number of matches that were actually found by the algorithm, also in terms of percentage over the maximum theoretical outcome, i.e., the number of potential passengers. NDA is able to assign most part of the potential passengers in both cities (around 77% of them), also corresponding to a relevant percentage of total routines (cars on road) saved, namely 18% in Pisa and 26% in Florence.

City	# routines	# linked	# can ride	# can drive	# saved trips
Pisa	$7,383$	$3,049$	$1,717$	$1,995$	$1,331$ (77.52%)
Florence	$9,801$	$5,712$	$3,305$	$4,140$	$2,546$ (77.03%)

Table 10.1: Number of routines extracted in the two cities, the routines that are linked to others in the carpooling network, those that might be served by others, those that might serve at least another one, and number of matches found by NDA (in percentage w.r.t. potential passengers).

City	km	min	$fuel(l)$	€	$CO_2(kg)$
Pisa	$10,868.36$	$24,174.58$	646.67	$1,001.49$	$1,445.49$
Florence	$16,748.99$	$43,300.28$	996.56	$1,543.37$	$2,227.62$

Table 10.2: Estimates of total potential savings in a normal day obtained by using the proactive carpooling proposed in this work. Savings are expressed in terms of total kilometers driven, time spent driving, fuel consumed, its cost and CO_2 emissions.

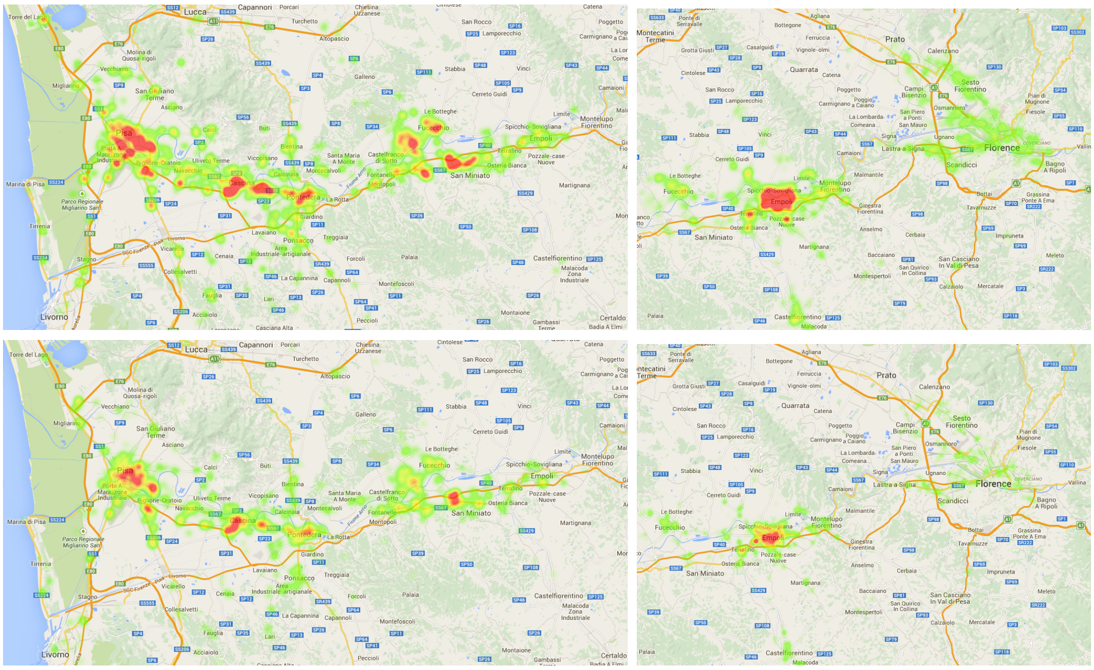
Tab. 10.2 reports the economic and environmental impact of the carpooling on the traffic reductions. Estimates of such impact are computed considering the most common car sold in the period of data collection, an average gasoline consumption of $0.0595l/km$, a gasoline cost in the observation period of 1.54869€ per liter, and a CO_2 emission of 133 g per km[2]. Considering that the estimates reported are relative to a single city and a single (typical) day, the reduction values are very significant, especially towards the environment.

Finally, we show in Fig. 10.15 the spatial distribution of pick-up (top row) and drop-off (bottom row) points of the solution found by NDA on Pisa (left) and Florence (right). We can see that in the case of Pisa, carpooling mainly (yet not exclusively) involves several smaller cities distributed along an important road towards East, connecting Pisa with the other major cities of the region. For Florence it is interesting to notice that a major hotspot, even larger than Florence itself, is located in a nearby city, Empoli, characterized by a huge flow of commuters towards Florence and the surrounding industrial areas. In general, carpooling is much more concentrated around a few dense areas than what happens for Pisa. In both cases, the drop-off points appear to be more concentrated around the main attractors, while pick-up points are slightly more dispersed.

10.1.4 Conclusion

We have proposed NDA, a novel and proactive approach that through Personal Data Mining, by exploiting the PDS in a PDE, can be used for analyzing the potentiality of a carpooling service and for suggesting an assignment among systematic car drivers in order to have them not to drive alone. We underline that such service can be realized only for the users of the PDE which are available to share their routines. By analyzing the collective knowledge extracted many useful observations resulted from our study. We discovered that indicators derived from the carpooling networks, like the number of only drivers, only passengers, passengers, and drivers, can be used to characterize different areas and cities in terms of applicability of carpooling. Also, a measure of empirical upper bound of the potential reduction of cars on the road can be inferred, whose average in the area of our experimentation is around 23%. The carpooling networks tend to be

[2] http://www.patentati.it/blog/articoli-auto/classifica-auto-2011.html,
http://dgerm.sviluppoeconomico.gov.it/dgerm/prezzimedi.asp?anno=2011,
http://www.ilsole24ore.com/speciali/emissioni



Pisa Florence

Figure 10.15: Spatial distribution of pick-up and drop-off points of NDA solution. First row: pick-up points; second row: drop-off points.

very sparse, and are characterized by long tailed distributions both for the in- out-degree and for the driverness and passengerness indexes. We showed how ranking measures and communities extracted from mobility networks can be used to characterize different aspects of human mobility. By exploiting them, we proposed our approach for boosting carpooling using network analysis. Furthermore, we have found that carpooling communities can be classified into autonomous communities, that, being independent from the rest of the car drivers, are made by many good carpoolers offering and taking lifts to many users, and non-autonomous communities, that being influenced by extra community car drivers, cannot be managed on their own. Therefore, if a new carpooling service is to be realized, a good start point would be autonomous communities. Finally, we saw how the potential carpooling network can be used to suggest assignments among systematic car drivers and how ranking measures considered on communities lead to valuable reductions of the cars employed in systematic mobility. The heuristics for carpooling assignments we developed greatly benefits from the knowledge provided by the driverness and passengerness scores, as well as the fragmentation into communities. Performances show a percentage of SOVs as low as 4.63%, which is less than half of what any random assignment can reach in practice. As overall result, among the users part of the system, i.e., the systematic users with at least a carpooling match, about 77% of the trips could be saved on both datasets, and the estimates of saved kms, time, fuel, money and CO_2 emissions are significant.

10.2 Enjoyable Carpooling from Crowdsourced Data

One of the many obstacles that prevent carpooling from being adopted as an everyday means of transport is a sort of "psychological barrier" that makes it less attractive. However, thanks to the advent of online social networks, in the last few years there have been some social aspects that people intentionally decide to share with the outside world, including strangers. In fact, interests, pictures, locations, are the basis of the success of services such as Facebook, Twitter, and Foursquare. The availability of such information allows external services and people to use data for third party applications. As a result, such social aspects can be now *measured* and *exploited* to overcome this invisible psychological barrier. We model data mobility and social aspects according to the Personal Data Analytics approach. Our goal is to measure how many users would *enjoy* sharing a trip with other people, and exploit the extracted insights to drive a carpooling optimization model for more enjoyable trips.

Inspired by the literature on carpooling [81, 199, 276, 309], and by the recent work on data-driven analysis of urban networks [229] and data-driven optimization of urban transit networks [33, 218], we present a formulation of the carpooling problem taking into account the above factors. The proposed methodology consists in a PDE that, similarly to the work presented in Section 10.1, thanks to the Personal Data Models extracted and provided by each user is able not only to automatically derive mobility matches, but also to consider social matches to be used as recommendations for the carpooling system.

In the following we present *GRAAL*, a methodology for *GReen And sociAL* carpooling [135, 127]. GRAAL optimizes a carpooling system, at the city level, not only by minimizing the number of cars needed, but also by maximizing the *enjoyability* of people traveling together. We introduce a measure of enjoyability based on people's interests, social links, and tendency to connect to people with similar or dissimilar interests. Specifically, our enjoyability measure takes into account two factors: *(i)* what we call *like-mindness*, i.e., a topic similarity between any two users; and *(ii)* what we define as *homophily*, i.e., the tendency of a person to group with similar ones. Previous attempts to use social context in carpooling include putting together in a car people who are friends [76]. However, by looking at only the direct (or even the two-hop) friends, we may loose good chances for optimization, as the set of potential drivers (or co-passengers) is usually much larger than the typical number of friend pairs in a social network. Finally, in GRAAL, we introduce a multiobjective optimization based on a weighted linear combination of two components: *(i)* number of cars (which is minimized) and *(ii)* total enjoyability of the users in the system (which is maximized). We present the results of applying GRAAL on real world crowd-sourced data from Twitter, geo-located in the cities of Rome and San Francisco. In order to enhance the dualism between individual and collective point of view, results are presented from both the city-wide perspective, and from the user perspective.

10.2.1 Problem Definition

The objective of the carpooling problem is the minimization of the number of cars, together with the maximization of the *enjoyability* experienced by the users. Our goal is to follow the main advantage of the carpooling idea, i.e. lowering the number of circulating cars, while ensuring that the passengers will enjoy traveling together. This may serve as an additional, non-monetary, incentive to motivate people to share a car.

Preliminaries

Enjoyability. We define a measure of enjoyability that takes into account not only whether two users share the same interests, but also whether they tend to connect to people with similar or dissimilar interests.

Let U be a set of users. Every user $i \in U$ may consider other users in U as friends, or interesting in general, and we denote such set of users as F_i. Therefore, the set of friends F_i is available as component of the PDS for each user $i \in U$. Each user i generates, or is interested in, a set of articles or documents D_i. Given i and D_i, we can build for each user a vector of topics $\vec{t_i}$,where each topic is weighted by its relative importance, i.e., frequency, within the documents. Once again, for each user, the topics vector $\vec{t_i}$ is part of her PDS.

We define a measure, which we call *like-mindness*, of how much two users are interested in the same topics, as follows.

Definition 40 (Like-Mindness)**.** *Given users i, j we call their* like-mindness *the number:*

$$lm_{ij} = 2\frac{\vec{t_i} \cdot \vec{t_j}}{\|\vec{t_i}\|\|\vec{t_j}\|} - 1$$

We say i and j are like-minded, i.e. they share a set of interests, if $lm_{ij}{\approx}1$, not-like-minded if $lm_{ij}{\approx}{-1}$. We want to take into account two different categories of people: those who are more prone to be in contact with other people with similar interests (*homopilous* people), and those who tend to connect with people with dissimilar interest (*heterophilous* people). We evaluate user's tendency to connect with people with whom she has a high or low like-mindness. In social networks, the concept of homophily is well known [205].

Definition 41 (Homophily)**.** *Given a user i we compute his/her* homophily *as the median of the like-mindness between i and other users in F_i:*

$$h_i = \operatorname*{median}_{j \in F_i} \ lm_{ij}$$

If $h_i \approx 1$, we say that i tends to be homophilous, while if $h_i \approx -1$ we say that i tends to be heterophilous. Our objective is to relate the like-mindness of a pair of users with the homophily/heterophily of the single user. Thus, we define the enjoyability as:

Definition 42 (Enjoyability)**.** *Given two users i, j, their like-mindness lm_{ij} and their homophily values h_i, h_j, we define the* enjoyability *of them being together as:*

$$e_{ij} = \frac{lm_{ij}h_i + lm_{ij}h_j}{2}$$

We refer to the set the enjoyabilities computed between each pair of users as E. Note that $e_{ij} \approx 1$ if either: *(i)* both i and j are homophilous and like-minded; or *(ii)* i and j are heterophilous and not like-minded. In the other cases, $e_{ij} \approx -1$. The added value of social diversity has been studied in social science, and finds applications also in the scientific community (sometimes referring to "serendipity" when something unexpected brings added value). Socio-cultural diversity is often considered fundamental [231] to make people enjoying a discussion.

The objective function we present in Section 10.2.1 is a linear combination of two components: number of cars and total enjoyability. As we minimize the number of cars, we take into account the *unenjoyability* of the system, rather than the enjoyability, to minimize this as well. The unenjoyability is computed as $\bar{e_{ij}} = 1 - \frac{1}{2}\left(e_{ij} + 1\right).$

Mobility Demand. In this approach we adopt a simplified version of the Personal Mobility Data Model capturing the users' mobility demand. We define a *location* l as any geo-referenced format. We divide the areas of interest in a grid of cells (of either 500m or 70m of width). Each user i can have a different *locations* l over time. We call *time-stamped location* a pair $tsl = (l, ts)$ where l is a location and ts is an associated relative time-stamp. Two time-stamped locations are defined to be close in space and time as follows:

Definition 43 (Close Time-Stamped Locations). *Given two time-stamped locations* $tsl_1 = (l_1, ts_1)$ *and* $tsl_2 = (l_2, ts_2)$*, we say that* tsl_1 is close to tsl_2 *(*$tsl_1 \simeq_{\delta,\tau} tsl_2$*) iff*

$$\text{space-dist}(l_1, l_2) \leq \delta \ \textit{and} \ \text{time-dist}(ts_1, ts_2) \leq \tau$$

where space-dist$(\cdot,\cdot)$ *and* time-dist$(\cdot,\cdot)$ *are two functions of spatial and temporal distance.*

The choice of the specific functions is left for the specific application. Examples for distance calculation include the Euclidean, Spherical, or Manhattan, and for time function one can consider simply the time difference. In this work, we use the spherical distance between two rectangular cells of the grid defined above, and the time difference. If ts_1 or ts_2 are undefined, then the $\simeq$ operator considers only space-dist$(l_1, l_2) \leq \delta$.

As usual, we name *trajectory* a sequence $tr = \{tsl_1, \ldots, tsl_n\}$ of time-stamped locations, and we refer to *mobility demand* $H_i = \{tr\}$ as the set of trajectories of user i. The mobility demand H_i together with the topics vector $\vec{t_i}$ form the Personal Data Model P_i adopted. We indicate with $\mathcal{H}_U = \{T_i\}$ the mobility demand of all the users.

Likewise in Section 10.1, we define a "carpooling match" between two trajectories. In this simplified mobility scenario, we chose to force a matching of the two initial time-stamped locations of the two trajectories, and allow for a match of the final time-stamped location of the trajectory of the candidate passenger with any of the locations of the trajectory of the candidate driver, including (where possible) the final time-stamped one. In carpooling terms, this means that the driver-passenger pair should depart from their initial locations, but the driver is allowed to drop the passenger on any of the locations along the associated trajectory which are close to. More formally, we define the following condition which is slightly different from the relation of Section 10.1.2.

Definition 44 (Trajectory Containment). *Given two trajectories* $tr' = \{tsl'_1, \ldots, tsl'_{\bar{n}}\}$ *and* $tr'' = \{tsl''_1, \ldots, tsl''_{\bar{m}}\}$*, we say that* tr' contains tr'' *(*$tr' \sqsubseteq_{\delta,\tau} tr''$*) iff*

$$tsl'_1 \simeq_{\delta,\tau} tsl''_1 \ \textit{and} \ \exists n, 1 < n \leq \bar{n} \ \textit{s.t.} \ tsl'_n \simeq_{\delta,\tau} tsl''_{\bar{m}}$$

Yet in line with Section 10.1, we fix the maximum walking distance from the passenger's departure/arrival locations to pick-up/drop-off points (set by the driver) as δ and the maximum time difference in departure and arrival times as τ. Given the above definition, two users i and j having trajectories tr_i and tr_j in their mobility demand, respectively, generate a recommendation for carpooling if tr_i is contained in tr_j or viceversa. More formally, we define the *recommendation* as follows.

Definition 45 (Recommendations). *Given a set of users* U*, we define* R_U *as the set of* recommendations *with respect to the users in* U*.* $R_U = \{r_{ij}\}$ *where* $i, j \in U$ *are users and* $r_{ij} = (i, j, tr_i, tr_j)$ *denoting that* passenger j *is recommended to* driver i *because*

$$\exists \ tr_j \in H_j \ \ \textit{and} \ tr_i \in H_i \ \textit{s.t.} \ tr_i \sqsubseteq_{\delta,\tau} tr_j$$

where H_i, H_j *are the mobility demands of* i *and* j*,* j *is the* passenger *and* i *is the* driver.

By Def. 44, a passenger has to walk no more than δ, and wait no more than τ. We can group all the recommendations in a set R_U, containing all the possible recommendations between any pair of users in U. We call D the set of *possible drivers* and P the set of *possible passengers*. For each recommendation we define the variable m_{ij} that is computed as the sum of the walking distances for pick-up and drop-off point and then normalized by the maximum. This is referred to as *normalized distance between trajectories*. Note that m_{ij} exists within the interval $[0, 1]$ only if a recommendation between i and j exists.

The objective of the optimization method is to find a set A_{R_U} of *assignments* containing a subset of recommendations of R_U, such that the total number of cars required to satisfy $\mathcal{H}_U$ is minimized, the total enjoyability of the system is maximized and the following constraints are satisfied: *(i)* no user is both passenger and driver; *(ii)* each vehicle holds no more than γ passengers; *(iii)* each user can be found in only one vehicle.

Optimization Problem

Given the enjoyability and mobility patterns described above we formulate the problem using an integer linear program. We start from a set of users that will be grouped together into cars. Within each car only one of the users is a driver while the other ones are defined as passengers. The number of drivers in the system indicates the number of cars allocated by the algorithm for the entire set of users. The grouping process is regulated by two aspects: *(i)* trajectory containment; *(ii)* enjoyability between users. The optimization procedure takes as input the enjoyability values and the set of recommendations and generates the optimal assignment A_{R_U}. From the recommendation set R_U we can build three sets: D, the set of candidate drivers in the system; P, the set of candidate passengers (D and P may overlap in the recommendations, but not in an assignment); C, the set of possible couples (i, j) driver-passenger. We define the following parameters:

- a parameter m_{ij} describing the normalized trajectory distance, with $m_{ij} \in [0, 1]$ if driver i can give a ride to passenger j. We set $m_{ij} > 1$ otherwise. We call M the set of all m_{ij} with $i, j \in U$;
- a parameter $\bar{e}_{ij}$ that describes the unenjoyability of two users traveling together, $\bar{e}_{ij} \in [0, 1]$ where 1 indicates that users i and j are not prone to travel together and 0 indicates that users i and j are prone to travel together. Further, $\bar{e}_{ii} = 1$ so as to indicate that a user will not enjoy traveling alone.

Additionally, we also define the following variables:

- a binary variable x_{ij} that describes the assignments between drivers and passengers, $x_{ij} = 1$ if i is the driver of passenger j, $x_{ii} = 1$ if i is a driver and zero otherwise;
- a binary variable y_{jki} indicates whether two passengers share the same car, $y_{jki} = 1$ if passengers j and k share the same car with driver i, and zero otherwise;

The optimization model finds the minimum over x_{ij} of the following objective function:

$$\alpha\rho \sum_{i \in D} x_{ii} + (1-\alpha)\Big(\sum_{(i,j) \in C} \bar{e}_{ij} \cdot x_{ij} + \sum_{i \in D} \sum_{(i,j)(i,k) \in C, j \neq k} \bar{e}_{jk} \cdot y_{jki} \Big) \tag{10.1}$$

where the parameter ρ is the cost of adding a new car to the system. The purpose of ρ it is to balance the two objectives of the optimization function: α and $(1-\alpha)$ are weights for the number of cars, and for the total unenjoyability in the system (the lower the better), respectively. The data-driven method to compute α, and ρ is explained in Section 10.2.4.

The optimization is subject to:

$$\sum_{j \in P} x_{ij} \leq \gamma x_{ii}, \forall i \in D \tag{10.2}$$

where the maximum number of passengers per car is set to γ.

$$\sum_{i \in D} x_{ij} = 1, \forall j \in P \tag{10.3}$$

where one driver has to be assigned to only one car.

$$m_{ij} \cdot x_{ij} \leq 1, \forall (i,j) \in C \tag{10.4}$$

a limit different than 1, within $[0,1]$ may be taken instead, to restrict the set of recommendations to take into account. For the sake of broader optimization, we take them all.

$$y_{jki} \leq x_{ij} \tag{10.5}$$

$$y_{jki} \leq x_{ik} \tag{10.6}$$

$$y_{jki} \geq x_{ij} + x_{ik} - 1 \tag{10.7}$$

$$\forall i \in D, j \in P, k \in P : (i,j) \in C, (i,k) \in C, j \neq k$$

that are used to linearize the relation $y_{jki} = x_{ij} \cdot x_{ik}$.

The algorithm proposed aims at minimizing the number of cars jointly with maximizing the enjoyability of the system (formulated as minimization of the unenjoyability for convenience here). The output is to group passengers in cars and at the same time ensure that they will enjoy the ride in each car.

10.2.2 Method

In this section, we present the GRAAL methodology (as well as some baselines), to derive an optimal assignment starting from Twitter data. While the problem formulation was intentionally left generic and agnostic to the real dataset used, this methodology assumes Twitter as sole source of data, although other compatible types can be used.

Assumptions

Twitter may be not the perfect source of data for any of the three dimensions (text, trajectory, and co-presence) that we need. However, it is among the few public ones providing some information in all of them. We tackled the problems arising by not having ideal data as follows:

- Co-presence: we estimate the co-presence of two users in a cell at the same time, and thus the mobility demand of users, by aggregating several days of data.
- Trajectory: as geo-tagged tweets are too sparse to track users between origin/destination pairs, we assume every user is following the best path between them, which we compute by running the same journey planner for every pair of user locations.

- Topic mining: tweets are short, and the typical usage of Twitter include typos, abbreviations and slang. However, topic extraction via *Latent Dirichlet Allocation* [38] is typical on documents, and shown to be usable also on Twitter [325].

Moreover, we work under the following assumptions, which are common in this context: *(i)* we assume all the users in the system travel by car; *(ii)* we assume all the cars moving from A to B follow the trajectory returned by a journey planner used by all the cars; *(iii)* we assume users accept the recommendations; *(iv)* we assume to be working on frequent, recurring mobility, rather than solving the on-demand carpooling problem; *(v)* having divided the space into a grid of cells, we perform the geo-match on the center of the cells.

GRAAL Algorithm

Alg. 13 shows the steps performed by our methodology to solve the socially-optimal carpooling problem. The algorithm takes five parameters, and in Section 10.2.4 we explain how to tune the last two in a data driven way through the PDE: *(i)* the bounding box where to perform carpooling, *(ii)* a spatial threshold δ, *(iii)* a temporal threshold τ to define the time-stamped locations, and to compute trajectory containment, *(iv)* α, to balance enjoyability and number of cars, and *(v)* ρ, the cost of adding a car to the result. Lines $1-3$ are used to get a geo-tagged corpus of tweets from the bounding box, to derive a set of users from it, and to filter those users with poor data. Namely, we remove users with an average tweet per day ratio below a certain threshold (see Section 10.2.4 for details), and with a ratio between average number of distinct words and number of tweets below 1.

This last step aims at removing automated tweets, and spammers. In lines $4-10$, for each user, we get her tweets (not necessarily geo-located) to build a larger corpus (geo-located tweets constitute a small fraction of the entire set of tweets), which we clean by removing stopwords and performing stemming. Then we get the users' friends list, i.e., the other users that the user is following. In line 7 we compute the vector of most visited (systematic) time-stamped locations of a user, given δ and τ. In particular, we define a spatial grid over the *boundingbox*, consisting in rectangular cells of width δ, while we slice time in non overlapping slots of duration τ. From this set, in line 8 we query a journey planner to derive trajectories connecting any two time-stamped locations in each users' L_u. Finally, in line 11, we compute the vector of topics contained in the users' documents. This is done by running a *Hierarchical Dirichlet Process* (*HDP*) [277] on the users' tweets texts. *HDP* is a parameter-free version of *Latent Dirichlet Allocation (LDA)* [38] that automatically infers the number of topics. Lines $12-13$ compute the like-mindness between any two users, and then, for each user, use the median value of it to compute the homophily in lines $14-15$. In lines $16-18$, we compute the enjoyability values between any two users. In lines $19-21$ we generate the recommendations from the set of mobility demands. In line 22, we build the matrix of mobility matches from the recommendations. Finally, we perform the multiobjective optimization in line 23 by finding a set of assignments minimizing our objective function described in Section 10.2.1.

To clarify what happens to each user in the system, let us reason from the user's perspective: assuming the user has passed the filter in line 3 (i.e., we have enough data about this user - this filter may be applied once for all, and could be lifted for different input data like mobile phone records, user-generated input, etc.), spatio-temporal as well as social and topic analytics are performed in lines 4-18 and the results associated to this user.

Algorithm 13: GRAAL (*boundingbox*, δ, τ, α, ρ)

```
   Input  : boundingbox - bounding box, $\delta$ - spatial threshold, $\tau$ - temporal threshold, $\alpha$ -
            balance enjoyability and number of cars, $\rho$ - cost of adding a car to the result,
   Output: $A_{R_U}$ - assignments
 1 $\mathcal{G} \leftarrow getTweets(boundingbox)$;
 2 $U \leftarrow getUsers(\mathcal{G})$;
 3 $U \leftarrow filterUsers(U)$;
 4 for $i \in U$ do
 5     $D_i \leftarrow getTweets(i)$;
 6     $F_i \leftarrow getFriends(i)$;
 7     $L_i \leftarrow computeTimeStampedLocations(D_i, \delta, \tau)$;
 8     $T_i \leftarrow computeTrajectories(L_i)$;
 9     $\mathcal{T}_U \leftarrow \mathcal{T} \cup T_i$;
10     $\mathcal{D}_U \leftarrow \mathcal{D} \cup D_i$;
11 end
12 $\{\vec{t_i}\} \leftarrow computeTopics(\mathcal{D}_U)$;
13 for $i, j \in U$ do
14     $lm_{i,j} \leftarrow computeLikemindness(\vec{t_i}, \vec{t_i})$;
15 end
16 for $i \in U$ do
17     $h_i \leftarrow computeHomophily(i, F_i)$;
18 end
19 for $i, j \in U$ do
20     $e_{i,j} \leftarrow computeEnj(lm_{i,j}, h_i, h_j)$;
21     $E \leftarrow E \cup e_{i,j}$;
22 end
23 for $tr_i, tr_j \in \mathcal{T}$ do
24     if $tr_i \sqsubseteq_{\delta,\tau} tr_j$ then $R_U \leftarrow R_U \cup (i, j, tr_i, tr_j)$ ;
25 end
26 $M \leftarrow computeMobilityMatches(R_U)$;
27 $A_{R_U} \leftarrow optimize(\alpha, \rho, E, M)$;
28 return $A_{R_U}$
```

In lines 19-22 an implicit “labeling” of users as possible passengers and drivers is happening. In fact, we review all the trajectories mined above, and we find matches between them. If, for a given user, there are no matches at all, this user will not be in the R_U set, and will be driving a single occupancy vehicle on her own. These users are not considered in the optimization at all, as no recommendations are possible for them. For every other users, generally speaking, it is true that they may be considered as either passengers or drivers. If user A has a trajectory including one of the trajectories of user B, and user B has a trajectory including one of the trajectories of user C, then A can potentially become a driver, B can potentially become either a driver or a passenger, and C can potentially become a passenger. However, the optimization in line 23 takes all these possibilities into account, and a user is finally either a passenger, or a driver, but cannot be both. In other words, we do not pre-select who are the drivers, and who are the passengers, but this is rather automatically discovered by the optimizer.

Complexity

The complexity of GRAAL is dominated by the optimization step. Optimization problems involving discrete decision variables are NP-Hard in general [282]. However, as this may be optionally replaced by heuristic approaches, for the sake of completeness we report also the complexity of the other relevant steps: *computeTimeStampedLocations* and *computeTrajectories* are linear in the number of locations; regarding HDP the time to process individual documents increases due to increased density, leading in the worst case to a super-linear increase (cubic in the number of terms) [187]; *computeLikemindness* is constant, but it is executed in lines $12-13$ which are quadratic in the number of users; in the same way, the computation of homophily in line 15 is constant but is repeated linearly in the number of users; lines $16-18$, computing the enjoyability which takes constant time for each pair of users, is quadratic in the number of users; line 21 is executed in a nested **for** loop which is quadratic in the number of trajectories.

Baselines

We compared GRAAL to a number of baselines, which we describe here informally. We compared with a random approach, a heuristic approach maximizing the enjoyability, and against GRAAL used with two particular values of α. Additionally, we used an approach based on the same rationale behind [76], maximizing the number of friends in a car. However, as the goal of the latter is different, and as their method also solves a different version of the carpooling problem, we present different types of results for it in Section 10.2.4.

All the baselines start from a set of recommendations R_U computed as described in this Section. Then, they each return a (potentially different) subset of it, together with the recommendations on the single occupancy vehicles that constitute different sets of assignments A_{R_U}. To describe the first two baselines, consider the set of recommendations R_U as a directed graph G_{R_U} built by putting a directed edge (i, j) if j can get a ride from i.

- *Random*: we rank randomly the edges of G_{R_U}, then we take the first edge (i, j) in the rank and, if i has not been already selected as a passenger and there are less than $\gamma = 4$ assignments (see Sec. 10.2.1) with i as driver, then we flip a coin: with probability 0.5, we thus remove all the edges linked to j and produce the assignment (i, j). Otherwise, we proceed to the next edge, and repeat the procedure for all subsequent edges in the ranking. If, at the end of the procedure, there are nodes (passengers) for which no final recommendation was made, they become drivers of SOV.

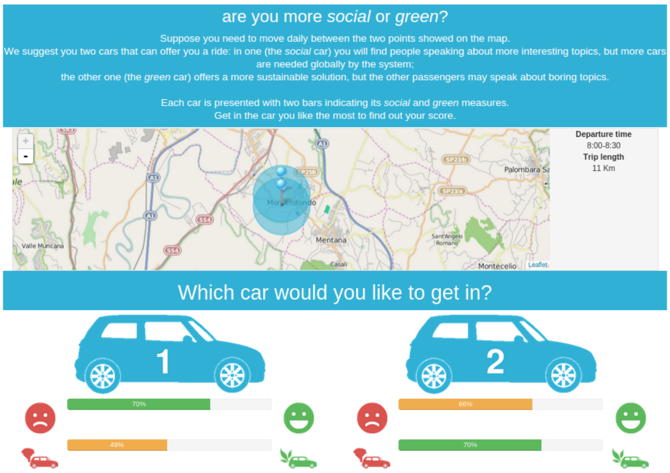
- *Heuristic*: we maximize the enjoyability with a greedy approach. We proceed like in *random* but the edges of G_{R_U} are ranked by descending enjoyability e_{ij}.

- *Social*: this is GRAAL with α=0, i.e. we maximize only the total enjoyability.

- *Green*: this is GRAAL with α=1, i.e. we minimize only the total number of cars.

10.2.3 Studying Users Preferences

In order to assess the effect of enjoyability in carpooling compared to other factors like sustainable mobility, we conducted a survey with potential end-users. The goal of this user study is to learn a crowd-sourced value for the weight α from the PDE. The survey was sent



are you more *social* or *green*?

Suppose you need to move daily between the two points showed on the map.
We suggest you two cars that can offer you a ride: in one (the *social* car) you will find people speaking about more interesting topics, but more cars are needed globally by the system;
the other one (the *green* car) offers a more sustainable solution, but the other passengers may speak about boring topics.

Each car is presented with two bars indicating its *social* and *green* measures.
Get in the car you like the most to find out your score.

Departure time
8:00-8:30
Trip length
11 Km

Which car would you like to get in?

Figure 10.16: Part of the landing web page of the survey

via direct Twitter messages, other social networks (e.g. Facebook, LinkedIn, etc., including dedicated carpooling groups), and direct e-mail and mailing lists. The webpage containing the survey is shown in Fig. 10.16. To generate the landing page, we picked a user $i \in U$ from our data, and computed which cars he/she would be assigned to using the two approaches (one minimizing the number of cars and the other maximizing enjoyability). The two solutions presented contain the following: *(i)* a bar indicating the average enjoyability among the occupants of the car; *(ii)* a bar indicating the "greenness" of the solution, computed as the collective amount of cars saved by the city-wide system if all the users were to click on this choice. The two cars were presented in random order, to minimize the probability of clicks performed on a given column. The two presented solutions are referred to as "social choice" and "green choice". The first one is the car with higher enjoyability but lower greenness value (obtained by Social), while the second choice is the car with lower enjoyability but higher greenness value (obtained by Green). Note that, while the enjoyability is a local property of the car, the greenness is a global, city-wide, property. That is, there are only two values of greenness for a city: the one obtained if every user were to click on the social choice, and the one obtained if every user were to click on the green one. After this step, the users were directed to a subsequent set of general questions on carpooling, including the following: "which of the following would make carpooling more attractive to you? Savings, sharing the car with interesting people, or sustainability of the solution?"

We collected 237 answers, with 39% in favor of a social solution. After collecting the answers, the values are exploited to learn the weight α in the multi-objective optimization model (i.e., the value of α) which represents how much the users are more likely to prefer the Social car with respect to the Green one. As mentioned, the page presents two cars with their enjoyability values of e_S (the enjoyability of the Social car) and e_G (the enjoyability of the Green car). If their difference ($e_S - e_G$) is high, meaning that the social car has a high value of enjoyability, while the green car has a low value for it, we may expect the user to be tempted to click on the social car, rather than the green one. As the greenness values of the Social and Green car for a given city are fixed (i.e., they do not change if a different pair of solutions is displayed), we do not take them into account in the learned weight. Instead, we consider the difference of enjoyabilities between the green and the social car, which depends on the pair of solutions displayed. We define the following two values: for the Green car, the value v_G is given as: $v_G = e_S - e_G$, while for the Social car, the value v_S is computed as: $v_S = 1 - (e_S - e_G)$. Given S, the set of the social choices that were obtained from the survey and G the set of the green choices, the values v_S and v_G are computed on their elements and the weight α is defined as the following ratio:

$$\alpha = \frac{\sum_{j=1}^{|G|} v_{G_j}}{\sum_{i=1}^{|S|} v_{S_i} + \sum_{j=1}^{|G|} v_{G_j}}$$

10.2.4 Case Study

In the following we report the results of running GRAAL and the baselines on real Twitter data. We employed the *Twitter* dataset described in Section 6.3. We present the tools used, the parameter tuning (this includes the results from the user study), the results of computing the social measures, and the results of GRAAL and all the baselines. The results of optimization were assessed from a city-wide collective perspective, i.e., by looking at the total values of the components of the objective function, and from an individual user perspective, i.e., looking at the distribution of the enjoyability of single cars and at the user impact with respect to carpooling.

Tools

GRAAL was written in Java and C, making use of external libraries for specific tasks. We used a publicly available Java implementation of HDP[3], to perform non-parametric topic modeling. To execute route planning we used OpenRouteService[4], a public Java library. As *space-dist* and *time-dist* we used the geo-spherical distance and the absolute difference respectively. To perform the optimization steps, we used the C APIs of IBM *CPLEX*[5].

Parameters

To run GRAAL on our data, besides the parameters of Alg. 13, we have to choose a sample of the data (in number of days) and a number of topics to put in the topic vectors. We decide to leave the bounding box, and the spatio-temporal parameters δ and τ as data driven tunable parameters of the PDE. This allows to try different optimizations in function of different temporal and spatial resolutions. In our experiments, we report results for δ set to 500 and 70 meters, and for τ set to 30 or 60 minutes. Note that the combination $\delta = 500m$ and $\tau = 30min$ agrees with common sense, or best practice, in journey planning: users are typically willing to walk distances up to 500 meters, and have a flexibility of waiting up to 30 minutes to find a means of transport [124]. In terms of number of most frequent locations, we chose 3 as it typically covers home, work, and the so called "third place". To decide the number of days of data to take, we saw that the ratio of people for which at least one

[3] https://github.com/arnim/HDP

[4] http://openrouteservice.org/

[5] http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/

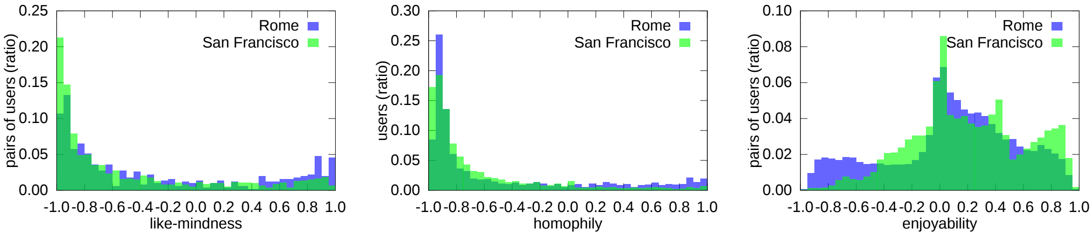


Figure 10.17: Social measures for all the couples of users.

of the top 3 locations changes if we take more than x consecutive days drops dramatically after $x = 40$. We thus chose to take 40 consecutive days of data in our sample.

We adopted a nonparametric HDP algorithm to estimate the number of topics automatically. Since HDP is nondeterministic, we ran it 2,000 times on our data, obtaining on average 25.48 topics (σ=1.56) on Rome and 25.61 (σ=1.54) on San Francisco. According to this, we selected the results relative to a number of topics of 25, to construct our vectors $\vec{t_i}$.

Parameter ρ is defined as the cost of adding a car to the result. We studied the effects of varying this parameter, in term of number of cars saved by varying $\rho \in [0, 10]$, and observed that ρ=2 had the largest impact on the number of cars saved.

The α parameter was learned looking at the results of the user study conducted as described in Section 10.2.3. We collected 237 responses coming from three different sources: 2% came from direct messages sent via Twitter; 12% came from sharing the survey in other social networks; 86% came from direct e-mail or mailing lists sharing. In total, 39% of people clicked on the social choice. This is encouraging, as it confirms the need for a social-aware carpooling system. Another encouraging result was provided by the answers to the additional survey question: 24% of the people was more attracted by sharing the car with interesting people, while 41% by the savings provided by carpooling, and 35% considered the sustainability to be the most attractive aspects of carpooling. We consider these numbers as a measure of the potential impact of a carpooling system taking into account *also* the enjoyability of a car, rather than just minimizing the cars. The final value we obtained for α, computed as explained in Section 10.2.3 is 0.36.

Results on Social Measures

Fig. 10.17 presents the distributions of like-mindness (top left), homophily (top right) and enjoyability between pairs of users (bottom row) for all the users. We report no significant differences in like-mindness and homophily between Rome and San Francisco. We observe that, computing a similarity based only on the like-mindness may end up recommending connections in a limited number of pairs of users. On the other hand, from the second plot, we learn that most of the people are heterophilous. If we combine the two things into the enjoyability, we see, in the third plot, that there is broader space for recommendations based on this measure, rather than the like-mindness. Moreover, the combination of the first two measures produces different distributions for Rome and San Francisco, highlighting that the enjoyability is capturing a different phenomenon than just the like-mindness.

δ	τ	Rome				San Francisco			
		$\lvert R_U\rvert$	$\lvert S_1\rvert$	$\lvert S_2\rvert$	$\lvert Z\rvert$	$\lvert R_U\rvert$	$\lvert S_1\rvert$	$\lvert S_2\rvert$	$\lvert Z\rvert$
500	60	6,883	81.56%	76.04%	18.44%	2,298	68.63%	57.41%	31.37%
500	30	5,870	79.84%	73.51%	20.16%	1,106	54.37%	36.88%	45.63%
70	60	349	26.85%	15.46%	73.15%	245	16.73%	9.60%	83.27%
70	30	309	24.68%	13.92%	75.32%	250	16.44%	9.79%	83.56%

Table 10.3: Statistics on user recommendations by δ and τ for GRAAL. $S_1 \subseteq U$ is the set of users with one or more recommendations, $S_2 \subseteq U$ is the set of users with two or more recommendations, $Z \subseteq U$ contains the users with no recommendations.

Results on Recommendations

Tab. 10.3 reports some statistics for the recommendations using different spatio-temporal resolutions. The first column reports the number of recommendations, in column S_1 we see the percentage of users with one or more recommendations, in column S_2 we see the percentage of users with two or more recommendations, (for which the optimization has more impact), while in column Z we report the percentage of users with no recommendations (these will end up being drivers of single occupancy vehicles in all the models). From this table, we see the clear effects of taking the same number of users in the two cities having very different geographical structure. In particular, San Francisco Bay Area is a much larger area than Rome. As carpooling in San Francisco works actually across the entire area, while it would not make much sense to keep the same user density per area and reduce the area over San Francisco, we decided not to take any corrective actions. In this way, we could also assess the effects of having different recommendation densities on the performances of the optimization. Thus, we report a larger room for optimization in Rome in general, and for δ=500m in general as well. In San Francisco, only δ=500m provides significant room for optimization. We expect this to be seen in the results at the city level.

Collective City-wide Perspective

All the parameters tuned and the recommendation calculated as explained in the previous section are then used to run the optimization. We considered them as applied on a single day of trips. The results are here presented at the city level, i.e., at the level of the entire optimization. GRAAL is able to save up to 57% of the cars needed in Rome and 40% in San Francisco, while the total enjoyability is up to double. We studied the variation of $\alpha \in [0, 1]$ (steps of 0.05), and in particular α=0.36, on the total number of cars saved and the total enjoyability of the system. We compared GRAAL for α=0.36 with all the baselines.

Fig. 10.18 reports the number of cars saved (in the top row) and the total enjoyability (in the bottom row) for Rome (in the left column) and San Francisco (in the right one). As we can see, the best performance is reached for the city of Rome with $\delta = 500m$ and for any values of τ. In all the other cases (and in San Francisco as well) we see a mostly flat behavior, which means that with less room for optimization, α can not make a big difference in the results. Moreover, $\delta = 70$ implies the lowest number of cars saved and the lowest enjoyability in both cities. This agrees with the lowest numbers in the S column of Tab. 10.3. Moreover, where the Z column of Tab. 10.3 is very high, we see negative values of enjoyability. This is due to the large number of people going alone, for which we assign an ejoyability score of -1, as described in Section 10.2.1.

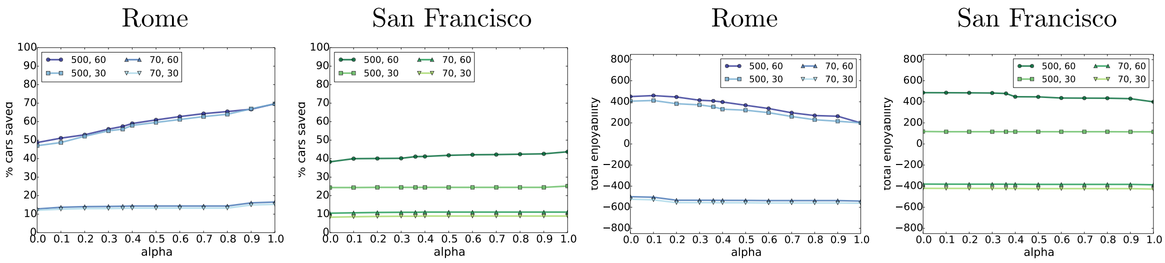


Figure 10.18: Cars saved (top row) and total enjoyability (bottom row), in Rome (left column) and San Francisco (right column) by running GRAAL with 20 values of α and different values of δ and τ. For all the plots, higher is better.

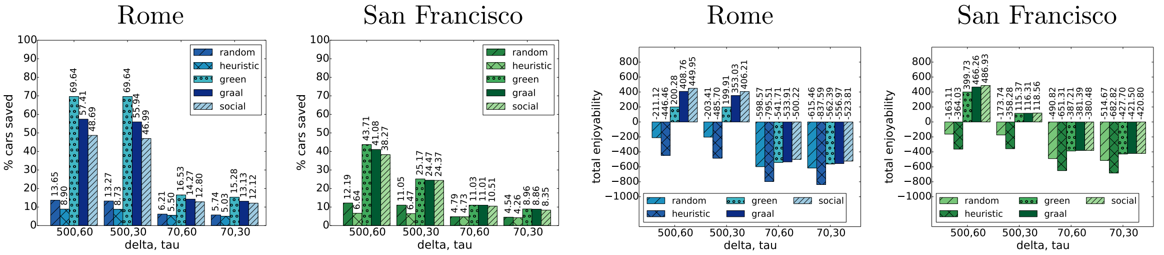


Figure 10.19: Cars saved (top row) and total enjoyability (bottom row), in Rome (left column) and San Francisco (right column) by running GRAAL with $\alpha = 0.36$ and all the baselines. For all the plots, higher is better.

Fig. 10.19 shows the percentage of cars saved (in the top row) and total enjoyability (in the bottom row), in Rome (left column) and San Francisco (right column) by running GRAAL with $\alpha = 0.36$ and all the baselines. As expected, the highest number of cars is saved by the Green approach. One encouraging result is that Social saves a significantly higher number of cars with respect to Random and Heuristic. This is due to the choice of assigning -1 as enjoyability to a person traveling alone. As a consequence, even if Social does not directly minimize the number of cars, it tends to put more people together anyway. The GRAAL approach with $\alpha = 0.36$ is a trade-off between Social and Green (which are basically GRAAL with the two possible extreme values for α).

Consider now the bottom row of Fig. 10.19, with $\delta = 500m$. The negative total enjoyability confirms that in those cases there is a significant number of people going alone. This is avoided by GRAAL with all alpha values (as reported in Fig. 10.18). In accordance with Tab. 10.3, reporting a high number of single occupancy vehicles for $\delta = 70m$, we have only negative total enjoyability for all models for this value of δ.

Finally, we report the results of comparing GRAAL with the Green model minimizing the number of cars, in terms of two KPIs: additional cars used, and additional km traveled by the cars in the system. Tab.10.4 reports these values for each city and combination of δ and τ. In the "% cars" cell, we report the percentage of additional cars used by GRAAL with respect to Green, normalized by the number of cars needed if all the users were taking a car. In the "% km" cell, we report the percentage of additional km traveled by the GRAAL drivers with respect to Green, normalized by the total amount of km traveled if all the users were taking a car. The first column can be seen as a way to measure the cost of adding a car to the system (for example, in terms of parking slots needed), while the second column can be seen as a way to measure the overall cost of the system (for

δ	τ	**Rome**		**San Francisco**	
		% cars	% km	% cars	% km
500	60	12.23	3.67	2.63	0.02
500	30	13.70	4.39	0.70	0.43
70	60	2.26	0.35	0.02	0.01
70	30	2.15	0.38	0.10	0.32

Table 10.4: Percentages of additional cars and km needed by GRAAL with respect to the Green.

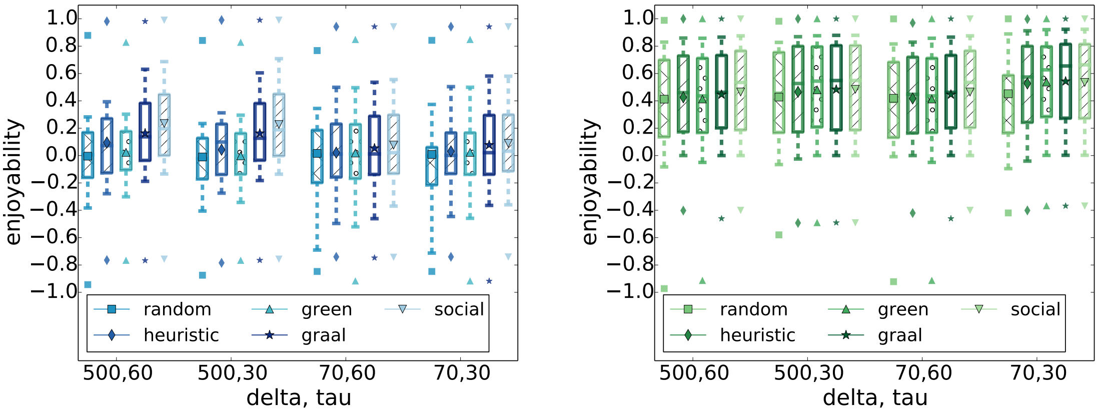


Figure 10.20: Enjoyability per cars (min, max, 10th, 25th, 50th, 75th, 90th percentiles, and average across all cars), for Rome (left) and San Francisco (right). Higher is better.

example, in terms of CO2 emissions). As we see, although we add up to 13% of cars into the system with GRAAL, they are typically used to cover short distances, as the additional km traveled, in percentage, are well below the percentage of cars added. We highlight a detail: in our model, drivers are not allowed to detour to pick up passengers. That is, giving a lift to someone always subtract distance from the total traveled.

Individual User Perspective

We assess the results from the user perspective, in terms of enjoyability in the single cars. As aggregates, we report minimum, maximum, average, 90th, 75th, 50th, 25th, and 10th percentiles of the distribution of the enjoyability across vehicles, in order to understand the improvement introduced for a user, in Fig. 10.20.

For this assessment, we consider only the users who received a recommendation. That is, we remove most of the effects of considering an enjoyability equal to -1 for a high number of people in these plots. The first clear result is that there is a globally higher enjoyability in San Francisco, compared to Rome. This is coherent with the results on the distribution of the enjoyability per city reported in Fig. 10.17, which shows both a higher negative tail in Rome for the enjoyability, and a higher positive tail for San Francisco. Despite the globally higher enjoyability, there is again the problem of the results being flatter than in Rome. Consider now the results in Rome, with $\delta = 500m$. In the Green model, where the optimization disregards the enjoyability, the results are inline with Random, while Heuristic does a better job. This is also true for the other value of δ, although less evident.

Consider now Tab. 10.5. We want to compare with the method described in [76], which tries to put friends (i.e., direct Twitter links) together, as their concept of enjoyability. We evaluate against them in terms of impact on users, reported in Tab. 10.5, which contains

δ	τ	Rome				San Francisco			
		$\|R_U\|$	$\|S_1\|$	$\|S_2\|$	$\|Z\|$	$\|R_U\|$	$\|S_1\|$	$\|S_2\|$	$\|Z\|$
500	60	189	11.36%	7.42%	88.64%	148	7.57%	4.89%	92.43%
500	30	183	11.12%	7.02%	88.88%	120	7.50%	4.23%	92.50%
70	60	53	9.37%	4.40%	90.13%	46	5.82%	2.31%	94.18%
70	30	51	9.40%	4.32%	90.60%	45	4.35%	2.25%	95.65%

Table 10.5: Statistics on user recommendations by δ and τ for [76]. $S_1 \subseteq U$ is the set of users with one or more recommendations, $S_2 \subseteq U$ is the set of users with two or more recommendations, $Z \subseteq U$ contains the users with no recommendations.

the same columns as Tab. 10.3. To produce it, we first ran Green, then we applied brute force to optimize by friendship (friends are put together in a car). As we clearly see, the number of recommendations between friends is much smaller than what we can achieve in GRAAL reported in Tab. 10.3, due to the sparsity of the friendship connections in Twitter (and in the real world, too), as opposed to the fact that we could compute the enjoyability between any two users in GRAAL. Thus, the room for optimization here is much smaller, with numbers in S_2 not reaching two digits. We put the S_1 column in the two tables to give more chances to this approach. In fact, even if we can optimize less, with at least one recommendation we can still put friends together. Nevertheless, numbers go up to slightly more than 11%. We did not compare with the same approach ran with a 2-hop network for the following reason: 2-hop friends (i.e., friends of friends), when they are not direct friends, are people with whom we can not give any guarantee on the enjoyability from a topic perspective, and neither they are direct friends. On the other side, 2-hop friends could be at least more trustworthy than unknown (but enjoyable) people. However, neither our methodology, nor the one in [76] are meant to be seeking a higher trust in the system, which is then left as future work.

Running Times

GRAAL ran in around 2 minutes with each of the α values under each of the δ and τ combinations, for both cities. Exceptions were $\delta = 500m$ in Rome, where a higher number of recommendations brought the running times up to 1 hour.

10.2.5 Conclusion

GRAAL is a multiobjective method that, through Personal Data Analytics exploits the Personal Data Models of the users in the Personal Data Ecosystem at individual and collective level to optimize carpooling recommendations for a weighted linear combination of number of cars used (which is minimized) and total enjoyability (which is maximized). GRAAL takes Twitter data in input, as this contains information on spatio-temporal, text, and social dimensions of geo-located user tweets. Through a survey we have tuned the weight of the linear combination in the optimization function. We have presented the results of the multiobjective optimization in terms of cars saved and enjoyability both from the city and the user perspective. With the crowd-sourced alpha, GRAAL is able to save up to 57% of the cars needed among those considered for matching, while the total enjoyability is up to double. From the user perspective, we have shown how the entire per-car distribution of enjoyability is increased with respect to the baselines.

Chapter 11

Socio-Economical Analysis of Well-Being

The availability of huge quantity of retail market data stimulates more and more challenging questions that can be answered by deep and smart analyses of different aspects related to shopping sessions of customers. Retail data is a really complex type of data. Indeed, it contains a wide set of different dimensions that can be analyzed under many points of views. The main dimensions are: *what* customers buy, i.e., the basket composition, *when* and *where* they make the purchases and which is the *relevance*, in terms of money spent or quantity, of the purchase. The choice of analyzing a set of dimensions rather than another one depends on the kind of phenomena to be investigated: considering all the dimensions in the same analysis can lead to very complex models or to weak generalizations.

In the previous part of this thesis we analyzed retail data from an individual point of view by generating personal indexes able to estimate the level of predictability in shopping habits. On the other hand, in this chapter we analyze these data from a collective point of view. In particular, we look for an added level of knowledge generated by the collective analysis but which starts from individual models. We propose two analysis. In the first one we develop a Personal Data Model for retail data which captures the temporal dimensions of shopping and we exploit the patterns extracted to group customers of the PDE having similar shopping trends. In the second one, we exploit data mining and complex network analysis to produce from the PDE a collective measure able to nowcast the GDP.

11.1 Discovering Temporal Shopping Regularities

We are interested in understanding *whether* and *when* a customer makes *typical* purchases. Which of these purchases are more systematic for the customer? Which are the *regular* sequences of shopping that the customer performs? To this aim, we define a *temporal purchasing profile* for Personal Data Analytics as part of the Personal Data Model that is able to describe the regular and characteristic temporal behaviors of an individual customer. The *individual person* is the key element that lies in between a single purchase and a whole customers population, i.e., the Personal Data Ecosystem. Each individual has her own regularities and habits outlining her behavior and making her a unique part of the mass. The analysis of individuals provides the basis for understanding routines in the purchasing behavior both at individual and collective level.

The "data unit" used is a temporal purchasing footprint, i.e., a vector estimating the shopping relevance during a time period. Our definition of temporal purchasing footprint of a customer is similar to the definition of user profile introduced in [105]. In [105] the authors extract from individual call detail records a profile summarizing the calls of a user. Their aim is to estimate the proportion of city users that can be classified as residents, commuters, visitors. We define the *temporal purchasing profile* of a customer as the set of her temporal shopping behavioral *footprints* and her *sequence of footprints* summarizing *whether* and *when* the customer typically makes a set of similar purchases. Then we define a method to provide to the individual and not-comparable profiles a *collective perspective* which makes them comparable and able to describe the shopping routines shared in a Personal Data Ecosystem by different customers. Note that most of the works in the literature are centering their attention on catching and comprehending the behaviors and habits by analyzing *what* customers buy [6, 165]. Just a few of them have exploited also the temporal dimension as a feature for enriching their models based primarily on the items purchased [129, 194, 203]. However, to the best of our knowledge, there is no previous work focusing on the temporal dimension i.e., the information about *when* a purchase is performed) and using it as the main building block to construct an individual temporal purchasing profile.

Our findings reveals three main typical collective behaviors characterizing the whole collection of customers on the basis of when they shop: *daily* spending behavior capturing purchases made every day; *one-shop* spending behavior, characterizing a regularity with a week containing a predominant shopping session; and an *occasional* spending behavior, describing a not habitual shopping sessions related to a very small expenditure amount. Among *one-shop* spending behaviors the analysis captures a further classification in with respect to the expenditure amount: *normal* spending behavior less than €50, *high* spending behavior with a typical expenditure between €50 and €100, and *big* spending behavior with an expenditure higher than €100. Finally, the most interesting finding is the identification of two categories of customers that we name *regular* and *changing*. We discover them by analyzing the number of purchasing behaviors characterizing each customer: a customer with a high number of behaviors is classified as *changing*, while a customer with a small number of temporal shopping behaviors is classified as *regular*.

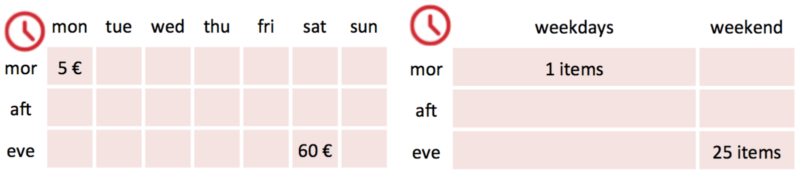
11.1.1 Method and Model

We represent a *shopping session* as a tuple $s = \langle customer, timestamp, shop, basket, amount \rangle$ containing information about the *customer*, the *timestamp* and the *shop* of the purchase, the *basket* composition and the *amount* spent. In the following, we do not consider the items composing the *basket* and the *shop*.

Individual Model

For each customer, we summarize the temporal information of a set of shopping sessions by introducing the notion of *temporal purchasing unit* (unit in short):

Definition 46 (Temporal Purchasing Unit)**.** *Given a period τ of $\bar{d}$ days, a* temporal purchasing unit *U is a matrix $U \in \mathbb{R}^{t \times d}$, where d is the number of day-intervals in τ with $d \leq \bar{d}$, t is the number of time windows considered for each day-interval, and U_{ij} estimates the relevance of the purchases in the i-th time window of the j-th day-interval.*



	mon	tue	wed	thu	fri	sat	sun
mor	5 €						
aft							
eve						60 €	

	weekdays	weekend
mor	1 items	
aft		
eve		25 items

Figure 11.1: Units with different day-interval granularity: single day (left) and weekdays-weekend (right).

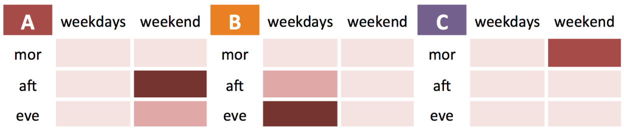


Figure 11.2: Footprints: the darker a cell the higher the relevance for a specific day-time-window.

With *day-interval* we refer to any aggregation of days, e.g., single day, weekdays-weekend, etc., while with *time window* we refer to any aggregation of hours, e.g., single hour, morning-afternoon, etc. Given a period τ, each unit U captures the relevance of the customer's shopping sessions during the period τ discretized into specific temporal slots. The *relevance* can be expressed by any value referred to shopping sessions: the total amount spent, the number of items bought, the number of shopping sessions, etc.

Fig. 11.1 illustrates some examples: on the left, a unit with three time windows *morning*, *afternoon*, and *evening* and the day-interval granularity set to single day; on the right, a unit with day-interval set to weekdays-weekend. In the left unit the relevance in F_{ij} represents the amount spent, while in the right one contains the number of items bought. The setting of time granularity and relevance depends on the aim of the analysis.

Given a customer c, her sequence of temporally ordered shopping sessions $S = \{s_1, \dots, s_n\}$, the temporal granularity for time window and day-interval t and d, and the width of the time period τ, then S can be *segmented* into a sequence of units $\widehat{S} = \{U^{(1)}, \dots, U^{(m)}\}$ with $m \leq n$. For example, if $S = \{s_1 = \langle Mon1 - h9, €5\rangle, s_2 = \langle Sat6 - h18, €60\rangle, s_3 = \langle Mon8 - h17, €10\rangle\}$ with $\tau = 7$, $d = 7$, $t = 3$ we obtain $\widehat{S} = \{U^{(1)}, U^{(2)}\}$ where $U^{(1)}_{0,0} = 5$, $U^{(1)}_{2,5} = 60$, $U^{(1)}_{2,0} = 10$ and zeros elsewhere. In other words, $\widehat{S}$ is an ordered sequence of matrices modeling and aggregating the shopping sessions according to some parameters.

Our goal is to *summarize* for each customer the knowledge contained in $\widehat{S}$ in a *temporal purchasing profile* describing the customer's typical temporal behaviors. In order to introduce the profile we need to define the units which are "distinctive" for the customer. Given a group G of similar units we define a *temporal purchasing footprint* (footprints in short) as the *representative* of the group G.

Definition 47 (Temporal Purchasing Footprint)**.** *Given a group $G = \{U^{(1)}, \dots, U^{(q)}\}$ of units, its* temporal purchasing footprint *is defined as the centroid of G, i.e., it is the matrix $F \in \mathbb{R}^{t \times d}$ such that*

$$F_{ij} = \frac{1}{|G|} \sum_{\forall U^{(h)} \in G} U^{(h)}_{ij} \ \forall i, j$$

F captures a temporal shopping behavior characterizing the customer. Given a sequence of units, we *detect groups* of units which are similar with respect to a distance function δ, based on temporal alignment and with respect to the relevant values considered:

Definition 48 (Temporal Purchasing Footprint Groups)**.** *Given a sequence of units $\widehat{S} = \{U^{(1)}, \dots, U^{(m)}\}$ and a distance function $\delta : \mathbb{R}^{t \times d} \times \mathbb{R}^{t \times d} \to \mathbb{R}$, the* temporal purchasing

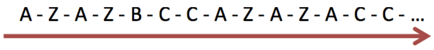


Figure 11.3: Footprint sequence: Z labels footprints indicating no shopping sessions.

Algorithm 14: *extractIndividualProfile*

```
Input  : $S$ - sequence of temporally ordered shopping sessions,
         $\tau$ - width of the time period covered by each footprint,
         $d$ - days-interval to be set in a footprint,
         $t$ - time windows to be set in a footprint,
         $rel$ - relevance function,
         $\delta$ - distance function
Output: $\mathcal{P}$ - temporal purchasing profile
$\widehat{S} \leftarrow segmentIntoUnits(S, \tau, d, t, rel)$;
$\mathcal{G} \leftarrow detectGroups(\widehat{S}, \delta)$;
$\mathcal{F} \leftarrow \{D^{(i)} | \ D^{(i)} = getCentroid(G_i) \ \forall G_i \in \mathcal{G}\}$;
$\widehat{F} \leftarrow buildSequence(\widehat{S}, \mathcal{G}, \mathcal{F})$;
return $\mathcal{P} = \langle \mathcal{F}, \widehat{F} \rangle$
```

footprint groups $\mathcal{G} = \{G_1, \ldots, G_k\}$*, are defined as a partitioning of* $\widehat{S}$ *into disjoint sets of similar footprints such that* $\forall G_i, G_j \in \mathcal{G},\ \forall U \in G_i,\ \delta(U, F^{(i)}) < \delta(U, F^{(j)})$ *, where* $F^{(i)}$ *and* $F^{(j)}$ *are the centroids of* G_i *and* G_j*.*

We name $\mathcal{F}$=$\{F^{(1)}, \ldots, F^{(k)}\}$ the *set of footprints* of a customer. Note that we are not considering the order of the footprints in $\widehat{S}$. Fig. 11.2 shows an example of footprints.

Given the groups $\mathcal{G}$ and their footprints $\mathcal{F}$, we can replace each unit in $\widehat{S}$ with the footprint representing the group to which it belongs to. We name the new sequence *temporal purchasing footprint sequence* (footprint sequence in short).

Definition 49 (Temporal Purchasing Footprint Sequence)**.** *Given a customer c, her sequence of units* $\widehat{S}$*, her groups* $\mathcal{G}$ *and her footprints* $\mathcal{D}$*, we define the* temporal purchasing footprint sequence *as the sequence* $\widehat{F}$ *obtained replacing in* $\widehat{S}$ *the units with the corresponding footprints in* $\mathcal{F}$ *according to the groups* $\mathcal{G}$*.*

For example, given $\mathcal{F}$=$\{F^{(1)}, F^{(2)}\}$, $\mathcal{G}$=$\{G_1, G_2\}$ where G_1=$\{U^{(1)}, U^{(4)}\}$, G_2=$\{U^{(2)}$, $U^{(3)}\}$, if $\widehat{S}$=$\{U^{(1)}, U^{(2)}, U^{(3)}, U^{(4)}\}$, we can construct the footprint sequence as $\widehat{F}$=$\{F^{(1)}$, $F^{(2)}, F^{(2)}$, $F^{(1)}\}$. Fig. 11.3 depicts the footprint sequence using the footprints of Fig. 11.2.

Finally, we define the *temporal purchasing profile* of a customer (profile in short) as:

Definition 50 (Temporal Purchasing Profile)**.** Given a customer c, her sequence of units $\widehat{S}$, and a distance function δ, the *temporal purchasing profile* of c is defined as $\mathcal{P}_c = \langle \mathcal{F}, \widehat{F} \rangle$ where $\mathcal{F}$, is the set of footprints derivable from the groups $\mathcal{G}$ detected on $\widehat{S}$ using δ, while $\widehat{F}$ is the footprint sequence derivable from $\mathcal{G}$, $\widehat{S}$ and $\mathcal{F}$.

Extracting Individual Temporal Purchasing Profile

The process for the extraction of the individual profiles is summarized in Alg. 14. The first step is the segmentation of the sequence of temporally ordered shopping sessions S, considering d days-intervals, t time-window for each day-interval and the relevant values

Symbol	***Description***	***Symbol***	***Description***
s	shopping session	$\mathcal{P}$	individual profile
d	day-intervals	C	collective footprint
t	time windows	L	collective group of footprint
τ	footprint width	$\mathcal{L}$	collective groups of footprint
$\widehat{S}$	sequence of footprints	$\mathcal{C}$	collective footprints of all customers
U	unit	$\mathcal{C}_c$	customer collective footprint
F	footprint	$\widehat{C}_c$	collective sequence
G	group of footprint	R_i	regular sub-sequence
$\mathcal{G}$	groups of footprints	w_i	support
δ	distance function	ω	support threshold
$\mathcal{F}$	footprints	$\mathcal{R}_c$	regular sub-sequences
$\widehat{F}$	footprints sequence	$\mathcal{P}^*$	collective perspective

Table 11.1: Symbols and descriptions.

returned by *rel* (*segmentIntoFootprints* function in Line 1). The result is the sequence of units $\widehat{S}$ where each unit covers a time period of width τ. Given $\widehat{S}$, *detectGroups* (Line 2) applies a clustering method to find groups of similar units on the basis of the distance function δ. An appropriate clustering method and distance function can be selected according to the aim of the analysis. Once the groups of units $\mathcal{G}$ are detected, from each group the *getCentroid* function (Line 3) extracts the centroid $F^{(i)}$ representing a footprint. Then, the footprint sequence $\widehat{F}$ is built considering $\widehat{S}$, $\mathcal{G}$ and $\mathcal{F}$ using the function *buildSequence* (Line 4). Finally, $\mathcal{F}$ and $\widehat{F}$ form the temporal purchasing profile $\mathcal{P}$. The computational complexity of Alg. 14 is dominated by the complexity of the *detectGroups* function that implements a clustering algorithm. With respect to the data treated in these analyzes, the profile $\mathcal{P}$ constitutes the Personal Data Model.

Collective Perspective of Individual Profiles

To compare individual profiles of different customers we need to provide them a *collective perspective.* This means to enable the comparison among footprints and among footprint sequences of different customers of the PDE such that each customer can benefit of a collective perspective that allow the comparison with other customers. Given customers b and c and their profiles $\mathcal{P}_b$, $\mathcal{P}_c$, our aim is to make comparable $\mathcal{F}_b$ and $\mathcal{F}_c$, and $\widehat{F}_b$ and $\widehat{F}_c$. To this end, we propose an approach which outlines the one used for individual models.

We start by comparing the footprints of the customers and by partitioning them into similar groups. Given a set of customers $\{c_1 \dots c_n\}$, the set of their individual profiles $\{\mathcal{P}_c = \langle \mathcal{F}_c, \widehat{F}_c \rangle\}$ and a distance function δ, we define the *collective temporal purchasing footprint* (collective footprint in short) and the *collective temporal purchasing footprint groups* (collective footprints groups in short) as follows.

Definition 51 (Collective Temporal Purchasing Footprint)**.** *Given a collective group $L = \{F^{(1)}, \dots F^{(q)}\}$ of individual footprints, its* collective temporal purchasing footprint *is defined as the centroid of L, i.e., it is the matrix $C \in \mathbb{R}^{t \times d}$ such that*

$$C_{ij} = \frac{1}{|L|} \sum_{\forall F^{(h)} \in L} F^{(h)}_{ij} \ \forall i, j$$

Algorithm 15: *extractCollectiveFootprints*

```
Input  : ${\mathcal{P}_c = \langle \mathcal{F}_c, \widehat{F}_c \rangle}$ - profiles of all customers,
         $\delta$ - distance function,
Output: $\mathcal{L}$ - collective groups,
         $\mathcal{C}$ - collective footprints
$\mathcal{L} \leftarrow detectGroups(\{\mathcal{F}_c\}, \delta)$;
$\mathcal{C} \leftarrow \{C^{(i)} | \; C^{(i)} = getCentroid(L_i) \; \forall L_i \in \mathcal{L}\}$;
return $\langle \mathcal{L}, \mathcal{C} \rangle$
```

Algorithm 16: *provideCollectivePerspective*

```
Input  : $\mathcal{P}_c = \langle \mathcal{F}_c, \widehat{F}_c \rangle$ - temporal purchasing profile,
         $\mathcal{C}$ - collective footprints of all customers,
         $\mathcal{L}$ - collective groups of footprints of all customers
Output: $\mathcal{P}^*_c = \langle \mathcal{R}, \mathcal{C}_c \rangle$ - collective perspective
$\mathcal{C}_c \leftarrow mapIntoCollective(\mathcal{F}_c, \mathcal{L}, \mathcal{C})$;
$\widehat{C}_c \leftarrow buildSequence(\widehat{F}_c, \mathcal{L}, \mathcal{C}_c)$;
$\mathcal{R} \leftarrow regularSubsequences(\widehat{C}_c)$;
return $\mathcal{P}^*_c = \langle \mathcal{C}_c, \mathcal{R}_c \rangle$
```

Definition 52 (Collective Temporal Purchasing Footprint Groups)**.** *Given a set of individual footprints $\{\mathcal{F}_c\}$ and a distance function $\delta : \mathbb{R}^{t \times d} \times \mathbb{R}^{t \times d} \to \mathbb{R}$, the* collective temporal purchasing footprint groups *$\mathcal{L} = \{L_1, \ldots, L_k\}$, are defined as a partitioning of $\{\mathcal{F}_c\}$ into disjoint sets of similar footprints such that $\forall L_i, L_j \in \mathcal{L}$, $\forall F \in L_i$, $\delta(F, C^{(i)}) < \delta(F, C^{(j)})$, where $C^{(i)}$ and $C^{(j)}$ are the centrods of L_i and L_j.*

We name $\mathcal{C} = \{C^{(1)}, \ldots, C^{(k)}\}$ the *set of collective footprints of all customers.* Given a customer c, her footprints $\mathcal{F}_c$, the collective footprints $\mathcal{C}$ and the collective groups $\mathcal{L}$, we denote the *collective perspective* of $\mathcal{F}_c$ with the customer collective footprints $\mathcal{C}_c = \{C^{(1)}, \ldots, C^{(q)}\}$, where $\mathcal{C}_c \subseteq \mathcal{C}$ and $\forall C^{(h)} \in \mathcal{C}_c \; \exists F^{(i)} \in \mathcal{F}_c$ s.t. $F^{(i)} \in L_h$ with $L_h \in \mathcal{L}$ and $C^{(h)}$ is the centroid of L_h. Note that two different footprints $F^{(i)}$ and $F^{(j)}$ in a collective perspective can belong to the same collective group L_h and thus, they can be represented with the same collective footprint $C^{(h)}$. We underline that we use the expression *customer collective footprints* to indicate $\mathcal{C}_c$ and *collective footprints of all customers* to indicate $\mathcal{C}$.

The customer collective footprints $\mathcal{C}_c$ empowers the collective perspective to the sequence $\widehat{F}_c$. Given the collective groups $\mathcal{L}$ and the collective footprints of all the customers $\mathcal{C}$, we can replace each individual footprint in $\{\widehat{F}_c\}$, with the customer collective footprint representing the collective group to which it belongs. Hence, for each customer c her sequence $\widehat{F}_c$ is mapped to an equivalent *collective temporal purchasing sequence* $\widehat{C}_c$ (collective sequence in short) which is comparable with the sequences of the other customers.

Definition 53 (Collective Temporal Purchasing Sequence)**.** *Given a customer c, her footprint sequence $\widehat{F}_c$, the collective groups $\mathcal{L}$ and the collective footprints of all customers $\mathcal{C}$, the* collective temporal purchasing sequence *is the sequence $\widehat{C}_c$ obtained replacing each footprint in $\widehat{F}_c$ with the corresponding collective footprint in $\mathcal{C}$ according to $\mathcal{L}$.*

To understand which are the sub-sequences most used and shared among customers we define the *regular temporal purchasing sub-sequences* (regular sub-sequences in short):

Definition 54 (Regular Temporal Purchasing Sub-Sequences). *Given a customer c, her collective sequence $\widehat{C}_c$ and a support threshold ω, the* regular temporal purchasing sub-sequences *is the set $\mathcal{R}_c = \{(R_1, w_1), \ldots, (R_m, w_m)\}$, where each R_i is a sub-sequence of $\widehat{C}_c$, w_i is its support and $\forall w_i \; w_i \geq \omega$.*

In other words, among all the possible sub-sequences of $\widehat{C}_c$, $\mathcal{R}_c$ contains only the most representative for customer c. For example, if all the possible sub-sequences of $\widehat{C}_c$ are

$$(\{C^{(1)}, C^{(1)}\}, 10), \;\; (\{C^{(1)}, C^{(1)}, C^{(2)}\}, 8),$$

$$(\{C^{(1)}, C^{(2)}\}, 2), \;\; (\{C^{(2)}, C^{(1)}\}, 2), \;\; (\{C^{(2)}, C^{(2)}\}, 1)$$

where the number is the support, i.e., the number of occurrences of that sub-sequence, then only the first two sub-sequences are *regular* and contained in $\mathcal{R}_c$ if $\omega = 5$. Given two customers b and c and $\mathcal{R}_b$ and $\mathcal{R}_c$ derivable from $\widehat{C}_b$ and $\widehat{C}_c$, we can now compare b and c with a distance function on $\mathcal{R}_b$ and $\mathcal{R}_c$ like the *Jaccard* or *cosine* distance.

Finally, we can define the *collective perspective* of a profile:

Definition 55 (Collective Perspective). *Given a customer c, her temporal purchasing profile $\mathcal{P}_c = \langle \mathcal{F}, \widehat{F} \rangle$ and the collective footprints of all customers $\mathcal{C}$, the* collective perspective *of the individual profile $\mathcal{P}_c$ is defined as $\mathcal{P}_c^* = \langle \mathcal{C}_c, \mathcal{R}_c \rangle$ where $\mathcal{C}_c \subseteq \mathcal{C}$ are the customer collective footprints, and $\mathcal{R}_c$ is the set of regular sub-sequences.*

The collective perspective of a user profile is an example of how the profile of an individual can be perceived by another user in the Personal Data Ecosystem.

Providing Collective Prospective to Individual Profiles

The process for providing the collective perspective to the individual profiles is summarized by Alg. 15 & 16. Alg. 15 employs *detectGroups* to detect from the individual profiles of all the customers the collective groups of footprints (Line 1). A clustering method is used to carry out this task. From each group in $\mathcal{L}$ the *getCentroid* function (Line 2) extracts the centroid $C^{(i)}$. The union of the centroids forms the collective footprints of all the customers. For each customer, Alg. 16 provides the collective perspective to the individual profile. Using as input the output of the Alg. 15, it extracts from $\mathcal{C}$ the collective footprints providing the collective perspective to the footprints $\mathcal{F}_c$ by considering the groups in $\mathcal{L}$ (Line 1). Then, by using *buildSequence* (Line 2), the collective perspective is provided to the footprint sequence $\widehat{F}_c$ generating the collective sequence $\widehat{C}_c$ by means of $\mathcal{C}_c$. The function *extractRegularSubSequences* (Line 3) extracts from $\widehat{C}_c$, the regular sub-sequences of the customer $\mathcal{R}_c$. Finally, Alg. 16 returns the collective perspective of the profile $\mathcal{P}_c^*$.

We implement the exaction of the regular sub-sequences by means of a suffix tree [121]. Given a customer c, her collective sequence $\widehat{C}_c$ is transformed into a string where each character corresponds to the label of a customer collective footprint. Hence, we generate a suffix tree for each customer. Following a branch of the tree from the root to a leaf we can read a sub-sequence R_i and, on the leaf, we have the support w_i of the sub-sequence generating that branch. We set the support threshold ω in a data-driven way by looking at the distribution of the support of the customer sub-sequences. In particular, we apply a technique known as “knee method” [275]. Given a set of pairs composed of items and their support this method sorts the pairs according to the frequencies and returns the

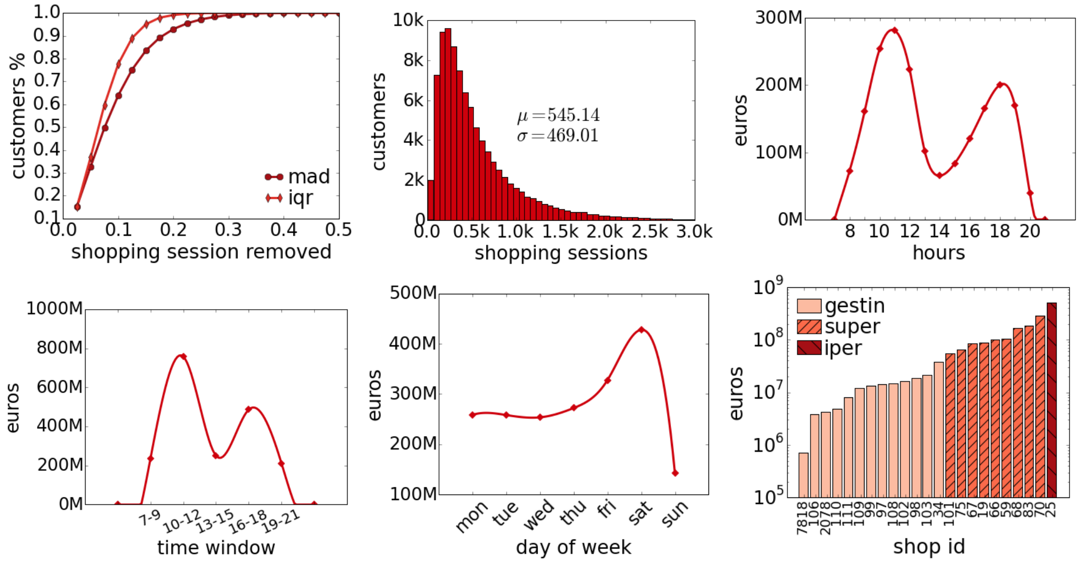


Figure 11.4: Dataset distributions: cumulative of shopping sessions removed, shopping sessions per customer, total amount spent per hours, per time window, per day of week and per shop.

most representative, i.e., the pairs with a support greater or equal than the support ω corresponding to the *knee* in the curve of the ordered frequencies. In this way ω is different for each customer and driven by personal data. For each customer, we cut the suffix tree considering only the *regular* sub-sequences, i.e., the sub-sequences R_i with support w_i greater or equal than ω. As for Alg. 14, also the complexity of Alg. 15 is dominated by the complexity of *detectGroups* that is implemented with a clustering algorithm; while Alg. 16 has a complexity which depends on the construction of the suffix tree [121].

11.1.2 Case Study

Dataset

We adopted the *Coop* dataset described in Section 6.2. For data cleaning purposes, we performed a series of filters on this dataset. We consider the 23 shops that are in Leghorn province. Indeed, the market penetration of the company in this province is so high that we can nearly say that all the inhabitants are represented in the dataset. Second, we drop all customers who did not perform at least ten shopping sessions per year in different months: sporadic customers might use their card in shops in Leghorn province, thus introducing noise in our estimates. Finally, for each customer we performed an individual filter aimed at removing possible errors and outliers: for each customer we analyzed the total amount spent in every shopping session. In Fig. 11.4 *(top left)* is reported the percentage of shopping sessions removed by using the inter-quartile range (IQR) [287] and the median absolute deviation (MAD) [148]. As the result is comparable but the inter-quartile range is more conservative we decided to use this approach to clean our data. After this filter phase, we end up with about $91k$ customers[1]. The province of Leghorn had an average population of about $343,000$ inhabitants during the years observed. Assuming an average size of two/three people per household, we estimate that we cover at least 60% of the population. The total number of shopping sessions considered amount at $49,590,010$.

[1] Note that "customer" refers to a customer card, and a card can be shared by a family or among flatmates.

Fig. 11.4 depicts stylized facts about shopping sessions. Fig. 11.4 *(top center)*: the number of shopping sessions per customer. The mode is ∼350, meaning that customers usually visit the shops around once a week. In the middle row is shown the total amount spent per time of the day. An *M-shaped* pattern appears: most shopping sessions happen in the morning or after working hours. We can summarize this trend using time windows instead of hours by reducing the intervals through the aggregation of the shopping sessions according to the data-driven time windows: 7-9, 10-12, 13-15, 16-18, 19-21. Indeed, the second and the fourth time window captures the peaks in the trend. These are the time window we are using on our experiments. In Fig. 11.4 *(bottom center)* is reported the total amount spent per weekday. Customers have a preference for shopping in days close to the weekend. Fewer shopping sessions happen on Tuesday, while Saturday is the most popular day. Finally, Fig. 11.4 *(bottom right)* illustrates the total amount spent per shop (in semi log-y). Each of the 23 shops is represented here. There is a correlation between the type of shop and the amount spent.

Experiments Setting

As humans we operate under the cadence of a *seven-day week* [321]. This cycle of activity is deeply rooted in human experience and in our psychological habits. Indeed, the weekdays alternation drives our routinary life. These are the reasons why we decided to set $\tau = 7$ and $d = 7$, i.e. in our experiments each footprint captures the behavior of a week and each day-interval corresponds to a single day. In our opinion, it is the best time discretization because is able to better schedule our life. Choosing *month* instead of *week* we might have the risk to flat the difference between some purchasing behaviors. Similarly, a contraction in weekdays-weekend would make similar customers shopping on Monday and on Friday. Thus we decided to adopt a week as time unit.

With respect to the time windows we wanted to adopt a granularity not too fine to avoid sparse matrices but which is able to capture the general trend. Considering all the hours, or even a finer granularity, would have generated very sparse matrices F and the need to employ distance function with slicing window like the dynamic time warping that have a high computational cost if compared to the Euclidean or to the cosine distance. Consequently, we applied the time window reported in Fig. 11.4 *(bottom left)* and we set $t = 5$.

As relevance function *rel* we used the total amount spent. We employed the *sum* as aggregation function because it is quite unlikely that a customer makes two distinct shopping sessions in the same time window of the same day. In the analysis we did not consider the *number of items bought* because is highly correlated with the *amount spent* for most of the shopping sessions. The mode of the distribution of this correlation is ∼0.85, and the average p-value is < 0.00005. Hence, the results of the analysis obtained for the models considering as relevance the euros spent are comparable with those that we would have obtained using the number of items bought. We have not used the *number of shopping sessions* because two shopping sessions of €50 and €5 would have been counted both as *1*.

We implemented the *detectGroups* function, both in Alg. 14 & 15, using the *k-means* clustering algorithm [275]. The k-means algorithm requires to specify k, the number of clusters. The standard approach to determine k is to run k-means by varying the k values, calculate the Sum of Squared Errors (SSE) for each k, and choose the k beyond which the SSE does not decrease significantly. For the extraction of the individual profiles, the number of clusters, i.e. the number of footprints, is automatically detected for each customer

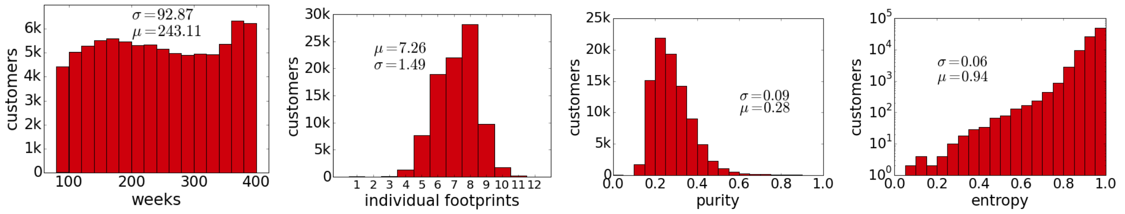


Figure 11.5: Individual distributions: shopping weeks, individual footprints, purity and entropy.

by running the algorithm for $k \in [2, 50]$ and selecting as number of cluster the k which can be considered the "knee" in the SSE curve. We select as *knee* the point on the SSE curve having the maximum distance from the straight line passing through the minimum and the maximum point of the SSE curve. As distance function δ we used the *cosine distance* because unlike the Euclidean or Manhattan, it does not suffer the problem of sparseness. Typically a customer purchases one or two times per week generating very sparse F.

Individual Footprints Analysis

In this section we look at the temporal purchasing profiles extracted employing Alg. 14 for all the customers. The computational time for the extraction of the profiles is about $0.5 - 1.0$ seconds per customer, depending on the number of non empty footprints. Empty footprints are clustered by default in the same group and represented by an empty footprint. For sake of simplicity, if not mentioned, the analysis we report in the following does not consider empty footprints. In Fig. 11.5 *(first)* is reported the distribution of the number of customers having not empty footprints, i.e. the weeks for which at least a purchase was performed. The distribution is quite uniform ranging from 100 to 400[2].

Fig. 11.5 *(second)* shows the distribution of the number of individual footprints. It is a Gaussian shape and with mode ~8. About 80% of the customers must be represented considering more than five footprints. This happens because even though a customer makes purchases on a certain day and time window, she can spend sometimes €50, sometimes €70 and sometimes €90. These behaviors appear in the same time slot but they are "distinct" due to the different nature of the amount spent, and they have a different meaning.

In Fig. 11.5 *(third and fourth)* are illustrated two indicators: purity *(left)* and entropy *(right)*. The *purity* indicates how much the customer is pure in terms of footprints

$$purity = \max_{F \in \mathcal{F}}(sup(F))$$

The *entropy* indicates how much a customer is heterogenous in terms of footprints [262]

$$entropy = -\sum_{F \in \mathcal{F}} sup(F) log(sup(F)^{-1}) / log(|\mathcal{F}|)$$

$sup(F^{(i)}) = |G_i| / |\hat{S}|$ is the relative support of a footprint, i.e. the number of footprints belonging to $F^{(i)}$. The purity distribution is a Gaussian with mode ~0.2, while the entropy has a long-tailed distribution with mean 0.94 and low standard deviation.

[2] Note that in reality 392 not empty footprints is the maximum number of footprints for our dataset since we are considering a period including 392 weeks, and that 400 is driven by the binning of the histogram.

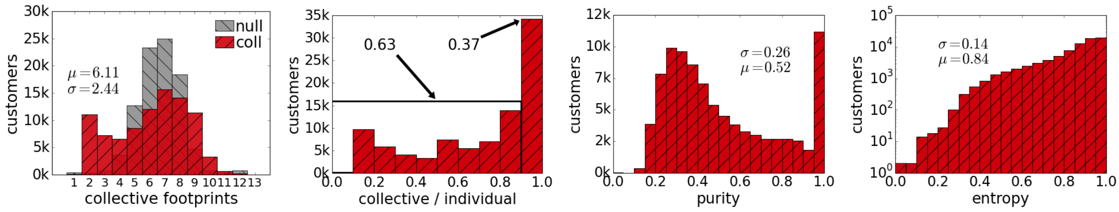


Figure 11.6: Collective distributions: collective footprints, ratio coll-ind, purity and entropy.

Collective Footprints Analysis

In this section we analyze the customers' collective footprints representing the collective perspective of each customer. We extracted the collective perspectives starting from the footprints $\{\mathcal{F}_c\}$ of the customers analyzed by applying in sequence Alg. 15 and Alg. 16. For the clustering in Alg. 15 we ran k-means with k varying from 2 to 150. Fig. 11.8 *(left)* depicts the evolution of the SSE values. We selected $k = 45$ as number of clusters. In Fig. 11.8 *(right)* we can observe the size of each collective group. Group *(29)* contains 500k footprints, the groups with id from *(1)* to *(36)* in the plot are populated by ~50k footprints, the rest of the groups contain about 5k elements. We remark that each customer can have individual footprints represented by different collective footprints, and that each customer can have more than one individual footprint represented by the same collective footprint.

In Fig. 11.7 we report some of the collective temporal purchasing footprints $\mathcal{C}$ of the collective groups $\mathcal{L}$ obtained[3]. The number in the bottom left square indicates how many customers have an individual footprint represented by that collective footprint. Nearly all the collective footprints describe a *one-shop* behavior with the exception of the collective footprints *(29)* and *(38)*. The choice of the day and time window of these *one-shop* purchase behaviors is spread among the various days and time windows. For example, customers having a behavior represented by *(1)* spend about €37 on Sat10-12, those having a behavior represented by *(14)* spend about €49 on Fri10-12, and those represented by *(4)* spend about €55 on Fri16-18. As anticipated by the M-shape in Section 11.1.2, the two time windows mostly used by the customers are 10-12 and 16-18. However, there are also some collective footprints in "unusual" time windows characterizing a smaller number of customers, e.g. collective footprints *(39)* and *(40)*.

In Fig. 11.7 we can notice that shopping behavior for the same day and time window is captured by different collective footprints. Examples of this are collective footprints *(1)* and *(2)* both acting on Sat10-12, and collective footprints *(19)* and *(12)* both acting on Fri16-18. Collective footprints *(2)* and *(12)* have a typical expenditure of €88 and €143: they contain individual footprints for fewer customers and enhance a spending behavior higher than those of *(1)* and *(19)* respectively. In general, observing all the collective footprints, we can classify these one-shop spending behaviors in three classes according to level of amount spent in their peak. We name *normal* spending behavior the collective footprints lower than €50, *high* spending behavior the collective footprints between €50 and €100, and *big* spending behavior the collective footprints with the peak higher than €100.

Collective footprint *(29)* captures occasional shopping sessions where a maximum of €3 is spent. There is not a precise day nor a precise time window but 87% of the customers have this behavior. This indicates that even though each customer has one of the *one-shop* behaviors, in some occasions she makes purchases employing an *occasional* spending

[3] All the collective footprints can be found at https://goo.gl/i7rRBZ.

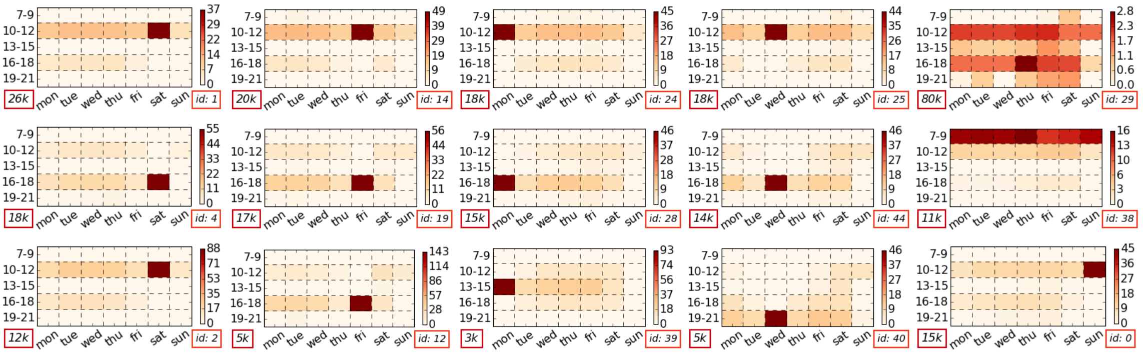


Figure 11.7: Collective footprints: the darker the color, the higher the total amount spent (in €).

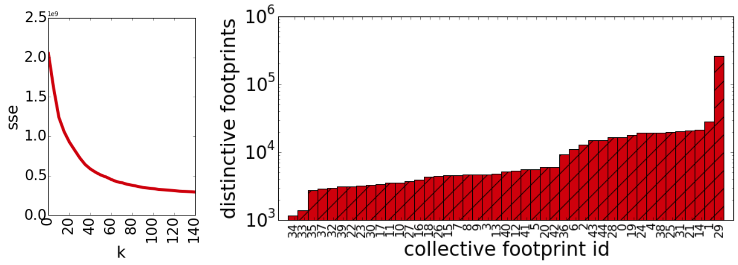


Figure 11.8: SSE varying $k \in [2, 150]$ *(left)*, clusters sizes *(right)*.

behavior. In practice, each customer sometimes occasionally purchases without following a fixed schema when she goes to the shop, and she buys only a few products she needs in that moment. Collective footprint *(38)* captures the behavior of customers that every very early morning (7-9) of the week make a purchase spending at most €16. We name this behavior *daily* spending behavior. The customers having this behavior can be retirees who go to the shopping center every morning to buy only what they need for the day, or workers going to the supermarket before work for buying their lunch.

Regular and Changing Customers. By analyzing the same indicators, we observed in the previous section, we discovered that when using the collective perspective customers can be identified as *regular* or *changing*. In Fig. 11.6 *(first)* is reported the distribution of the number of customer collective footprints $|\mathcal{C}_c|$ of each customer. The already discussed phenomenon that two individual footprints F and F' can be represented by the same customer collective footprint C affects in a not negligible way a consistent subset of the customers. However, instead of generating the same Gaussian distribution observed in Fig.11.5 *(second)* with an increased skewness, we observe the bi-modal distribution shown in Fig. 11.6 *(first)*. This phenomenon recalls the explorer-returner phenomenon observed in mobility behavior [227]. In the same figure we report the *null-distribution* representing the number of collective footprints assigned using a *null model*: for each customer each individual footprint is assigned randomly to a collective footprint. This distribution is Gaussian and its mode is ∼7. Since the two distributions are very different we can state that the result observed is not random. The U-shape of the bi-modal distribution highlights two set of customers: *regular* customers are represented by a limited set of behaviors, less than 5 collective footprints, while *changing* customers must be represented considering a higher number of behaviors, more than 4 collective footprints. Using 4 as threshold we have a 27%-83% partitioning. Thus, *regular* customers are more predictable than *changing* customers since they can adopt a smaller range of characterizing behaviors. Indeed, if we

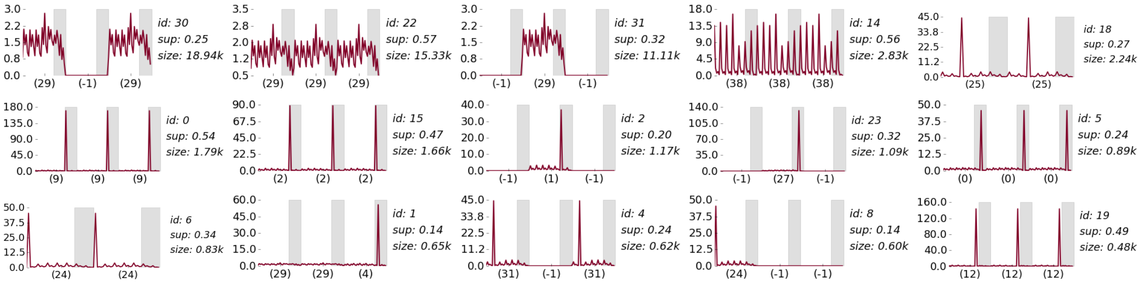


Figure 11.9: Selection of representatives regular sub-sequences of the medoids of the collective clustering. Gray rectangles highlight the weekends.

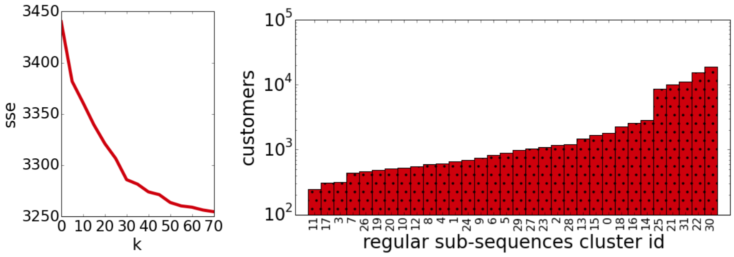


Figure 11.10: SSE varying $k \in [2, 80]$ *(left)*, and clusters sizes *(right)*.

consider only the amount spent, on average, the standard deviation of the amount spent by *regular* customer is 7.61, while it is 32.38 for a *changing* customer.

Fig. 11.6 *(second)* illustrates the distribution of the ratio between the number of the customer collective footprints and the number of individual footprints, i.e. $|\mathcal{C}_c|/|\mathcal{F}_c|$. For 37% customers each individual footprint belongs to a different collective group, for the rest the collective perspective changes the personal definition of behavior. This confirms that, the perception of the temporal purchasing behavior of a customer obtained only observing her own purchases differs from that one we get observing also the other customers.

It is interesting to notice how the distributions of purity and entropy change as consequence of these phenomena. For the purity we can observe a novel group of more than 10k pure customers, and the decreasing of the skewness of the curve. The average purity for a *regular* customer is 0.94, while it is just 0.19 for a *changing* customer. A similar effect is detected w.r.t. the entropy. The entropy distribution for customers' collective footprints is "less long-tailed" than the entropy distribution for individual footprints. Once again, the average entropy for a *regular* customer is 0.65 while it is 0.91 for a *changing* customer. This confirms the higher unpredictability of changing customers. Finally, it is worth to notice the growth of the standard deviation σ for both measures. It reports the improved variability of these measures due to the regular-changing diversification.

Collective Sequence Analysis

While the analysis of the collective footprints can reveal a customers segmentation w.r.t. their temporal purchasing behavior, the analysis of the regular sub-sequences can unveil also a partitioning which describes for each group the order in which the most meaningful collective footprints are repeated. To this end, we cluster and analyze the choices of the regular sub-sequences $\mathcal{R}_c = \{(R_1, w_1), \dots, (R_n, s_n)\}$. Thus, for each customer c we consider her regular sub-sequences and their support as statistical unit for the clustering.

Since given a set of customers represented as set of regular sub-sequences $\{\mathcal{R}_c\}$ is not possible to define a centroid, we segmented the customers using *k-medoids* [164] instead of k-means. We ran k-medoids varying k from 2 to 80. In Fig. 11.10 *(left)* we can observe the SSE values: we selected $k = 33$ as number of clusters. Fig. 11.10 *(right)* shows the size of each cluster, i.e., the number of customers. Clusters *25*, *21*, *31*, *22* and *30* contain more than $2,000$ customers, while all the others contain from 500 to $1,000$ customers.

To analyze the customers segmentation we show in Fig. 11.9 some sub-sequences of different medoids. For each medoid we report the sub-sequence with the higher trade-off between length and support. We highlight that the sub-sequences shown in Fig. 11.9 are not expressing the fact that the customers belonging to that cluster always behave in that way, but they are only describing one of their most common behavior, i.e. their *regularities*.

Clusters 30, 22 and 31 are represented by different permutation of the collective footprints *(29)* and *(-1)*. We indicate with *(-1)* the collective footprint capturing the no-shopping behavior. Their most frequent succession of weeks consists of going to the shop without a very regular pattern for buying only the products they need at that moment. Then sometimes, without a fixed schema, they adopt one of the *one-shop* behaviors. These customers could fall into the category of *casual* customers defined in [129]. Indeed they are quite unpredictable in their *occasional* spending behavior. However, through our approach we discovered differences among those customers looking outside the week unit: the first group has a Yes-No-Yes (Y-N-Y) sequence, the second one buys every week (Y-Y-Y), while the last one is characterized by a N-Y-N sequence. Most of the other clusters are characterized by a repetition of the same collective footprint in the sub-sequences, e.g. clusters 14, 18, 0, 15, 5, 6 and 19. It seems that customers belonging to these clusters have their preferred time to shop and they need to shop in that particular moment. This behavior is probably driven by their weekly time table. However, the fact that there are not no-shopping behavior separating these *one-shop* behaviors is a signal that they consume all the products bought and they need to shop every week. Cluster 14 reveals that also the *daily* spenders repeat regularly their behavior through the weeks. Finally, clusters 2 and 23 capture two different repetitions of *one-shop* behavior following a N-Y-N schema, i.e. these customers depletes her storage in the first week, go to shopping in the second week (with different level of spending between 2 and 3), and consumes the novel supplies in the third week. Cluster 4 is complementary to cluster 2. It is interesting to analyze also clusters 1 and 8. They are specular each other. The first pattern reveals that customers of cluster 1 do not purchase for two weeks and then, on Saturday morning of the third week they spent about €60. On the other hand, customers of cluster 8 have a shopping session on Monday morning with €45, and then they do not need to purchases for two weeks.

11.1.3 Conclusion

By adopting a Personal Data Model designed for temporal shopping behavior, we have investigated the regularities characterizing the temporal purchasing profile of retail customers. Then, we have made the profiles comparable among different customers of the PDE by providing the collective perspectives of the individual profiles. These collective perspective have enabled the analysis and the segmentation of the customers considered. Our case study revealed that for most of the customers the vision of the individual profile is different from its collective perspective and that customers can be classified into *regular* and *changing* according to the number of behaviors needed to describe them.

11.2 Using Retail Market Data to Nowcast Well-Being

Objectively estimating a country's prosperity is a fundamental task for modern society. One such test is the estimation of the Gross Domestic Product, or GDP. GDP is defined as the market value of all officially recognized final goods and services produced within a country in a given period of time. The idea of GDP is to capture the average prosperity that is accessible to people living in a specific region. No prosperity test is perfect, so it comes as no surprise to reveal that GDP is not perfect either. GDP has been harshly criticised for several reasons [85]. First: GDP is not easy to be estimated. It takes time to evaluate the values of produced goods and services, as to evaluate them they first have to be produced and consumed. Second: GDP does not accurately capture the well-being. For instance income inequality skews the richness distribution, making the per capita GDP uninteresting, because it does not describe the majority of the population any more. Moreover, arguably it is not possible to quantify well-being just with the money in someone's pocket: she might have dreams, aspirations and sophisticated needs that bear little to no correlation with the status of her wallet. The critiques to GDP we mentioned have resulted in the proliferation of alternative well-being indicators. We mention the Index of Sustainable Economic Welfare (ISEW), the Genuine Progress Indicator (GPI) [178] and the Human Development Index (HDI)[4]. A more in depth review about well-being alternatives is provided in [144]. These indicators are designed to correct some shortcomings of GDP, namely incorporating sustainability and social cost. However, they are still affected by long delays between measurements and evaluation. They are also affected by other criticisms: for instance, GPI includes a list of adjustment items that is considered inconsistent and somewhat arbitrary. Corrections have been developed [179], but so far there is no final reason to prefer them to GDP and thus we decide to adhere to the standard and we consider only the GDP measure, and we remark that no alternative has addressed the two mentioned issues of GDP in a universally recognized satisfactory way.

Employing data mining tools to nowcast GDP is a promising field of research especially to resolve the delay issues of GDP. Our proposal is to nowcast the GDP level using retail data based on the collaborative flows of personal indicators that can emerge from the Personal Data Ecosystem [130]. Indeed, the approach we follow comes from a recent branch of research that considers markets as self-organizing complex systems. In particular, we define two indicators of the Personal Data Model as measures of product and customer sophistication. The proposed measures are the average sophistication of the satisfiable personal needs of a population. We are able to estimate such personal measure by connecting products sold in the country to the customers buying them in significant quantities, generating a customer-product bipartite network. The sophistication measure is created by recursively correcting the degree of each customer in the network. Customers are sophisticated if they purchase sophisticated products, and products are sophisticated if they are bought by sophisticated customers. Once this recursive correction converges, the aggregated sophistication level of the network is our well-being estimation. In other words, we produce personal indicators of economic sophistication and we transform an aggregation of them into a data-driven collective indicator of well-being.

[4] http://hdr.undp.org/en/statistics/hdi

The approach we follow was first proposed in [142], where the authors model the global export market as a bipartite network, connecting the countries with the products they export. This usage of complex networks has been replicated both at the macro economy level [52] and at the micro level of retail [61]. Also in [235] was showed the power of measures calculated following the same approach to explain the distance traveled by customers to buy the products they need [236]. We borrow these indicators to tackle the problem of nowcasting GDP. An alternative methodology uses electronic payment data [106]. However, in this case the only issue addressed is the timing issue, but no attempt is made into making the measure more representative of the satisfaction of people's needs.

The average sophistication of the personal satisfiable needs addresses the two issues of GDP we discussed. First, it shows a high correlation with the GDP of the country, when shifting the GDP by two quarters. The average sophistication of the bipartite network is an effective nowcasting of the GDP, making it a promising predictor of the GDP value the statistical office will release after six months. Second, our measure is by design an estimation of the sophistication of the needs satisfied by the population.

11.2.1 Dataset

To reach our goal we employed the *Coop* dataset described in Section 6.2. As time granularity for our observation period we choose to use a quarterly aggregation. This because we want to compare our results with GDP, and GDP assumes a better relevance in a quarterly aggregation. For each quarter, we have $\sim 500k$ active customers. Since our objective is to establish a correlation between the supermarket data and the GDP of Italy, we need a reliable data source for GDP. We rely on the Italian National Bureau of Statistic ISTAT. ISTAT publishes quarterly reports about the status of the Italian country under several aspects, including the official GDP estimation. ISTAT is a public organization and its estimates are the official data used by the Italian central government. We downloaded the GDP data from the ISTAT website[5]. As can be observed in Fig. 6.1 (right) the shop distribution is not homogeneous: shops are located in a few Italian regions. Therefore, the coverage of these regions is much more significant. Our analysis is performed on national GDP data, because regional GDP data is disclosed only with a yearly aggregation. However, the correlation between national GDP and the aggregated GDP of the observed regions Tuscany, Lazio and Campania during our observation period is 0.95 ($p < 0.001$). This is because Italy has a high variation on the North-South axis, which we cover, while the West-East variation, which we cannot cover, is very low.

Seasonality

Both GDP and the behavior of customers in the retail market are affected by seasonality. Different periods of the year are associated with different economic activities. This is particularly true for Italy in some instances: during the month of August, Italian productive activities come to an almost complete halt, and the country hosts its peak tourist population. The number and variety of products available in the supermarket fluctuate too, with more fruit and vegetables available in different months, or with Christmas season and subsequent sale shocks. A number of techniques have been developed to deal with seasonal changes in GDP. One of the most popular seasonal adjustments is done through

[5] `http://dati.istat.it/Index.aspx?lang=en&themetreeid=91`, date of last access: Sept 23rd, 2015

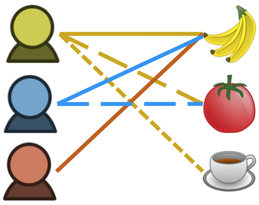


Figure 11.11: The bipartite network connecting customers to the products they buy.

the X-13-Arima method, developed by the U.S. Census Bureau [213]. However, we are unable to use this methodology because it requires an observation period longer than the one we have available for this analysis, and because, the methodologies present in literature are all fine-tuned to specific phenomena that are not comparable to the shopping patterns we are observing. Thus we cannot apply them to our sophistication timelines. Given that we are not able to make a seasonal adjustment for the sophistication, we chose to not seasonally adjust GDP too. We acknowledge this as a limitation of our study and we leave the development of a seasonal adjustment for sophistication as a future work.

11.2.2 Method

The sophistication indicator is used to objectively quantify the sophistication level of the needs of the customers buying products. The sophistication index is introduced in [236], which is an adaptation from [52], necessary to scale up to large datasets. We briefly report here how to compute the customer sophistication index which would be part of the Personal Data Model. The starting point is a matrix with customers on rows and products on the columns. This matrix is generated for each quarter of each year of observation. Each cell contains the number of items purchased by the customer of the product in a given quarter (e.g. Q1 of 2007, Q2 of 2007 and so on). We then have 30 of such matrices. The matrices are very sparse, with an average fill of 1.4% (ranging from 33 to 37 million non zero values). Our aim is to increase the robustness of these structures, by constructing a bipartite network connecting customers exclusively to the subset of products they purchase in significant quantities. Fig. 11.11 provides a simple depiction of the bipartite network.

To filter the edges, we calculate the Revealed Comparative Advantage (RCA, also known as Lift [6]) of each product-customer cell [19], following [226]. Given a product p_i and a customer c_j, the RCA of the couple is defined as follows:

$$RCA(p_i, c_j) = \frac{X(p_i, c_j)}{X(p_*, c_j)} \left(\frac{X(p_i, c_*)}{X(p_*, c_*)} \right)^{-1},$$

where $X(p_i, c_j)$ is the number of p_i bought by c_j, $X(p_*, c_j)$ is the number of products bought by c_j, $X(p_i, c_*)$ is the total number of times p_i has been sold and $X(p_*, c_*)$ is the total number of products sold. RCA takes values from 0 (when $X(p_i, c_j) = 0$, i.e. customer c_j never bought a single instance of product p_i) to $+\infty$. When $RCA(p_i, c_j) = 1$, it means that $X(p_i, c_j)$ is exactly the expected value under the assumption of statistical independence, i.e. the connection between customer c_j and product p_i has the expected weight. If $RCA(p_i, c_j) < 1$ it means that the customer c_j purchased the product p_i less than expected, and vice-versa. Therefore, we keep an edge in the bipartite network iff its corresponding RCA is larger than 1. Note that most edges were already robust. When filtering out the edges, we keep 93% of the original connections.

Differently from [236] that used the traditional economic complexity [226], we use the Cristelli formulation of economic complexity [86]. The two measures are highly correlated, therefore, there is no reason to prefer one measure over the other, and we make the choice of using only one for clarity and readability. Consider the bipartite network $G{=}(C, P, E)$ described by the adjacency matrix $M^{|C|\times|P|}$, where C are customers and P are products. Let c and p be two ranking vectors to indicate how much a C-node is linked to the most linked P-nodes and, similarly, P-nodes to C-nodes. It is expected that the most linked C-nodes connected to nodes with high p_j score have a high value of c_i, while the most linked P-nodes connected to nodes with high c_i score have a high value of p_j. This corresponds to a flow among nodes of the bipartite graph where the rank of a C-node enhances the rank of the P-node to which is connected and vice-versa. Starting from $i \in C$, the unbiased probability of transition from i to any of its linked P-nodes is the inverse of its degree $c_i^{(0)} = \frac{1}{k_i}$, where k_i is the degree of node i. P-nodes have a corresponding probability of $p_j^{(0)} = \frac{1}{k_j}$. Let n be the iteration index. The sophistication is defined as:

$$c_i^{(n)} = \sum_{j=1}^{|P|} \frac{1}{k_j} M_{ij} p_j^{(n-1)} \forall i \quad p_j^{(n)} = \sum_{i=1}^{|C|} \frac{1}{k_i} M_{ij} c_i^{(n-1)} \forall j$$

These rules can be rewritten as a matrix-vector multiplication

$$c = \bar{M} p \quad p = \bar{M}^T c$$

where $\bar{M}$ is the weighted adjacency matrix. So, like previously we have

$$c^{(n)} = \bar{M}\bar{M}^T c^{(n-1)} \quad p^{(n)} = \bar{M}^T \bar{M} p^{(n-1)}$$
$$c^{(n)} = \mathcal{C} c^{(n-1)} \quad p^{(n)} = \mathcal{P} p^{(n-1)}$$

where $\mathcal{C}^{(|C|\times|C|)}{=}\bar{M}\bar{M}^T$ and $\mathcal{P}^{(|P|\times|P|)}{=}\bar{M}^T\bar{M}$ are related to $x^{(n)}{=}Ax^{(n-1)}$. This makes sophistication solvable using the power iteration method (and it is proof of convergence). Note that this procedure is equivalent to the HITS ranking algorithm, as proved in [126].

At the end of our procedure, we have a value of customer and product sophistication for each customer for each quarter. For the rest of the section we focus on customer sophistication for space reasons, and because, in line with the Personal Data Analytics approach, the customer sophistication is, similarly to BRE and STRE, an index part of the Personal Data Model. Each customer is associated with a timeline of 30 different sophistications. The overall sophistication is normalized to take values between 0 and 1. Fig. 11.12 shows the distribution of the customer sophistication per quarter and per year. We chose to aggregate the visualization by quarter because the same quarters are similar across years but different within years, due to seasonal effects. To prove the quality of our sophistication measure in capturing need sophistication, we report in Tab. 11.2 a list of the top and bottom sophisticated products, calculated aggregating data from all customers. Top sophisticated products are non daily needed products, while the least complex products are mostly food items. Being data from Italian retails, pasta is the most basic product.

11.2.3 Case Study

In this section we test the relation between the statistical properties of the bipartite networks generated with our methodology and the GDP values. We first show the evolution

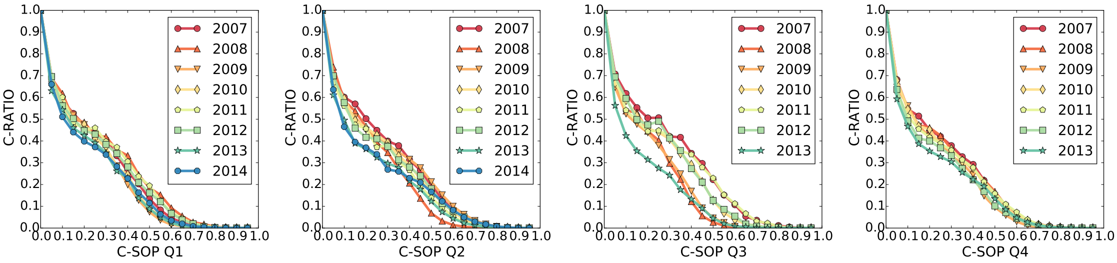


Figure 11.12: The customer sophistication distributions per quarter and per year. Each plot reports the probability (y axis) of a customer to have a given sophistication value (x axis), from quarter 1 to quarter 4 (left to right) for each year.

SOP Rank	Product	SOP Rank	Product
1	Cosmetics	...	...
2	Underwear for men	-5	Fresh Cheese
3	Furniture	-4	Red Meat
4	Multimedia services	-3	Spaghetti
5	Toys	-2	Bananas
...	...	-1	Short Pasta

Table 11.2: The most and least sophisticated products in our dataset.

of aggregated measures of expenditure, number of items, degree and sophistication along our observation period. Then we test the correlation with GDP, with various temporal shifts to highlight the potential predictive power of some of these measures.

We already shown that the sophistication distribution is highly skewed and best represented as an exponential function. The expenditure and the number of items purchased present a skewed distribution among customers: few customers spend high quantities of money and buy many items, many customers spend little quantities of money and buy few items. For this reason, we cannot aggregate these measures using the average over the entire distribution, as it is not well-behaved for skewed values. To select the data we use the inter-quantile range, the measure of spread from the first to the third quantile. In practice, we trim the outliers out of the aggregation and then we compute the average, the Inter-Quartile Mean, or “IQM”. Assuming n sorted values, the IQM is calculated as:

$$x_{\text{IQM}} = \frac{2}{n} \sum_{i=\frac{n}{4}+1}^{\frac{3n}{4}} x_i$$

All the timelines we present have been normalized and the variables take values between 0 and 1, where 0 represents the minimum value observed and 1 the maximum. We report in Tab. 11.3 the abbreviations used in the text and in the captions of the figures

The first relation we discuss is between GDP and the most basic customer variables. Fig. 11.13 depicts the relation between GDP and the IQM expenditure (left), and GDP and IQM of the number of items purchased (right). Besides the obvious seasonal fluctuation, we can see that the two measures are failing to capture the overall GDP dynamics. GDP has an obvious downward trend, due to the fact that our observation window spans across the global financial crisis, which hit Italy starting from the first quarter of 2009. However, the expenditure in the observed supermarket has not been affected at all. Also the number

Abbreviation	Description
IQM	Inter-Quartile Mean.
GDP	Gross Domestic Product.
EXP	IQM of the total expenditure per customer.
PUR	IQM of the total number of items purchased per customer.
C-DEG	IQM of the number of products purchased in significant quantities (i.e. the bipartite network degree) per customer.
P-DEG	IQM of the number of customers purchasing the product in significant quantities (i.e. the bipartite network degree).
C-SOP	IQM of the sophistication per customer.
P-SOP	IQM of the sophistication per product.

Table 11.3: The abbreviations for the measures used in the experiment section.

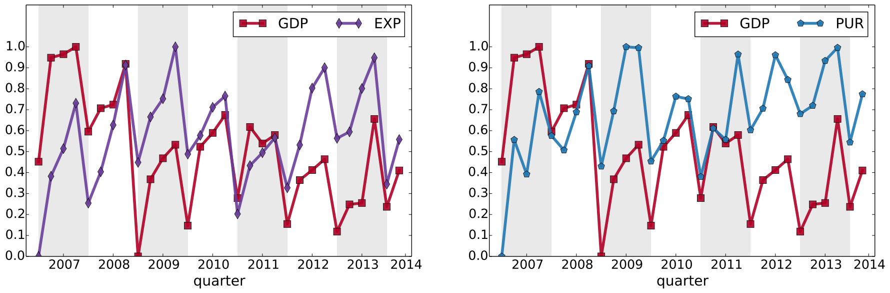


Figure 11.13: GDP and IQM customer expenditure (left) and IQM items purchased (right).

of items has not been affected. If we calculate the corresponding correlations, we notice a negative relationship which, however, fails to pass a stringent null hypothesis test (p>0.01).

Turning to our sophistication measure, Fig. 11.14 depicts the relation between GDP and our complex measures of sophistication. On the left we have the measure of customer sophistication we discussed so far. We can see that the alignment is indeed not perfect. However, averaging out the seasonal fluctuation, customer sophistication captures the overall downward trend of GDP. The financial crisis effect was not only a macroeconomic problem, it also affected the sophistication of the satisfiable needs of the population.

Note that, again, we have a negative correlation. This means that, as GDP shrinks, customers become more sophisticated. This is because the needs that once were classified as basic are not basic any more, hence the rise in sophistication of the population. Differently from before, the correlation is actually statistically significant ($p < 0.01$).

We also report on the left the companion sophistication measure: since we can define the customer sophistication as the average sophistication of the products they purchase, we can also define a product sophistication as the average sophistication of the customers purchasing them. Fig. 11.14 (right) shows the reason why we do not focus on product sophistication: the overall trend for product sophistication tends to be the opposite of the customer sophistication. This anti-correlation seems to imply that, as the customers struggle in satisfying their needs, the once top-sophisticated products are not purchased any more, lowering the overall product sophistication index. However, this is only one of many possible interpretations and we need further investigation in future works.

We sum up the correlation tests performed in Tab. 11.4. We report the correlation values for all variables. We test different shift values, where the GDP timeline is shifted of a given number of quarters with respect to the tested measure. When shift = -1, it means that we align the GDP with the previous quarter of the measure (e.g. GDP Q4-08 aligned

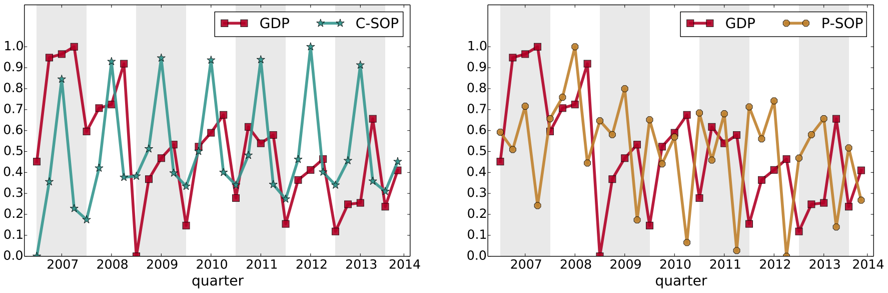


Figure 11.14: GDP and IQM customer (left) and product (right) sophistication.

Shift Measure	-3	-2	-1	0	1	2
EXP	-0.29302	-0.49830	-0.53078*	0.23976	-0.27619	-0.37073
PUR	-0.27091	-0.49836*	-0.53046**	0.18638	-0.30909	-0.32432
C-DEG	0.24624	0.39808	-0.55479*	0.13727	0.08191	0.36001
P-DEG	-0.12409	-0.26289	-0.57657**	0.30255	-0.22198	-0.28325
C-SOP	-0.32728	**-0.67007*****	0.23261	0.09251	-0.15844	-0.58773**
P-SOP	-0.02675	-0.12916	0.60974**	-0.18587	0.15342	-0.03843

$^{*}p < 0.1$, $^{**}p < 0.05$, $^{***}p < 0.01$

Table 11.4: The correlations of all the used measures with GDP at different shift values. We highlight the statistically significant correlations.

with measure's Q3-08). We also report the significance levels of all correlations. Note that all p-values are being corrected for the multiple hypothesis test. When considering several hypotheses, as we are doing here, the problem of multiplicity arises: the more hypotheses we check, the higher the probability of a false positive. To correct this issue, we apply a Holm-Bonferroni correction. The Holm-Bonferroni method is an approach that controls the family-wise error rate (the probability of witnessing one or more false positive) by adjusting the rejection criteria of each of the individual hypotheses [149]. Once we adjust the p-values, we obtain the significance levels reported in the table. Only one correlation passes the Holm-Bonferroni test for significance at $p < 0.01$ and it is exactly the one involving the customer sophistication with shift equal to -2. This correlation is highlighted in bold in Table 11.4, and it represents the main result of this analysis.

Note that in the table we also report the correlation values using the IQM for the customer and product degree measures, of which we have not shown the timelines, due to space constraints. We include them because, as we discussed previously, our sophistication measures are corrected degree measures. If the degree measures were able to capture the same correlation with GDP there would be no need for our more complex measures. Since the degree measures do not pass the Holm-Bonferroni test we can conclude that the sophistication measures are necessary to achieve our results.

We finally provide a visual representation of the customer and product sophistication correlations with GDP at different shift levels in Fig. 11.15. The figure highlights the different time frames in which the two measures show their predictive power over GDP. The customer sophistication has its peak at shift equal to -2. The cyclic nature of the data implies also a strong, albeit not significant, correlation when the shift is equal to 2. Instead, the product sophistication obtains its highest correlation with GDP with shift equal to -1. This might still be useful in some cases, as the GDP for a quarter is usually released by the statistical office with some weeks of delay.

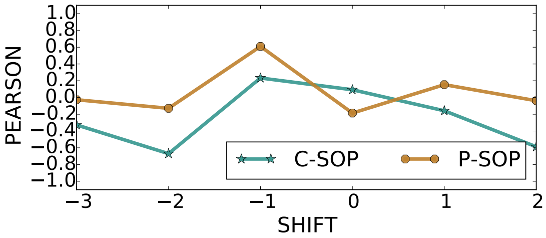


Figure 11.15: Correlation between average customer sophistication and GDP with different shifts.

11.2.4 Conclusion

We have introduced a personal indicator in the Personal Data Model and we have aggregated them at collective level in the Personal Data Ecosystem for having a fast and reliable test for estimating the well-being of a population. Traditionally, this is achieved with GDP. However, GDP is affected by several issues. By using retail information, we have been able to estimate the overall sophistication of the needs satisfied by a population by constructing and analyzing a customer-product bipartite network. We have shown that our customer sophistication measure is a promising predictor of the future GDP value, anticipating it by six months. It is also a measure less linked with the amount of richness around a person, and it focuses more on the needs this person is able to satisfy.

Chapter 12

Deployment of Personal Data Analytics

In the last chapter we present some first attempts of real adoption of Personal Data Analytics. In the previous sections, the Personal Data Analytics approach has been formulated at theoretical level, and experiments and case studies have been presented supposing that each person has her own PDS with their own Personal Data Models. However, in practice, the datasets analyzed were not distributed but centralized and treated separately for each person, and the experiments were a sort of "what-if" analysis with respect to certain performance indicators to evaluate the goodness of the method, model, or service proposed.

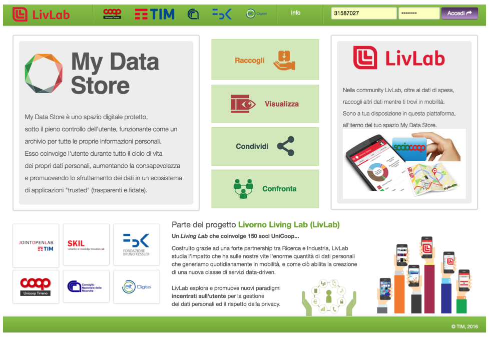
As a concrete test-bed to experiment and to validate the ideas developed in this thesis, in April 2015 we founded the *LivLab*, a living laboratory. The partners of this laboratory are the KDDLAD of UNIPI and CNR in collaboration with Tim and Unicoop Tirreno, (i.e., the largest Italian telecom operator and one of the largest retail distribution companies in Italy)[1]. The LivLab aims at creating a participatory eco-system for setting-up experiments on a trusted user-centric platform, and based on personal mobility and purchasing data.

Moreover, we present a case study of the PETRA project as an additional application of Personal Data Analytics for improving the benefits of the collectivity. In such scenario, we consider together the *public* transportation system of a city with the *private* one formed by the drivers eligible for carpooling according to their PDS, and we evaluated the travel time reduction of the system which uses also the private cars.

12.1 LivLab: The Adoption of the Personal Data Store

The main goals of LivLab are: *(i)* creating and managing a community of volunteers, *(ii)* realizing services built on top of users' data, and *(iii)* increasing users' awareness about their daily activities. The laboratory is involving customers in the province of Leghorn to collect GPS signals to track movements, and purchasing data in their PDS, and to offer them analytical services exploiting these data. The LivLab currently is composed of about 100 active users. Through a web interface or a mobile app, each customer can login to her home page (Fig. 12.1 (left)). The simplest service implemented is the *historical data visualization* where the customer can navigate and consult her own data (purchases, GPS, calls, etc.) together with basic information (see Fig. 12.1 (right)).

[1] http://goo.gl/44JnRk, http://goo.gl/OD02nl



My Data Store

My Data Store è uno spazio digitale protetto, sotto il pieno controllo dell'utente, funzionante come un archivio per tutte le proprie informazioni personali. Esso coinvolge l'utente durante tutto il ciclo di vita dei propri dati personali, aumentando la consapevolezza e promuovendo lo sfruttamento dei dati in un ecosistema di applicazioni "trusted" (trasparenti e fidate).

LivLab

Nella community LivLab, oltre ai dati di spesa, raccogli altri dati mentre ti trovi in mobilità. Sono a tua disposizione in questa piattaforma, all'iterno del tuo spazio My Data Store.

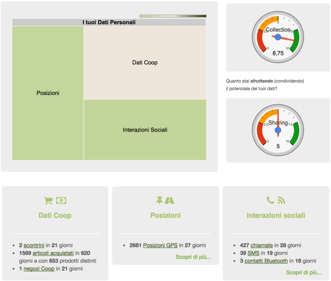
Parte del progetto Livorno Living Lab (LivLab)

Un *Living Lab* che coinvolge 150 soci UniCoop...

Costruito grazie ad una forte partnership tra Ricerca e Industria, LivLab studia l'impatto che ha sulle nostre vite l'enorme quantità di dati personali che generiamo quotidianamente in mobilità, e come ciò abilita la creazione di una nuova classe di servizi data-driven.

LivLab esplora e promuove nuovi paradigmi **incentrati sull'utente** per la gestione dei dati personali ed il rispetto della privacy.

Figure 12.1: PDS LivLab home page.



Quanto stai ***sfruttando*** (condividendo) il potenziale dei tuoi dati?

- **2** scontrini in **21** giorni
- **1569** articoli acquistati in **620** giorni e con **653** prodotti distinti
- **1** negozi Coop in **21** giorni

- **2681** Posizioni GPS in **27** giorni

Scopri di più...

- **427** chiamate in **28** giorni
- **39** SMS in **19** giorni
- **3** contatti Bluetooth in **18** giorni

Scopri di più...

Figure 12.2: Personal statistics.

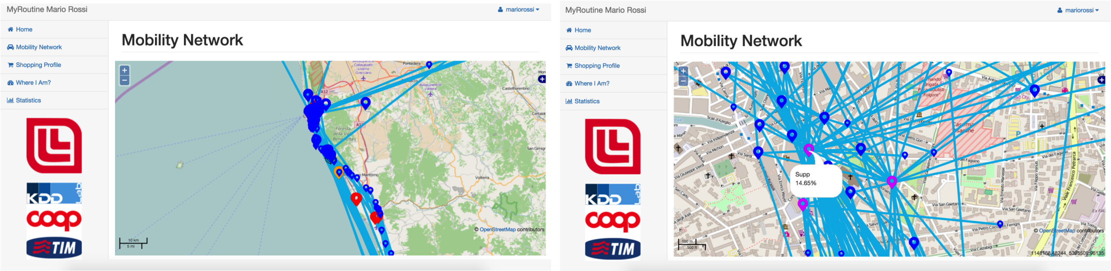
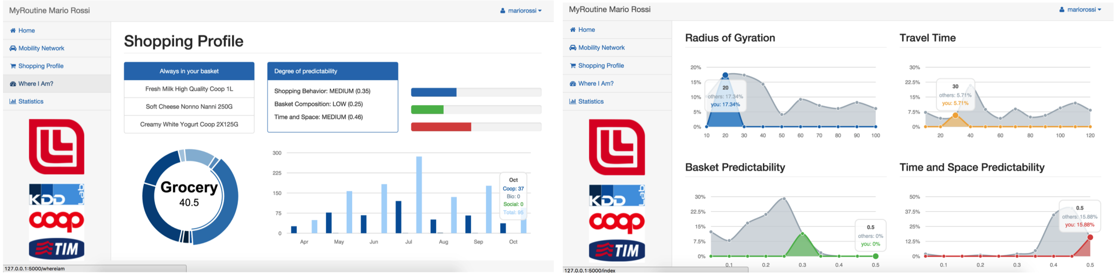


Figure 12.3: Visualization of individual mobility network.



Always in your basket
Fresh Milk High Quality Coop 1L
Soft Cheese Nonno Nanni 250G
Creamy White Yogurt Coop 2X125G

Degree of predictability

Shopping Behavior: MEDIUM (0.35)
Basket Composition: LOW (0.25)
Time and Space: MEDIUM (0.46)

Figure 12.4: Analysis of purchases and "Where I Am?", i.e. comparison with the others.

Through Personal Data Analytics, we have applied some of the techniques and models described in this thesis in order to improve the self-awareness of the customers. Some examples of services that we have implemented are: *(i)* visualization on a map of the individual mobility network (see Section 8.3) with the possibility to filter among frequent/unfrequent locations (see Fig. 12.3), *(ii)* self-analysis of purchasing behavior including the BRE and STRE indicators presented in Section 8.1 together with the most frequent shopping pattern extracted with the *txmeans* algorithm (Section 7.2) that are used to recommend the item for the next shopping list, and statistics about the favorite market category and biological products (see Fig. 12.4 (left)), and *(iii)* the "Where I Am?" service, i.e., a comparison of the personal models and indicators against those of the collectivity, that is the behavior of other users in the LivLab Personal Data Ecosystem that want to understand how they behave with respect to the mass (see Fig. 12.4 (right)).

Therefore, by adopting the LivLab PDS, a user can gain the control on her data together with additional information extracted by using the Personal Data Analytics approach This knowledge can suggest to the user the adoption of new and different behaviors able to improve her lifestyle. Moreover, extra advantages are provided by models and data sharing, enabling the user to understand how much is different or similar to the others.

12.2 Integration of Private and Public Means of Transport

The development of smart mobility services is mandatory to build our continuously evolving smart cities. The PErsonal TRansport Advisor (PETRA) EU FP7 project[2], has as main aim the development of an integrated platform to supply urban travelers with smart journeys and activity advisers, on a multi-modal network, while considering uncertainty. It is a first prototype of a real service which requires as core component a Personal Data Ecosystem made of interconnected Personal Data Stores able to share meaningful patterns.

In the following, we briefly describe the architecture of the PETRA platform, and we present how a journey planner considering public services can be boosted with private means of transport in form of knowledge coming from the Personal Mobility Data Models [42]. In particular, we show how by integrating private and individual transport systematic *routines* R_u into a public transit network it is possible to devise better collective advises, measured both in terms of the number of requests satisfied, and in terms of the expected time of arrivals. Besides validating the utility of the multimodal carpooling system origintaed by the union between private and public means of transports, the experiments we show in the following are also part of the validation for the PETRA use case on Rome, where we assess the quality of the advises coming from the innovative integrated platform.

12.2.1 PETRA Journey Planner

Fig. 12.5 shows the diagram of a simplified architecture for PETRA. We describe in the following the main modules used for private and public means of transport integration.

Data Manager. The PETRA project highlights the need to integrate different types of urban data, from unstructured data to real-time information retrieved from city sensors. Handling such large volumes of data requires a tailored and scalable data management platform, from which we highlight the following modules: *(i)* data acquisition, responsible for ingesting the heterogeneous city data and needs to consider the case of streaming data returned from sensors and other city traffic sources; *(ii)* distributed data storage and indexing, providing indexes designed for the different formats of data that can be handled by the system (relational, tabular, and graph data), and also their different types (geospatial, textual, etc); *(iii)* partitioning, distributing the acquired data across the different nodes of the data storage; *(iv)* query and searching, providing a combination of structural query processing and search techniques in order to answer different kinds of queries. The Data Manager (DM) exposes its data to the other PETRA components via a set of APIs. In particular, the Journey Planner (JP) retrieves the required General Transit Feed Specification (GTFS) data from the DMâĂŹs internal stored version by using the APIs.

Mobility Mining. The Mobility Mining module fetches GPS data about private vehicle trajectories from the DM. It uses a data mining process named *mobility profiling* to extract patterns from these traces. This process for each user $u \in U$, takes as input the user's trajectories H_u and returns the set of personal *routines* R_u describing her systematic movements (see Section 8.3). The set of all the collective routines $R_C = \{R_u\}$ can be exploited as "alternative bus route" by the JP. These newly introduced routes can be exploited as an embedded carpooling service, transparently available through the PETRA platform.

[2] http://www.petraproject.eu

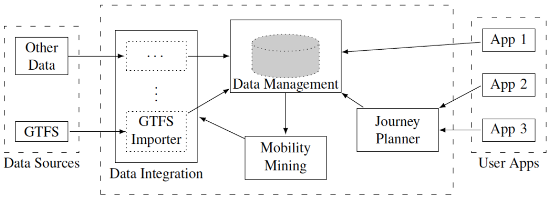


Figure 12.5: Simplified PETRA architecture.

The Multi-Modal Journey Planner Multi-Modal journeys in a city allow to combine, as part of the same trip, multiple transportation modes, such as buses, trams, subways and trains. In the context of our use case we deploy DOCIT [41] that is a multi-modal planner taking into account uncertainties related to the expected arrival time of the different modes of transport available in a given city. DOCIT provides functions such as journey plan computation, plan execution monitoring, and replanning. The DOCIT components that are the most relevant to our scenario are DIJA multi-modal Journey Planner (JP) [43], which is used for the initial planning of journey, and a simulator for plan execution, which is used to monitor the validity of active journey plans. To better perform the planning an simulation task DOCIT requires updated data. To achieve that we created a connection between DOCIT and the DM, and thus deploying DOCIT in RomeâĂŹs use case.

12.2.2 Case Study

The PETRA platform is being deployed by the partner cities of Rome, Venice, and Haifa. However, in the following we analyze a use case for the city of Rome. In the Rome's case study, the PETRA platform, from the traveler's *individual* perspective, provides journey plans from place A to place B. From the *collective* perspective of the Personal Data Ecosystem, this is done by: importing static and real time urban transport data; fusing private routines into the public transport data; computing uncertainty-aware multi-modal advises. In the following we describe the data used, how the import step works, and the results obtained with and without the fusion of private personal routines R_C.

Rome Data. The city of Rome, through Agenzia Mobilità, constantly provides updated open data about its public transport systems. Two main sources of information are offered via its website[3]: *(i)* Rome public transport General Transit Feed Specification (GTFS) data[4], which is a static snapshot of the entire public transport network updated every few weeks and can be downloaded from the website; *(ii)* Rome public transport real-time API that consists of a set of XML-RPC methods[5], which provide updated transport information e.g., expected arrival times. Moreover, Agenzia Mobilità is gathering a large collection of GPS traces from private cars, similar to those described in Section 6.1. These GPS data are used by the mobility mining module to extract the mobility routines.

Importing Rome's Data. Importing Rome's data relies on an ad-hoc data acquisition module (named RDI), that acts as a bridge between the different kinds of data and the internal DM. Given as initial state the Rome public transport GTFS, we can divide the work of the RDI in two sub-tasks: *(i)* the daily update, and *(ii)* the real-time update.

[3] See http://www.agenziamobilita.roma.it/

[4] https://developers.google.com/transit/gtfs/reference

[5] http://www.xmlrpc.com/spec/

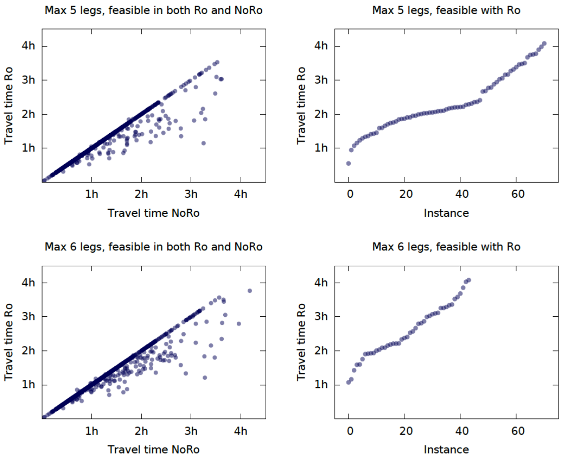
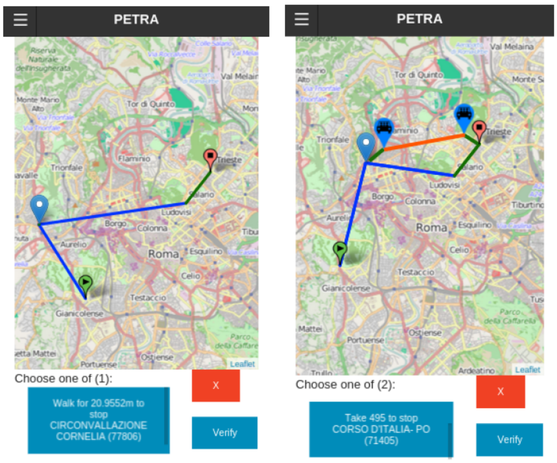


Figure 12.6: Impact of routines on travel time.



Choose one of (1):

Walk for 20.9552m to stop CIRCONVALLAZIONE CORNELIA (77806)

Choose one of (2):

Take 495 to stop CORSO D'ITALIA- PO (71405)

Figure 12.7: The PETRA JP application: *(left)* the original plan, *(right)* the adapted plan.

The *daily update* consists of two steps: discovering bus stops routines and enforcing privacy over these routines. First, the RDI transforms the private car routines into sequences of bus stops and combines them as bus lines: each GPS location is mapped to the closest bus stop within a given radius. Then, the RDI discards any bus stop routines consisting of one stop, or only two bus stops which are closer than a given threshold. In order to guarantee car driversâĂŹ privacy, the RDI checks if an external attacker could exploit the bus stops routines to discover their identity by analyzing their vulnerability against the linking attack model [210]. To avoid this kind of attack the RDI performs a privacy risk analysis following a methodology similar to [28], the result of this method is a probability distribution of the risk of identifying drivers for each routine. If possible, the routines with an identification probability higher than a given acceptable risk are transformed into a safer version by removing some bus stops, otherwise they are deleted. Finally, all the valid bus stop routines are added to the Rome GTFS data and sent to the DM.

In the real-time update, the RDI queries the Rome public transport real-time API every t minutes and checks if there is any update (e.g. buses which have been delayed or cancelled) by comparing expected arrival times on the existing GTFS data with real-time arrivals. Then it converts possible updates into GTFS format, and sends them to the DM.

Impact of Routines in Journey Planning. We ran the planning system in two different settings. In the *NoRo* setting, the planner uses all the public transport data available, but no routines. The *Ro* setting contains both routines and public transport data. In each setting, we solved 2,000 queries (instances) with the origins and destinations generated at random. Instead of using synthetic queries on the city of Rome, we had access to the logs of the official journey planner of Agenzia Mobilità, among which we selected a random sample. In a query, users can set parameters such as the maximum walking time per journey m_w, and the maximum number of legs per journey m_l. We set m_w to 20 minutes, the default planner value. Half of the queries have m_l set to 5, and the other half is for $m_l = 6$. The public transport data we used has 8,896 stops and 391 routes. Each route is served by a number of trips, to a total of 39,422 trips per day. The Rome roadmap has 522,529 nodes and 566,400 links. In the GTFS data, we represent routines with a structure similar to public transport data. Each routine introduces a new route and a new trip. Our test data contains $|R_C| = 15,205$ routines.

Fig. 12.6 illustrates the impact of adding routines as an additional mode. At the left, we compare the travel time in the *Ro* and *NoRo* settings. As expected, in a subset of cases, the travel time is the same. On the other hand, all points located below the main diagonal show instances where routines improve the time. In fact, routines can improve both the travel time and the number of legs per journey. The latter has two advantages. First, it makes a trip more convenient to the traveler, as it reduces the number of interchanges. Secondly, it helps increase the set of feasible instances (i.e., instances for which a solution exists). This is important because user-imposed constraints on m_l and m_w can restrict the set of feasible instances. For example, without using routines, in 29.3% of our queries (instances), it is impossible to complete the journey with at most 20 minutes of walking and at most 5 legs in the journey. Charts at the right in Fig. 12.6 show instances that become feasible after adding routines. When m_l is set to 5, routines are part of the returned plan in 17.5% of the instances. Routines increase the percentage of feasible instances by 7.1%, to a total of 77.8%. In 9.6% of the instances, routines improve the travel time, the average savings per trip being equal to 25.5 minutes. When m_l=6, routines become part of the plans in 22.3% of the instances. They increase the percentage of feasible instances from 84.5 to 88.9%. In 14.3% of the instances, routines improve the travel time, the average improvement amounting to 22.05 minutes per trip.

An Example of Reactive Journey Plan. Consider the following example, a user wants to travel from the bus stop “Zambarelli-Ceres” to “Tagliamento-Chian” at 20:25. Users of the PETRA mobile application can specify the departure and arrival locations, departure time, and additional constraints such as the maximum amount of time they want to spend in the different transport modes. Upon having received the user request the JP computes the plan using the available information and returns a journey plan, as shown in Fig. 12.7 *(left)*. If a delay is detected on the next line of the plan while the user is still on the first bus, the JP automatically calculates a new plan and displays the new choices to the user (shown in Fig. 12.7 *(right)*). In this case, along with an alternative bus line, the user can also select a carpooling option (displayed in orange).

12.2.3 Conclusion

We have presented our results obtained by employing Personal Data Analytics and data mining on the available types of data, i.e., either individual data from private drivers and public data from Agenzia Mobilità, and for this special case of mobility recommendation by running the PETRA platform on the city of Rome for journey planning. Our goal was to assess the impact of adding the results of the mobility mining module into the data feeding the journey planner. Our results show an increased number of planning instances satisfied thanks to the personal mobility routines, along with a reduced average expected travel time. These analyzes proof that the Personal Data Ecosystem and the sharing of personal patterns is a crucial point for the development of advanced services in order to improve the quality of life, to reduce the traffic and the travel times.

Chapter 13

Conclusion

The continuous and unstoppable growth of the digital breadcrumbs that each individual leaves behind while performing her daily activities constitutes an invaluable source of unused knowledge. At the current state of the art there are not models and algorithms specifically designed for personal data, nor services developed by exploiting individual and collective information. In this thesis, we have proposed to extend the idea of Personal Data Store by means of the *Personal Data Analytics* approach that is able to capture the individual's behavior through Personal Data Mining algorithms and Personal Data Models. An additional element consists in the cooperation of the users in the *Personal Data Ecosystem* where individual patterns can be shared for leveraging a collective knowledge in order to enable and improve personal services for providing also collective benefits.

First, we have recalled the basic notions of data mining and user profiling, and we have presented the relevant literature of personal models and individual and collective services. Then, we have accurately discussed the current state of Personal Data Stores, which are the available implementations and what they offer to the user. On top of that, we have defined our vision of Personal Data Analytics, that is a PDS containing Personal Data Models extracted with Personal Data Mining algorithms. In order to leverage the individual awareness with the collective one, we have described the PDS as part of a Personal Data Ecosystem in form of a distributed network of users.

We have addressed the goals of our thesis by realizing the Personal Data Analytics approach and by employing it for developing real services. As first step we have defined innovative parameter-free clustering algorithms for extracting the user profile. These methods, besides being usable on the datasets of different users without requiring parameter tuning, are able to overcome the state-of-the-art competitors both in efficiency and quality of the clusters returned. As second step, we have described Personal Data Models able to capture and measure systematic behaviors for purchasing transactions, mobility data, and listening data. These models enable the analysis of individual features which accurately describe each user, and can be used for the development of a personal dashboard for the user's self-awareness. As third step, in the last part of this thesis, we have deployed services on top of the Personal Data Ecosystem which exploit the models and methods described in the previous part. We have developed individual services for improving personal mobility: a trajectory predictor and a personalized route planner. Yet exploiting the systematic movements of the users in the Personal Data Ecosystem, we have constructed a system providing proactive carpooling suggestions driven by the analysis of habitual paths.

Furthermore, we have tried to reduce the reticence among the users when sharing the car with strangers by considering also some social aspects. Then, we have shown how the application of the Personal Data Ecosystem on shopping transactions can lead to a new level of knowledge with respect to the temporal dimension of shopping, and how a personal measure computed at collective level can be used to nowcast the level of well-being of a society. Finally, we have reported a real case study where a prototype of the PDS proposed in this thesis is adopted by a small set of users.

The results that we have obtained in this thesis in terms of algorithms, models, and services represent an example of how Personal Data Analytics can be useful in the development of the user-centric perspective. Clearly, we do not intend to conclude our study on Personal Data and on the dualism between individual and collective with the contents of this thesis, since many interesting research problems are still open. There are at least three future research directions for the analytical approaches proposed in this thesis.

As consequence of the massive availability of different types of data for each user, the first line of research involves the development of personal multidimensional models. Indeed, besides the mobility and shopping transactional data considered in the LivLab experience described in Chapter 12, we would like to collect for each user also musical listening, tweets, messages, phone calls, credit card transactions, health data, etc. In turn, the extraction of multidimensional models requires ad-hoc algorithms. Therefore, we would like to develop a novel living laboratory where more different types of data and dimensions are considered for each user, and these users are observed for a longer period. This novel laboratory will enable the challenging development of complex models and algorithms to deal with multidimensional data. Therefore, through these models, we could study how and if the multidimensional self-awareness, and the awareness of the collectivity, influence the user's behaviors. Moreover, we would like to provide to the final users, some real services among those theorized in this thesis, e.g. a carpooling service.

The second track of research is related to the adaptation of the models developed with the privacy-by-design paradigm: all the information that a user shares with the collectivity, either raw data or part of the user profile, should be anonymized in such a way the user would not be identifiable through her public data. Moreover, multidimensional data models go over the state-of-the-art with respect to existing privacy preserving method. Thus, is required the development of novel techniques and the adjustment of the existing ones. Another question related to privacy left open by this thesis that should be carefully analyzed in the future is the legality of possession of personal data and patterns and their use in the courts. Is a PDS a fair game to serve as evidence in a trial or will it be treated differently? In the US this relates to the age old legal debate of whether such personally collected data falls under the 4th or 5th amendment of the constitution [73].

Finally, the third track of research involves the study of a technology for the development and deployment of the Personal Data Store and of the Personal Data Ecosystem. Due to its nature, blockchain can be able to regulate the exchange of data and patterns and can also help in guaranteeing a certain level of privacy. Our final goal will be not only to provide the scientific community with methods and models able to automatically deal with personal data during the extraction of personal patterns, but also to offer the basis for a real paradigm for the development of distributed and privacy preserving services, where the core of the systems themselves is the user.

Bibliography

[1] S. Abiteboul, B. André, and D. Kaplan. Managing your digital life. *Communications of the ACM*, 58(5):32–35, 2015.

[2] S. Abiteboul, A. Bonifati, G. Cobéna, I. Manolescu, and T. Milo. Dynamic xml documents with distribution and replication. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 527–538. ACM, 2003.

[3] O. Abul, F. Bonchi, and M. Nanni. Never walk alone: Uncertainty for anonymity in moving objects databases. In *2008 IEEE 24th International Conference on Data Engineering*, pages 376–385. Ieee, 2008.

[4] G. Adomavicius and A. Tuzhilin. Using data mining methods to build customer profiles. *Computer*, (2):74–82, 2001.

[5] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6):734–749, 2005.

[6] R. Agrawal, T. Imieliński, and A. Swami. Mining association rules between sets of items in large databases. In *Acm sigmod record*, volume 22, pages 207–216. ACM, 1993.

[7] I. F. Akyildiz and W. Wang. The predictive user mobility profile framework for wireless multimedia networks. *IEEE/ACM Transactions on Networking (TON)*, 12(6):1021–1035, 2004.

[8] J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.

[9] C.-N. E. Anagnostopoulos and S. Hadjiefthymiades. Intelligent trajectory classification for improved movement prediction. *Systems, Man, and Cybernetics: Systems, IEEE Transactions on*, 44(10):1301–1314, 2014.

[10] T. Anagnostopoulos, C. Anagnostopoulos, and S. Hadjiefthymiades. Mobility prediction based on machine learning. In *2011 IEEE 12th International Conference on Mobile Data Management*, volume 2, pages 27–30. IEEE, 2011.

[11] H. Andersen, M. Andreasen, and P. Jacobsen. The crm handbook–from group to multi-individual. *Norhaven: PricewaterhouseCoopers*, 1999.

[12] C. Anderson. The end of theory. *Wired magazine*, 16(7):16–07, 2008.

[13] G. Andrienko, N. Andrienko, S. Rinzivillo, M. Nanni, D. Pedreschi, and F. Giannotti. Interactive visual clustering of large collections of trajectories. In *Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on*, pages 3–10. IEEE, 2009.

[14] P. Andritsos, P. Tsaparas, R. J. Miller, and K. C. Sevcik. Limbo: Scalable clustering of categorical data. In *International Conference on Extending Database Technology*, pages 123–146. Springer, 2004.

[15] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander. Optics: ordering points to identify the clustering structure. In *ACM Sigmod Record*, volume 28, pages 49–60. ACM, 1999.

[16] M. Armendáriz, J. C. Burguillo, A. Peleteiro-Ramallo, G. Arnould, and D. Khadraoui. Carpooling: A multi-agent simulation in netlogo. In *ECMS*, pages 61–67, 2011.

[17] D. Ashbrook and T. Starner. Using gps to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing*, 7(5):275–286, 2003.

[18] A. Bachmann, C. Borgelt, and G. Gidófalvi. Incremental frequent route based trajectory prediction. In *Proceedings of the Sixth ACM SIGSPATIAL International Workshop on Computational Transportation Science*, page 49. ACM, 2013.

[19] B. Balassa. Trade liberalisation and âĂIJrevealedâĂİ comparative advantage1. *The Manchester School*, 33(2):99–123, 1965.

[20] P. Ball. *Why society is a complex matter*. Springer, 2012.

[21] Z. Bao, Y. Zeng, and Y. Tay. sonlp: Social network link prediction by principal component regression. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 364–371. ACM, 2013.

[22] A.-L. Barabasi. The origin of bursts and heavy tails in human dynamics. *Nature*, 435(7039):207–211, 2005.

[23] A.-L. Barabási. *Bursts: the hidden patterns behind everything we do, from your e-mail to bloody crusades*. Penguin, 2010.

[24] D. Barbará, Y. Li, and J. Couto. Coolcat: an entropy-based algorithm for categorical clustering. In *Proceedings of the eleventh international conference on Information and knowledge management*, pages 582–589. ACM, 2002.

[25] V. Barnett and T. Lewis. *Outliers in statistical data*, volume 3. 1994.

[26] D. Barth, S. Bellahsene, and L. Kloul. Mobility prediction using mobile user profiles. In *Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), 2011 IEEE 19th International Symposium on*, pages 286–294. IEEE, 2011.

[27] D. Barth, S. Bellahsene, and L. Kloul. Combining local and global profiles for mobility prediction in lte femtocells. In *Proceedings of the 15th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems*, pages 333–342. ACM, 2012.

[28] A. Basu, A. Monreale, J. C. Corena, F. Giannotti, D. Pedreschi, S. Kiyomoto, Y. Miyake, T. Yanagihara, and R. Trasarti. A privacy risk model for trajectory data. In *IFIP International Conference on Trust Management*, pages 125–140. Springer, 2014.

[29] G. Bell. A personal digital store. *Communications of the ACM*, 44(1):86–91, 2001.

[30] T. Bellemans, S. Bothe, S. Cho, F. Giannotti, D. Janssens, L. Knapen, C. Körner, M. May, M. Nanni, D. Pedreschi, et al. An agent-based model to evaluate carpooling at large manufacturing plants. *Procedia Computer Science*, 10:1221–1227, 2012.

[31] M. Berlingerio, V. Bicer, A. Botea, S. Braghin, N. Lopes, R. Guidotti, and F. Pratesi. Managing travels with petra: the rome use case. 2015.

[32] M. Berlingerio, F. Calabrese, G. Di Lorenzo, X. Dong, Y. Gkoufas, and D. Mavroeidis. Safercity: a system for detecting and analyzing incidents from social media. In *2013 IEEE 13th International Conference on Data Mining Workshops*, pages 1077–1080. IEEE, 2013.

[33] M. Berlingerio, F. Calabrese, G. Di Lorenzo, R. Nair, F. Pinelli, and M. L. Sbodio. Allaboard: a system for exploring urban mobility and optimizing public transport using cellphone data. In *Machine learning and knowledge discovery in databases*, pages 663–666. Springer, 2013.

[34] M. J. Berry and G. S. Linoff. *Data mining techniques: for marketing, sales, and customer relationship management.* John Wiley & Sons, 2004.

[35] C.-E. Bichot. Co-clustering documents and words by minimizing the normalized cut objective function. *Journal of Mathematical Modelling and Algorithms*, 9(2):131–147, 2010.

[36] N. Bicocchi and M. Mamei. Investigating ride sharing opportunities through mobility data analysis. *Pervasive and Mobile Computing*, 14:83–94, 2014.

[37] K. Bischoff. We love rock'n'roll: analyzing and predicting friendship links in last. fm. In *Proceedings of the 4th Annual ACM Web Science Conference*, pages 47–56. ACM, 2012.

[38] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[39] C. A. Bliss, M. R. Frank, C. M. Danforth, and P. S. Dodds. An evolutionary algorithm approach to link prediction in dynamic social networks. *arXiv preprint arXiv:1304.6257*, 2013.

[40] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[41] A. Botea, M. Berlingerio, E. Bouillet, S. Braghin, F. Calabrese, B. Chen, Y. Gkoufas, M. Laummans, R. Nair, and T. Nonner. Docit: an integrated system for risk-aware multimodal journey advising. *IBM Research, Dublin, Ireland, Tech. Rep*, 2014.

[42] A. Botea, S. Braghin, N. Lopes, R. Guidotti, and F. Pratesi. Managing travels with petra: the rome use case. In *ICDE*. IEEE, 2015.

[43] A. Botea, E. Nikolova, and M. Berlingerio. Multi-modal journey planning in the presence of uncertainty. In *ICAPS*, 2013.

[44] M. Bouguessa. A practical approach for clustering transaction data. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pages 265–279. Springer, 2011.

[45] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 43–52. Morgan Kaufmann Publishers Inc., 1998.

[46] M. Brunato and R. Battiti. Pilgrim: A location broker and mobility-aware recommendation system. In *Pervasive Computing and Communications, 2003.(PerCom 2003). Proceedings of the First IEEE International Conference on*, pages 265–272. IEEE, 2003.

[47] P. Brusilovsky, A. Kobsa, and W. Nejdl. *The adaptive web: methods and strategies of web personalization*, volume 4321. Springer, 2007.

[48] I. E. Burbey. *Predicting future locations and arrival times of individuals.* PhD thesis, Virginia Polytechnic Institute and State University, 2011.

[49] J. Burez and D. Van den Poel. Handling class imbalance in customer churn prediction. *Expert Systems with Applications*, 36(3):4626–4636, 2009.

[50] R. Burke. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12(4):331–370, 2002.

[51] F. Calabrese, G. Di Lorenzo, and C. Ratti. Human mobility prediction based on individual and collective geographical preferences. In *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pages 312–317. IEEE, 2010.

[52] G. Caldarelli, M. Cristelli, A. Gabrielli, L. Pietronero, A. Scala, and A. Tacchella. Ranking and clustering countries and their products; a network analysis. *arXiv preprint arXiv:1108.2590*, 2011.

[53] F. Cao, J. Liang, and L. Bai. A new initialization method for categorical data clustering. *Expert Systems with Applications*, 36(7):10223–10228, 2009.

[54] L. Cao, J. Luo, A. C. Gallagher, X. Jin, J. Han, and T. S. Huang. A worldwide tourism recommendation system based on geotaggedweb photos. In *ICASSP*, pages 2274–2277. Citeseer, 2010.

[55] L.-J. Cao and F. E. H. Tay. Support vector machine with adaptive parameters in financial time series forecasting. *Neural Networks, IEEE Transactions on*, 14(6):1506–1518, 2003.

[56] Y. Cao and Y. Li. An intelligent fuzzy-based recommendation system for consumer electronic products. *Expert Systems with Applications*, 33(1):230–240, 2007.

[57] J. P. Carrascal, C. Riederer, V. Erramilli, M. Cherubini, and R. de Oliveira. Your browsing behavior for a big mac: Economics of personal information online. In *Proceedings of the 22nd international conference on World Wide Web*, pages 189–200. ACM, 2013.

[58] A. Cavoukian. Privacy design principles for an integrated justice system. Technical report, Working paper. www.âĂŃ ipc.âĂŃ on.âĂŃ ca/âĂŃ index.âĂŃ asp, 2000.

[59] M. Ceci, A. Appice, and D. Malerba. Time-slice density estimation for semantic-based tourist destination suggestion. In *ECAI*, pages 1107–1108, 2010.

[60] E. Cesario, G. Manco, and R. Ortale. Top-down parameter-free clustering of high-dimensional categorical data. *IEEE Transactions on Knowledge and Data Engineering*, 19(12):1607–1624, 2007.

[61] S. Chawla. Feature selection, association rules network and theory building. In *FSDM*, pages 14–21, 2010.

[62] K. Chen and L. Liu. The" best k" for entropy-based categorical data clustering. 2005.

[63] L. Chen, M. Lv, and G. Chen. A system for destination and future route prediction based on trajectory mining. *Pervasive and Mobile Computing*, 6(6):657–676, 2010.

[64] M. Chen, Y. Liu, and X. Yu. Nlpmm: a next location predictor with markov modeling. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 186–197. Springer, 2014.

[65] M.-C. Chen, A.-L. Chiu, and H.-H. Chang. Mining changes in customer behavior in retail marketing. *Expert Systems with Applications*, 28(4):773–781, 2005.

[66] X. Chen, J. Z. Huang, and J. Luo. Purtreeclust: A purchase tree clustering algorithm for large-scale customer transaction data. In *ICDE*, pages 661–672. IEEE, 2016.

[67] Z.-Y. Chen and Z.-P. Fan. Distributed customer behavior prediction using multiplex data: A collaborative mk-svm approach. *Knowledge-Based Systems*, 35:111–119, 2012.

[68] K.-W. Cheung, J. T. Kwok, M. H. Law, and K.-C. Tsui. Mining customer product ratings for personalized marketing. *Decision Support Systems*, 35(2):231–243, 2003.

[69] E. Cho, S. A. Myers, and J. Leskovec. Friendship and mobility: user movement in location-based social networks. In *KDD*, pages 1082–1090, 2011.

[70] S. Cho, A.-U.-H. Yasar, L. Knapen, T. Bellemans, D. Janssens, and G. Wets. A conceptual design of an agent-based interaction model for the carpooling application. *Procedia Computer Science*, 10:801–807, 2012.

[71] Y. B. Cho, Y. H. Cho, and S. H. Kim. Mining changes in customer buying behavior for collaborative recommendations. *Expert Systems with Applications*, 28(2):359–369, 2005.

[72] Y. H. Cho, J. K. Kim, and S. H. Kim. A personalized recommender system based on web usage mining and decision tree induction. *Expert Systems with Applications*, 23(3):329–342, 2002.

[73] B. H. Choi. For whom the data tolls: A reunified theory of fourth and fifth amendment jurisprudence. *Cardozo Law Review*, 37:185, 2015.

[74] H. Choi and H. Varian. Predicting the present with google trends. *Economic Record*, 88(s1):2–9, 2012.

[75] A. K. Chorppath and T. Alpcan. Trading privacy with incentives in mobile commerce: A game theoretic approach. *Pervasive and Mobile Computing*, 9(4):598–612, 2013.

[76] B. Cici, A. Markopoulou, E. Frias-Martinez, and N. Laoutaris. Assessing the potential of ride-sharing using mobile and social data: a tale of four cities. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 201–211. ACM, 2014.

[77] P. Cintia and M. Nanni. An effective time-aware map matching process for low sampling gps data. Technical Report cnr.isti/2015-TR-011.

[78] P. Cintia, R. Trasarti, L. Cruz, C. Costa, and J. A. F. de Macedo. A gravity model for speed estimation over road network. In *Mobile Data Management (MDM), 2013 IEEE 14th International Conference on*, volume 2, pages 136–141. IEEE, 2013.

[79] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5):603–619, 2002.

[80] D. Coppersmith, T. Nowicki, G. Paleologo, C. Tresser, and C. W. Wu. The optimality of the online greedy algorithm in carpool and chairman assignment problems. *TALG*, 7(3):37, 2011.

[81] G. Correia and J. M. Viegas. Applying a structured simulation-based methodology to assess carpooling time–space potential. *Transportation Planning and Technology*, 33(6):515–540, 2010.

[82] G. Correia and J. M. Viegas. Carpooling and carpool clubs: Clarifying concepts and assessing value enhancement possibilities through a stated preference web survey in lisbon, portugal. *Transportation Research Part A: Policy and Practice*, 45(2):81–90, 2011.

[83] M. Coscia, F. Giannotti, and D. Pedreschi. A classification for community discovery methods in complex networks. *Stat. Anal. Data Min.*, 4(5):512–546, Oct. 2011.

[84] M. Coscia, G. Rossetti, F. Giannotti, and D. Pedreschi. Demon: a local-first discovery method for overlapping communities. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 615–623. ACM, 2012.

[85] R. Costanza, I. Kubiszewski, E. Giovannini, H. Lovins, J. McGlade, K. E. Pickett, K. V. Ragnarsdóttir, D. Roberts, R. De Vogli, and R. Wilkinson. Time to leave gdp behind. *Nature Comment*, 2014.

[86] M. Cristelli, A. Gabrielli, A. Tacchella, G. Caldarelli, and L. Pietronero. Measuring the intangibles: A metrics for the economic complexity of countries and products. *PloS one*, 8(8):e70726, 2013.

[87] C. Cumby, A. Fano, R. Ghani, and M. Krema. Predicting customer shopping lists from point-of-sale purchase data. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 402–409. ACM, 2004.

[88] E. M. Daly, F. Lecue, and V. Bicer. Westland row why so slow?: fusing social media and linked data sources for understanding real-time traffic conditions. In *Proceedings of the 2013 international conference on Intelligent user interfaces*, pages 203–212. ACM, 2013.

[89] M. De Domenico, A. Lima, and M. Musolesi. Interdependence and predictability of human mobility and social interactions. *Pervasive and Mobile Computing*, 9(6):798–807, 2013.

[90] Y.-A. de Montjoye, E. Shmueli, S. S. Wang, and A. S. Pentland. openpds: Protecting the privacy of metadata through safeanswers. *PloS one*, 9(7):e98790, 2014.

[91] Y.-A. de Montjoye, S. S. Wang, A. Pentland, D. T. T. Anh, A. Datta, et al. On the trusted use of large-scale personal data. *IEEE Data Eng. Bull.*, 35(4):5–8, 2012.

[92] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[93] I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 269–274. ACM, 2001.

[94] R. R. Dholakia and N. Dholakia. Mobility and markets: emerging outlines of m-commerce. *Journal of Business research*, 57(12):1391–1396, 2004.

[95] J.-P. Dittrich and M. A. V. Salles. idm: A unified and versatile data model for personal dataspace management. In *Proceedings of the 32nd international conference on Very large data bases*, pages 367–378. VLDB Endowment, 2006.

[96] T. M. T. Do and D. Gatica-Perez. Where and what: Using smartphones to predict next locations and applications in daily life. *Pervasive and Mobile Computing*, 12:79–91, 2014.

[97] N. R. Draper, H. Smith, and E. Pownell. *Applied regression analysis*, volume 3. Wiley New York, 1966.

[98] E. Dumbill. What is big data, 2012.

[99] A. S. Dunk. Product life cycle cost analysis: the impact of customer profiling, competitive advantage, and quality of is information. *Management Accounting Research*, 15(4):401–414, 2004.

[100] D. Easley and J. Kleinberg. *Networks, crowds, and markets: Reasoning about a highly connected world*. Cambridge University Press, 2010.

[101] F. Ellis-Chadwick, R. Mayer, K. Johnston, and D. Chaffey. *Internet marketing: strategy, implementation and practice*. Pearson Education, 2009.

[102] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[103] C. Foroni and M. Marcellino. A comparison of mixed frequency approaches for nowcasting euro area macroeconomic aggregates. *International Journal of Forecasting*, 30(3):554–568, 2014.

[104] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315(5814):972–976, 2007.

[105] B. Furletti, L. Gabrielli, C. Renso, and S. Rinzivillo. Identifying users profiles from mobile calls habits. In *Proceedings of the ACM SIGKDD International Workshop on Urban Computing*, pages 17–24. ACM, 2012.

[106] J. W. Galbraith and G. Tkacz. Nowcasting gdp with electronic payments data. 2015.

[107] S. Galland, N. Gaud, A.-U.-H. Yasar, L. Knapen, D. Janssens, and O. Lamotte. Simulation model of carpooling with the janus multiagent platform. *Procedia Computer Science*, 19:860–866, 2013.

[108] S. Galland, L. Knapen, A.-U.-H. Yasar, N. Gaud, D. Janssens, O. Lamotte, A. Koukam, and G. Wets. Multi-agent simulation of individual mobility behavior in carpooling. *Transportation Research Part C: Emerging Technologies*, 2014.

[109] S. Galland, A. Yasar, L. Knapen, N. Gaud, D. Janssens, O. Lamotte, G. Wets, and A. Koukam. Multi-agent simulation of individual mobility behavior in carpooling using the janus and jasim platforms. *Transportation Research Part C*, 2013.

[110] G. Gan and J. Wu. Subspace clustering for high dimensional categorical data. *ACM SIGKDD Explorations Newsletter*, 6(2):87–94, 2004.

[111] L. Geng and H. J. Hamilton. Interestingness measures for data mining: A survey. *ACM Computing Surveys (CSUR)*, 38(3):9, 2006.

[112] R. Ghani and A. Fano. Building recommender systems using a knowledge base of product semantics. In *Proceedings of the Workshop on Recommendation and Personalization in ECommerce at the 2nd International Conference on Adaptive Hypermedia and Adaptive Web based Systems*, pages 27–29, 2002.

[113] J. Ghosh, M. J. Beal, H. Q. Ngo, and C. Qiao. On profiling mobility and predicting locations of wireless users. In *Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality*, pages 55–62. ACM, 2006.

[114] D. Giannone, L. Reichlin, and D. Small. Nowcasting: The real-time informational content of macroeconomic data. *Journal of Monetary Economics*, 55(4):665–676, 2008.

[115] F. Giannotti, C. Gozzi, and G. Manco. Clustering transactional data. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 175–187. Springer, 2002.

[116] F. Giannotti, M. Nanni, D. Pedreschi, F. Pinelli, C. Renso, S. Rinzivillo, and R. Trasarti. Unveiling the complexity of human mobility by querying and mining massive trajectory data. *The VLDB JournalÃćÂĂŤThe International Journal on Very Large Data Bases*, 20(5):695–719, 2011.

[117] F. Giannotti, M. Nanni, D. Pedreschi, F. Pinelli, C. Renso, S. Rinzivillo, and R. Trasarti. Unveiling the complexity of human mobility by querying and mining massive trajectory data. *VLDB Journal*, 20(5), 2011.

[118] F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi. Trajectory pattern mining. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 330–339. ACM, 2007.

[119] F. Giannotti, D. Pedreschi, A. Pentland, P. Lukowicz, D. Kossmann, J. Crowley, and D. Helbing. A planetary nervous system for social mining and collective awareness. *The European Physical Journal Special Topics*, 214(1):49–75, 2012.

[120] G. Gidófalvi and F. Dong. When and where next: individual mobility prediction. In *Proceedings of the First ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems*, pages 57–64. ACM, 2012.

[121] R. Giegerich and S. Kurtz. From ukkonen to mccreight and weiner: A unifying view of linear-time suffix tree construction. *Algorithmica*, 19(3):331–353, 1997.

[122] J. B. Gomes, C. Phua, and S. Krishnaswamy. Where will you go? mobile data mining for next place prediction. In *International Conference on Data Warehousing and Knowledge Discovery*, pages 146–158. Springer, 2013.

[123] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, 2008.

[124] E. Guerra, R. Cervero, and D. Tischler. The half-mile circle: Does it best represent transit station catchments? *Transportation Research Record: Journal of the Transportation Research Board*, 2276:101 – 109, 2012.

[125] S. Guha, R. Rastogi, and K. Shim. Rock: A robust clustering algorithm for categorical attributes. In *Data Engineering, 1999. Proceedings., 15th International Conference on*, pages 512–521. IEEE, 1999.

[126] R. Guidotti. *Mobility ranking-human mobility analysis using ranking measures.* University of Pisa, Pisa, 2013.

[127] R. Guidotti and M. Berlingerio. Where is my next friend? recommending enjoyable profiles in location based services. In *Complex Networks VII*, pages 65–78. Springer, 2016.

[128] R. Guidotti and P. Cintia. Towards a boosted route planner using individual mobility models. In *International Conference on Software Engineering and Formal Methods*, pages 108–123. Springer, 2015.

[129] R. Guidotti, M. Coscia, D. Pedreschi, and D. Pennacchioli. Behavioral entropy and profitability in retail. In *Data Science and Advanced Analytics (DSAA), 2015. 36678 2015. IEEE International Conference on*, pages 1–10. IEEE, 2015.

[130] R. Guidotti, M. Coscia, D. Pedreschi, and D. Pennacchioli. Going beyond gdp to nowcast well-being using retail market data. In *International Conference and School on Network Science*, pages 29–42. Springer, 2016.

[131] R. Guidotti, A. Monreale, S. Rinzivillo, D. Pedreschi, and F. Giannotti. Retrieving points of interest from human systematic movements. In *Software Engineering and Formal Methods*, pages 294–308. Springer, 2014.

[132] R. Guidotti, A. Monreale, S. Rinzivillo, D. Pedreschi, and F. Giannotti. Unveiling mobility complexity through complex network analysis. *Social Network Analysis and Mining*, 6(1):59, 2016.

[133] R. Guidotti, M. Nanni, S. Rinzivillo, D. Pedreschi, and F. Giannotti. Never drive alone: Boosting carpooling with network analysis. *Information Systems*, 2016.

[134] R. Guidotti, G. Rossetti, and D. Pedreschi. Audio ergo sum a personal data model for musical preferences. 2016.

[135] R. Guidotti, A. Sassi, M. Berlingerio, A. Pascale, and B. Ghaddar. Social or green? a data-driven approach for more enjoyable carpooling. In *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*, pages 842–847. IEEE, 2015.

[136] R. Guidotti, R. Trasarti, and M. Nanni. Tosca: Two-steps clustering algorithm for personal locations detection. In *SIGSPATIAL*, pages 38:1–38:10. ACM, 2015.

[137] R. Guidotti, R. Trasarti, M. Nanni, and F. Giannotti. Towards user-centric data management: individual mobility analytics for collective services. In *Proceedings of the 4th ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems*, pages 80–83. ACM, 2015.

[138] J. F. Hair, W. C. Black, B. J. Babin, R. E. Anderson, R. L. Tatham, et al. Multivariate data analysis (vol. 6), 2006.

[139] Y. Hamuro, N. Katoh, I. H. Edward, S. L. Cheung, and K. Yada. Combining information fusion with string pattern analysis: a new method for predicting future purchase behavior. In *Information Fusion in Data Mining*, pages 161–187. Springer, 2003.

[140] T. Hansen, J. M. Jensen, and H. S. Solgaard. Predicting online grocery buying intention: a comparison of the theory of reasoned action and the theory of planned behavior. *International Journal of Information Management*, 24(6):539–550, 2004.

[141] J. A. Hartigan and M. A. Wong. Algorithm as 136: A k-means clustering algorithm. *Applied statistics*, pages 100–108, 1979.

[142] R. Hausmann, C. A. Hidalgo, S. Bustos, M. Coscia, A. Simoes, and M. A. Yildirim. *The atlas of economic complexity: Mapping paths to prosperity.* Mit Press, 2014.

[143] D. Helbing. The automation of society is next: How to survive the digital revolution. *Available at SSRN 2694312*, 2015.

[144] D. Helbing and S. Balietti. How to create an innovation accelerator. *The European Physical Journal Special Topics*, 195(1):101–136, 2011.

[145] M. Hilbert and P. López. The worldâĂŹs technological capacity to store, communicate, and compute information. *science*, 332(6025):60–65, 2011.

[146] M. Hildebrandt. Defining profiling: a new type of knowledge? In *Profiling the European citizen*, pages 17–45. Springer, 2008.

[147] A. Hinneburg and D. A. Keim. Optimal grid-clustering: Towards breaking the curse of dimensionality in high-dimensional clustering. In *VLDB*, 1999.

[148] D. C. Hoaglin, F. Mosteller, and J. W. Tukey. *Understanding robust and exploratory data analysis*, volume 3. Wiley New York, 1983.

[149] S. Holm. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, pages 65–70, 1979.

[150] J. I. Hong and J. A. Landay. An architecture for privacy-sensitive ubiquitous computing. In *Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pages 177–189. ACM, 2004.

[151] L. Huang, Q. Li, and Y. Yue. Activity identification from gps trajectories using spatial temporal pois' attractiveness. In *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks*, pages 27–30. ACM, 2010.

[152] Z. Huang. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data mining and knowledge discovery*, 2(3):283–304, 1998.

[153] C.-C. Hung, C.-W. Chang, and W.-C. Peng. Mining trajectory profiles for discovering user communities. In *Proceedings of the 2009 International Workshop on Location Based Social Networks*, pages 1–8. ACM, 2009.

[154] L.-p. Hung. A personalized recommendation system based on product taxonomy for one-to-one marketing online. *Expert systems with applications*, 29(2):383–392, 2005.

[155] T. Ishioka. An expansion of x-means for automatically determining the optimal number of clusters. In *Procs of Int. Conf. on Computational Intelligence*, pages 91–96, 2005.

[156] H. Jeung, Q. Liu, H. T. Shen, and X. Zhou. A hybrid prediction model for moving objects. In *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, pages 70–79. IEEE, 2008.

[157] H. Jeung, M. L. Yiu, X. Zhou, and C. S. Jensen. Path prediction and predictive range querying in road network databases. *The VLDB Journal*, 19(4):585–602, 2010.

[158] S. Johnson. *Future perfect: The case for progress in a networked age.* Penguin UK, 2012.

[159] S. C. Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967.

[160] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1):99–134, 1998.

[161] C. Kalapesi. Unlocking the value of personal data: From collection to usage. In *World Economic Forum technical report*, 2013. `http://www3.weforum.org/docs/WEF_IT_UnlockingValuePersonalData_CollectionUsage_Report_2013.pdf`.

[162] J. Kang and H.-S. Yong. A frequent pattern based prediction model for moving objects. *Int. J. Comput. Sci. Netw. Secur*, 10(3):200–205, 2010.

[163] R. E. Kass and L. Wasserman. A reference bayesian test for nested hypotheses and its relationship to the schwarz criterion. *Journal of the american statistical association*, 90(431):928–934, 1995.

[164] L. Kaufman and P. Rousseeuw. *Clustering by means of medoids*. North-Holland, 1987.

[165] E. Kim, W. Kim, and Y. Lee. Combination of multiple classifiers for the customer's purchase behavior prediction. *Decision Support Systems*, 34(2):167–175, 2003.

[166] K.-j. Kim. Financial time series forecasting using support vector machines. *Neurocomputing*, 55(1):307–319, 2003.

[167] S.-W. Kim, J.-I. Won, J.-D. Kim, M. Shin, J. Lee, and H. Kim. Path prediction of moving objects on road networks through analyzing past trajectories. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pages 379–389. Springer, 2007.

[168] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, 1999.

[169] L. Knapen, D. Keren, A.-U.-H. Yasar, S. Cho, T. Bellemans, D. Janssens, and G. Wets. Estimating scalability issues while finding an optimal assignment for carpooling. *Procedia Computer Science*, 19:372–379, 2013.

[170] I. D. Kocakoç and S. Erdem. Business intelligence applications in retail business: Olap, data mining & reporting services. *Journal of Information & Knowledge Management*, 9(02):171–181, 2010.

[171] M. R. Korupolu, C. G. Plaxton, and R. Rajaraman. Analysis of a local search heuristic for facility location problems. *Journal of algorithms*, 37(1):146–188, 2000.

[172] J. Kotrlik and C. Higgins. Organizational research: Determining appropriate sample size in survey research appropriate sample size in survey research. *ITLPJ*, 19(1):43, 2001.

[173] L. Kotthoff, M. Nanni, R. Guidotti, and B. OâĂŹSullivan. Find your way back: Mobility profile mining with constraints. In *International Conference on Principles and Practice of Constraint Programming*, pages 638–653. Springer, 2015.

[174] J. A. Kroll, I. C. Davey, and E. W. Felten. The economics of bitcoin mining, or bitcoin in the presence of adversaries. In *Proceedings of WEIS*, volume 2013. Citeseer, 2013.

[175] J. Krumm and E. Horvitz. Predestination: Inferring destinations from partial trajectories. In *UbiComp 2006: Ubiquitous Computing*, pages 243–260. Springer, 2006.

[176] C. Krumme, A. Llorente, M. Cebrian, E. Moro, et al. The predictability of consumer visitation patterns. *Scientific reports*, 3, 2013.

[177] N. Lathia and L. Capra. Mining mobility data to minimise travellers' spending on public transport. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1181–1189. ACM, 2011.

[178] P. A. Lawn. A theoretical foundation to support the index of sustainable economic welfare (isew), genuine progress indicator (gpi), and other related indexes. *Ecological Economics*, 44(1):105–118, 2003.

[179] P. A. Lawn. An assessment of the valuation methods used to calculate the index of sustainable economic welfare (isew), genuine progress indicator (gpi), and sustainable net benefit index (snbi). *Environment, Development and Sustainability*, 7(2):185–208, 2005.

[180] D. Lazer, R. Kennedy, G. King, and A. Vespignani. The parable of google flu: traps in big data analysis. *Science*, 343(14 March), 2014.

[181] D. Lazer, A. S. Pentland, L. Adamic, S. Aral, A. L. Barabasi, D. Brewer, N. Christakis, N. Contractor, J. Fowler, M. Gutmann, et al. Life in the network: the coming age of computational social science. *Science (New York, NY)*, 323(5915):721, 2009.

[182] D. Lee and S. H. Liang. Crowd-sourced carpool recommendation based on simple and efficient trajectory grouping. In *SIGSPATIAL*, pages 12–17, 2011.

[183] P.-R. Lei, T.-J. Shen, W.-C. Peng, and J. Su. Exploring spatial-temporal trajectory model for location prediction. In *Mobile Data Management (MDM), 2011 12th IEEE International Conference on*, volume 1, pages 58–67. IEEE, 2011.

[184] V. Lerenc. Increasing throughput for carpool assignment matching, Dec. 19 2011. US Patent App. 13/329,899.

[185] J. Letchner, J. Krumm, and E. Horvitz. Trip router with individualized preferences (trip): Incorporating personalization into route planning. In *Proceedings of the National Conference on Artificial Intelligence*, volume 21, page 1795. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006.

[186] C. Leventi, J. Navicke, O. Rastrigina, and H. Sutherland. Nowcasting the income distribution in europe. 2014.

[187] A. Q. Li, A. Ahmed, S. Ravi, and A. J. Smola. Reducing the sampling complexity of topic models. In *KDD*, pages 891–900, 2014.

[188] H. Li, C. Tang, S. Qiao, Y. Wang, N. Yang, and C. Li. Hotspot district trajectory prediction. In *Web-Age Information Management*, pages 74–84. Springer, 2010.

[189] T. Li, S. Ma, and M. Ogihara. Entropy-based criterion in categorical clustering. In *Proceedings of the twenty-first international conference on Machine learning*, page 68. ACM, 2004.

[190] D. Liben-Nowell and J. Kleinberg. The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7):1019–1031, 2007.

[191] T. Liebig, N. Piatkowski, C. Bockermann, and K. Morik. Predictive trip planning-smart routing in smart cities. In *EDBT/ICDT Workshops*, pages 331–338, 2014.

[192] H.-F. Lin. Predicting consumer intentions to shop online: An empirical test of competing theories. *Electronic Commerce Research and Applications*, 6(4):433–442, 2008.

[193] M. Lin and W.-J. Hsu. Brownian bridge model for high resolution location predictions. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 210–221. Springer, 2014.

[194] P. Lingras, M. Hogo, M. Snorek, and C. West. Temporal analysis of clusters of supermarket customers: conventional versus interval set approach. *Information Sciences*, 172(1):215–240, 2005.

[195] A. Llorente, M. Cebrian, E. Moro, et al. Social media fingerprints of unemployment. *arXiv preprint arXiv:1411.3140*, 2014.

[196] S. Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[197] E.-C. Lu, V. S. Tseng, and P. S. Yu. Mining cluster-based temporal mobile sequential patterns in location-based service environments. *Knowledge and Data Engineering, IEEE Transactions on*, 23(6):914–927, 2011.

[198] L. Lü and T. Zhou. Link prediction in complex networks: A survey. *Physica A: Statistical Mechanics and its Applications*, 390(6):1150–1170, 2011.

[199] V. Maniezzo, A. Carbonaro, and H. Hildmann. *New Optimization Techniques in Engineering*, chapter An ANTS Heuristic for the Long — Term Car Pooling Problem, pages 411–430. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[200] C. D. Manning, P. Raghavan, et al. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[201] G. Marketos, E. Frentzos, I. Ntoutsi, N. Pelekis, A. Raffaetà, and Y. Theodoridis. Building real-world trajectory warehouses. In *Proceedings of the seventh ACM international workshop on data engineering for wireless and mobile access*, pages 8–15. ACM, 2008.

[202] D. W. Massaro, B. Chaney, S. Bigler, J. Lancaster, S. Iyer, M. Gawade, M. Eccleston, E. Gurrola, and A. Lopez. Carpoolnow-just-in-time carpooling without elaborate preplanning. In *WEBIST*, pages 219–224, 2009.

[203] W. J. McDonald. Time use in shopping: the role of personal characteristics. *Journal of Retailing*, 70(4):345–365, 1994.

[204] L. McGinty and B. Smyth. Turas: a personalised route planning system. In *PRICAI 2000 Topics in Artificial Intelligence*, pages 791–791. Springer, 2000.

[205] M. McPherson, L. Smith-Lovin, and J. M. Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, pages 415–444, 2001.

[206] A. Mild and T. Reutterer. An improved collaborative filtering approach for predicting cross-category purchases based on binary market basket data. *JRCS*, 10(3):123–133, 2003.

[207] C. Moiso, F. Antonelli, and M. Vescovi. How do i manage my personal data?-a telco perspective. In *DATA*, pages 123–128, 2012.

[208] C. Moiso and R. Minerva. Towards a user-centric personal data ecosystem the role of the bank of individuals' data. In *Intelligence in Next Generation Networks (ICIN), 2012 16th International Conference on*, pages 202–209. IEEE, 2012.

[209] A. Mokhtari, O. Pivert, and A. Hadjali. Integrating complex user preferences into a route planner: A fuzzy-set-based approach. 2009.

[210] A. Monreale, G. L. Andrienko, N. V. Andrienko, F. Giannotti, D. Pedreschi, S. Rinzivillo, and S. Wrobel. Movement data anonymity through generalization. *Transactions on Data Privacy*, 3(2):91–121, 2010.

[211] A. Monreale, F. Pinelli, R. Trasarti, and F. Giannotti. Wherenext: a location predictor on trajectory pattern mining. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 637–646. ACM, 2009.

[212] A. Monreale, S. Rinzivillo, F. Pratesi, F. Giannotti, and D. Pedreschi. Privacy-by-design in big data analytics and social mining. *EPJ Data Science*, 3(1):1, 2014.

[213] B. C. Monsell. Update on the development of x-13arima-seats. In *Proceedings of the Joint Statistical Meetings: American Statistical Association*, 2009.

[214] R. J. Mooney and L. Roy. Content-based book recommending using learning for text categorization. In *Proceedings of the fifth ACM conference on Digital libraries*, pages 195–204. ACM, 2000.

[215] M. Morzy. Prediction of moving object location based on frequent trajectories. In *International Symposium on Computer and Information Sciences*, pages 583–592. Springer, 2006.

[216] M. Morzy. Mining frequent trajectories of moving objects for location prediction. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pages 667–680. Springer, 2007.

[217] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.

[218] J. Naoum-Sawaya, R. Cogill, B. Ghaddar, S. Sajja, R. Shorten, N. Taheri, P. Tommasi, R. Verago, and F. Wirth. Stochastic optimization approach for the car placement problem in ridesharing systems. *Transportation Research Part B: Methodological*, 80:173 – 184, 2015.

[219] J. Navicke, O. Rastrigina, and H. Sutherland. Nowcasting indicators of poverty risk in the european union: a microsimulation approach. *Social Indicators Research*, 119(1):101–119, 2014.

[220] M. E. Nergiz, M. Atzori, and Y. Saygin. Towards trajectory anonymization: a generalization-based approach. In *Proceedings of the SIGSPATIAL ACM GIS 2008 International Workshop on Security and Privacy in GIS and LBS*, pages 52–61. ACM, 2008.

[221] M. Nishino, Y. Nakamura, T. Yagi, S. Muto, and M. Abe. A location predictor based on dependencies between multiple lifelog data. In *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks*, pages 11–17. ACM, 2010.

[222] A. Noulas, S. Scellato, R. Lambiotte, M. Pontil, and C. Mascolo. A tale of many cities: universal patterns in human urban mobility. *PloS one*, 7(5):e37027, 2012.

[223] J. S. Olson, J. Grudin, and E. Horvitz. A study of preferences for sharing and privacy. In *CHI'05 extended abstracts on Human factors in computing systems*, pages 1985–1988. ACM, 2005.

[224] B. Padmanabhan, Z. Zheng, and S. O. Kimbrough. Personalization from incomplete data: what you don't know can hurt. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 154–163. ACM, 2001.

[225] R. Pálovics and A. A. Benczúr. Temporal influence over the last. fm social network. *Social Network Analysis and Mining*, 5(1):1–12, 2015.

[226] L. Pappalardo, S. Rinzivillo, Z. Qu, D. Pedreschi, and F. Giannotti. Understanding the patterns of car travel. *EPJ Special Topics*, 215(1):61–73, 2013.

[227] L. Pappalardo, F. Simini, S. Rinzivillo, D. Pedreschi, F. Giannotti, and A.-L. Barabási. Returners and explorers dichotomy in human mobility. *Nature communications*, 6, 2015.

[228] M.-H. Park, J.-H. Hong, and S.-B. Cho. Location-based recommendation system using bayesian userâĂŹs preference model in mobile devices. In *Ubiquitous Intelligence and Computing*, pages 1130–1139. Springer, 2007.

[229] A. Pascale, T. L. Hoang, and R. Nair. Characterization of network traffic processes under adaptive traffic control systems. *Transportation Research Part C: Emerging Technologies*, 2015.

[230] M. J. Pazzani and D. Billsus. Content-based recommendation systems. In *The adaptive web*, pages 325–341. Springer, 2007.

[231] D. Pedreschi. Big data, social mining, diversity, and wellbeing. In *SIS*, pages 1–6, 2014.

[232] D. Pelleg, A. W. Moore, et al. X-means: Extending k-means with efficient estimation of the number of clusters. In *ICML*, volume 1, 2000.

[233] M.-P. Pelletier, M. Trépanier, and C. Morency. Smart card data use in public transit: A literature review. *Transportation Research Part C: Emerging Technologies*, 19(4):557–568, 2011.

[234] D. Pennacchioli, M. Coscia, and D. Pedreschi. Overlap versus partition: marketing classification and customer profiling in complex networks of products. In *Data Engineering Workshops (ICDEW), 2014 IEEE 30th International Conference on*, pages 103–110. IEEE, 2014.

[235] D. Pennacchioli, M. Coscia, S. Rinzivillo, F. Giannotti, and D. Pedreschi. The retail market as a complex system. *EPJ Data Science*, 3(1):1, 2014.

[236] D. Pennacchioli, M. Coscia, S. Rinzivillo, D. Pedreschi, and F. Giannotti. Explaining the product range effect in purchase data. In *Big Data, 2013 IEEE International Conference on*, pages 648–656. IEEE, 2013.

[237] D. Pennacchioli, G. Rossetti, L. Pappalardo, D. Pedreschi, F. Giannotti, and M. Coscia. The three dimensions of social prominence. In *Social Informatics*, pages 319–332. Springer, 2013.

[238] A. Pentland. Reinventing society in the wake of big data. *Edge. Available online at: http://www. edge. org/conversation/reinventing-society-in-the-wake-of-big-data*, 2012.

[239] A. Pentland et al. Personal data: The emergence of a new asset class. In *World Economic Forum*, 2011. `http://www3.weforum.org/docs/WEF_ITTC_PersonalDataNewAsset_Report_2011.pdf`.

[240] D. M. W. Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *Int. J. of Machine Learning Technology*, 2(1):37–63, 2011.

[241] M. Pujari and R. Kanawati. Supervised rank aggregation approach for link prediction in complex networks. In *Proceedings of the 21st international conference companion on World Wide Web*, pages 1189–1196. ACM, 2012.

[242] J. Putzke, K. Fischbach, D. Schoder, and P. A. Gloor. Cross-cultural gender differences in the adoption and usage of social media platforms–an exploratory study of last. fm. *Computer Networks*, 75:519–530, 2014.

[243] D. Qiu, P. Papotti, and L. Blanco. Future locations prediction with uncertain data. In *Machine Learning and Knowledge Discovery in Databases*, pages 417–432. Springer, 2013.

[244] D. Quercia, R. Schifanella, and L. M. Aiello. The shortest path to happiness: Recommending beautiful, quiet, and happy routes in the city. In *Conference on Hypertext and social media*, pages 116–125. ACM, 2014.

[245] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971.

[246] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *WWW*, pages 811–820. ACM, 2010.

[247] F. Ricci. Travel recommender systems. *IEEE Intelligent Systems*, 17(6):55–57, 2002.

[248] F. Ricci, L. Rokach, and B. Shapira. *Introduction to recommender systems handbook*. Springer, 2011.

[249] S. Rinzivillo, L. Gabrielli, M. Nanni, L. Pappalardo, D. Pedreschi, and F. Giannotti. The purpose of motion: Learning activities from individual mobility networks. In *Data Science and Advanced Analytics (DSAA), 2014 International Conference on*, pages 312–318. IEEE, 2014.

[250] J. Rose and C. Kalapesi. Rethinking personal data: Strengthening trust. In *World Economic Forum*, 2012. http://www3.weforum.org/docs/WEF_IT_RethinkingPersonalData_Report_2012.pdf.

[251] A. Rosenberg and J. Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *EMNLP-CoNLL*, volume 7, pages 410–420, 2007.

[252] G. Rossetti, R. Guidotti, I. Miliou, D. Pedreschi, and F. Giannotti. A supervised approach for intra/inter-community interaction prediction in dynamic social networks. *Social Network Analysis and Mining*, 2016.

[253] G. Rossetti, R. Guidotti, D. Pennacchioli, D. Pedreschi, and F. Giannotti. Interaction prediction in dynamic networks exploiting community discovery. In *2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 553–558. IEEE, 2015.

[254] M. Rosvall and C. T. Bergstrom. Multilevel compression of random walks on networks reveals hierarchical organization in large integrated systems. *PloS one*, 6(4):e18209, 2011.

[255] G. Salton and M. J. McGill. Introduction to modern information retrieval. 1986.

[256] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295. ACM, 2001.

[257] S. Scellato, M. Musolesi, C. Mascolo, V. Latora, and A. T. Campbell. Nextplace: a spatio-temporal prediction framework for pervasive systems. In *International Conference on Pervasive Computing*, pages 152–169. Springer, 2011.

[258] C. Scholz, M. Atzmueller, and G. Stumme. On the predictability of human contacts: Influence factors and the strength of stronger ties. In *Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom)*, pages 312–321. IEEE, 2012.

[259] G. Schwarz et al. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.

[260] M. Sghaier, H. Zgaya, S. Hammadi, and C. Tahon. A distributed dijkstra's algorithm for the implementation of a real time carpooling service with an optimized aspect on siblings. In *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pages 795–800. IEEE, 2010.

[261] L. Shangguan, Z. Zhou, X. Zheng, L. Yang, Y. Liu, and J. Han. Shopminer: Mining customer shopping behavior in physical clothing stores with cots rfid devices. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, pages 113–125. ACM, 2015.

[262] C. E. Shannon. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55, 2001.

[263] N. Shibata, Y. Kajikawa, and I. Sakata. Link prediction in citation networks. *Journal of the American Society for Information Science and Technology*, 63(1):78–85, 2012.

[264] D. Shrier, W. Wu, and A. Pentland. Blockchain & infrastructure (identity, data security). 2016.

[265] R. Sibson. Slink: an optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, 16(1):30–34, 1973.

[266] A. Singla and A. Krause. Incentives for privacy tradeoff in community sensing. In *First AAAI Conference on Human Computation and Crowdsourcing*, 2013.

[267] A. Singla and A. Krause. Truthful incentives in crowdsourcing tasks using regret minimization mechanisms. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1167–1178. ACM, 2013.

[268] C. Song, T. Koren, P. Wang, and A.-L. Barabasi. Modelling the scaling properties of human mobility. *Nature Physics*, 6, 2010.

[269] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási. Limits of predictability in human mobility. *Science*, 327(5968):1018–1021, 2010.

[270] H. S. Song, J. kyeong Kim, and S. H. Kim. Mining the change of customer behavior in an internet shopping mall. *Expert Systems with Applications*, 21(3):157–168, 2001.

[271] S. Spiegel, J. Clausen, S. Albayrak, and J. Kunegis. Link prediction on evolving data using tensor factorization. In *New Frontiers in Applied Data Mining*, pages 100–110. Springer, 2012.

[272] M. Steinbach, G. Karypis, V. Kumar, et al. A comparison of document clustering techniques. In *KDD workshop on text mining*, volume 400, pages 525–526. Boston, 2000.

[273] K. W. Steininger and G. Bachner. Extending car-sharing to serve commuters: An implementation in austria. *Ecological Economics*, 101:64–66, 2014.

[274] X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009:4, 2009.

[275] P.-N. Tan, M. Steinbach, V. Kumar, et al. *Introduction to data mining*, volume 1. Pearson Addison Wesley Boston, 2006.

[276] R. F. Teal. Carpooling: who, how and why. *Transportation Research Part A: General*, 21(3):203–214, 1987.

[277] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical dirichlet processes. *Journal of the american statistical association*, 101(476), 2006.

[278] M. Terrovitis and N. Mamoulis. Privacy preservation in the publication of trajectories. In *The Ninth International Conference on Mobile Data Management (mdm 2008)*, pages 65–72. IEEE, 2008.

[279] R. Tijdeman. The chairman assignment problem. *Discrete Mathematics*, 32(3):323–330, 1980.

[280] J. L. Toole, C. Herrera-Yaqüe, C. M. Schneider, and M. C. González. Coupling human mobility and social ties. *Journal of The Royal Society Interface*, 12(105):20141128, 2015.

[281] J. L. Toole, Y.-R. Lin, E. Muehlegger, D. Shoag, M. C. Gonzalez, and D. Lazer. Tracking employment shocks using mobile phone data. *arXiv preprint arXiv:1505.06791*, 2015.

[282] C. A. Tovey. Tutorial on computational complexity. *Interfaces*, 32(3):30–61, 2002.

[283] L. H. Tran, M. Catasta, L. K. McDowell, and K. Aberer. Next place prediction using mobile data. In *Proceedings of the Mobile Data Challenge Workshop (MDC 2012)*, number EPFL-CONF-182131, 2012.

[284] R. Trasarti, R. Guidotti, A. Monreale, and F. Giannotti. Myway: Location prediction via mobility profiling. *Information Systems*, 2015.

[285] R. Trasarti, F. Pinelli, M. Nanni, and F. Giannotti. Mining mobility user profiles for car pooling. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1190–1198. ACM, 2011.

[286] L. H. Ungar and D. P. Foster. Clustering methods for collaborative filtering. In *AAAI*, volume 1, pages 114–129, 1998.

[287] G. Upton and I. Cook. *Understanding statistics.* Oxford University Press, 1996.

[288] D. Van den Poel and W. Buckinx. Predicting online-purchasing behaviour. *European Journal of Operational Research*, 166(2):557–575, 2005.

[289] D. Van den Poel and Z. Piasta. Purchase prediction in database marketing with the probrough system. In *Rough sets and current trends in computing*, pages 593–600. Springer, 1998.

[290] T. Vanoutrive, E. Van De Vijver, L. Van Malderen, B. Jourquin, I. Thomas, A. Verhetsel, and F. Witlox. What determines carpooling to workplaces in belgium: location, organisation, or promotion? *Journal of Transport Geography*, 22:77–86, 2012.

[291] T. Vanoutrive, L. Van Malderen, B. Jourquin, I. Thomas, A. Verhetsel, and F. Witlox. Carpooling and employers: a multilevel modelling approach. In *3rd Transport Research Day*, pages 335–349. VUB Press, 2009.

[292] A. Vázquez, J. G. Oliveira, Z. Dezsö, K.-I. Goh, I. Kondor, and A.-L. Barabási. Modeling bursts and heavy tails in human dynamics. *Physical Review E*, 73(3):036127, 2006.

[293] P. C. Verhoef and B. Donkers. Predicting customer potential value an application in the insurance industry. *Decision support systems*, 32(2):189–199, 2001.

[294] M. Vescovi, C. Moiso, M. Pasolli, L. Cordin, and F. Antonelli. Building an eco-system of trusted services via user control and transparency on personal data. In *Trust Management IX*, pages 240–250. Springer, 2015.

[295] M. Vescovi, C. Perentis, C. Leonardi, B. Lepri, and C. Moiso. My data store: toward user awareness and control on personal data. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, pages 179–182. ACM, 2014.

[296] N. X. Vinh, J. Epps, and J. Bailey. Information theoretic measures for clusterings comparison: is a correction for chance necessary? In *ICML*, pages 1073–1080. ACM, 2009.

[297] D. Wang, D. Pedreschi, C. Song, F. Giannotti, and A.-L. Barabasi. Human mobility, social ties, and link prediction. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1100–1108. ACM, 2011.

[298] K. Wang, C. Xu, and B. Liu. Clustering transactions using large items. In *Proceedings of the eighth international conference on Information and knowledge management*, pages 483–490. ACM, 1999.

[299] P. Wang, J. Guo, et al. Modeling retail transaction data for personalized shopping recommendation. In *CIKM*, pages 1979–1982, 2014.

[300] Y. Z. Y. S. Y. Wang. Nokia mobile data challenge: Predicting semantic place and next place via mobile data. *Work*, 80(100):120, 2012.

[301] M. Wedel and W. A. Kamakura. *Market segmentation: Conceptual and methodological foundations*, volume 8. Springer Science & Business Media, 2012.

[302] S.-S. Weng and M.-J. Liu. Feature-based recommendations for one-to-one marketing. *Expert Systems with Applications*, 26(4):493–508, 2004.

[303] N. Wilson, K. Mason, M. Tobias, M. Peacey, Q. Huang, and M. Baker. Interpreting google flu trends data for pandemic h1n1 influenza: the new zealand experience. *Euro surveillance: bulletin européen sur les maladies transmissibles= European communicable disease bulletin*, 14(44):429–433, 2008.

[304] Y. Xiao and M. H. Dunham. Interactive clustering for transaction data. In *International Conference on Data Warehousing and Knowledge Discovery*, pages 121–130. Springer, 2001.

[305] T. Xiong, S. Wang, A. Mayers, and E. Monga. Dhcc: Divisive hierarchical clustering of categorical data. *Data Mining and Knowledge Discovery*, 24(1):103–135, 2012.

[306] G. Xue, Y. Luo, J. Yu, and M. Li. A novel vehicular location prediction based on mobility patterns for routing in urban vanet. *EURASIP Journal on Wireless Communications and Networking*, 2012(1):1–14, 2012.

[307] K. Yada, H. Motoda, T. Washio, and A. Miyawaki. Consumer behavior analysis by graph mining technique. *New Mathematics and Natural Computation*, 2(01):59–68, 2006.

[308] H. Yan, K. Chen, and L. Liu. Efficiently clustering transactional data with weighted coverage density. In *CIKM*, pages 367–376. ACM, 2006.

[309] H. Yang and H.-J. Huang. Carpooling and congestion pricing in a multilane highway with high-occupancy-vehicle lanes. *Transportation Research Part A: Policy and Practice*, 33(2):139 – 155, 1999.

[310] N. Yang, X. Kong, F. Wang, and P. S. Yu. When and where: Predicting human movements based on social spatial-temporal events. *arXiv preprint arXiv:1407.1450*, 2014.

[311] Y. Yang, X. Guan, and J. You. Clope: a fast and effective clustering algorithm for transactional data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 682–687. ACM, 2002.

[312] Y. Yang and B. Padmanabhan. Ghic: A hierarchical pattern-based clustering algorithm for grouping web transactions. *IEEE Transactions on Knowledge and Data Engineering*, 17(9):1300–1304, 2005.

[313] A. Yassine and S. Shirmohammadi. Privacy and the market for private data: a negotiation model to capitalize on private data. In *2008 IEEE/ACS International Conference on Computer Systems and Applications*, pages 669–678. IEEE, 2008.

[314] G. Yavaş, D. Katsaros, Ö. Ulusoy, and Y. Manolopoulos. A data mining approach for location prediction in mobile environments. *Data & Knowledge Engineering*, 54(2):121–146, 2005.

[315] J. J.-C. Ying, W.-C. Lee, T.-C. Weng, and V. S. Tseng. Semantic trajectory mining for location prediction. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 34–43. ACM, 2011.

[316] H. Yoon, Y. Zheng, X. Xie, and W. Woo. Smart itinerary recommendation based on user-generated gps trajectories. In *Ubiquitous Intelligence and Computing*, pages 19–34. Springer, 2010.

[317] P. S. Yu. Data mining and personalization technologies. In *Database Systems for Advanced Applications, 1999. Proceedings., 6th International Conference on*, pages 6–13. IEEE, 1999.

[318] J. Yuan, Y. Zheng, X. Xie, and G. Sun. Driving with knowledge from the physical world. In *KDD*, pages 316–324. ACM, 2011.

[319] C.-H. Yun, K.-T. Chuang, and M.-S. Chen. Adherence clustering: an efficient method for mining market-basket clusters. *Information systems*, 31(3):170–186, 2006.

[320] M. J. Zaki and M. Peters. Clicks: Mining subspace clusters in categorical data via k-partite maximal cliques. In *21st International Conference on Data Engineering (ICDE'05)*, pages 355–356. IEEE, 2005.

[321] E. Zerubavel. *The seven day circle: The history and meaning of the week*. University of Chicago Press, 1989.

[322] T. Zhang, R. Ramakrishnan, and M. Livny. Birch: A new data clustering algorithm and its applications. *Data Mining and Knowledge Discovery*, 1(2):141–182, 1997.

[323] Y. Zhang and J. R. Jiao. An associative classification-based recommendation system for personalization in b2c e-commerce applications. *Expert Systems with Applications*, 33(2):357–367, 2007.

[324] N. Zhao, W. Huang, G. Song, and K. Xie. Discrete trajectory prediction on mobile data. In *Web Technologies and Applications*, pages 77–88. Springer, 2011.

[325] W. X. Zhao, J. Jiang, J. Weng, J. He, E.-P. Lim, H. Yan, and X. Li. Comparing twitter and traditional media using topic models. In *Advances in Information Retrieval*, pages 338–349. 2011.

[326] V. W. Zheng, Y. Zheng, X. Xie, and Q. Yang. Collaborative location and activity recommendations with gps history data. In *Proceedings of the 19th international conference on World wide web*, pages 1029–1038. ACM, 2010.

[327] C. Zhou, D. Frankowski, P. Ludford, S. Shekhar, and L. Terveen. Discovering personal gazetteers: an interactive clustering approach. In *Int. workshop on Geographic information systems*, pages 266–273. ACM, 2004.