IMT Institute for Advanced Studies, Lucca

Lucca, Italy

Big Data, Complex Networks and Markets

PhD Program in Computer Science and Engineering

XXV Cycle

By

Diego Pennacchioli

2014

“I’ve missed more than 9,000 shots in my career.
I’ve lost almost 300 games.
26 times I’ve been trusted to take the game-winning shot and missed.
I’ve failed over and over and over again in my life.
And that is why I succeed.”

Michael Jeffrey Jordan
in “Nike Culture: The sign of the Swoosh” (1998)
by R. Goldman and S. Papson, p. 49

The dissertation of Diego Pennacchioli is approved.

Program Coordinator: Prof. Rocco De Nicola, IMT Institute for Advanced Studies, Lucca

Supervisor: Prof. Fosca Giannotti, National Research Council of Italy, Pisa

Supervisor: Prof. Dino Pedreschi, University of Pisa

Supervisor: Prof. Massimo Riccaboni, IMT Institute for Advanced Studies, Lucca

Tutor: Dr. Alberto Lluch Lafuente, IMT Institute for Advanced Studies, Lucca

The dissertation of Diego Pennacchioli has been reviewed by:

Prof. Fabrizio Lillo, Scuola Normale Superiore, Pisa, Italy

Prof. Arno Siebes, Universiteit Utrecht, The Netherland

IMT Institute for Advanced Studies, Lucca

2014

This thesis represents the final step of a period full of personal and professional satisfactions and difficulties. Naming all persons who had a role in this result would require another book, so I will just recall the most directly “connected” to this achievement.

First of all, I need to thank Fosca Giannotti, Dino Pedreschi and Massimo Riccaboni for their precious work during my Ph.D. period. They helped me in finding interesting problems, guiding me to their solutions, always encouraging me to give the best of myself.

Among all my co-authors, a special thanks goes to Michele Coscia: he taught me how to write papers and how to be a better scientist, you have all my gratitude.

My gratefulness goes also to the reviewers of the thesis for their useful and positive comments: Arno Siebes and Fabrizio Lillo.

Moreover, I need to mention Michele Berlingerio and Anna Monreale. They, some time ago, gave me a very important speech. You know what I mean, guys. Thank you.

In order to arrive at the destination, in a trip like the Ph.D., the companions are very important, and at KDD Lab I met very special fellows. Thanks to Fabio, Chiara F, Chiara R, Salvo, Roberto, Mirco and Lorenzo (if you have a scientific or administrative problem, certainly one of these guys can solve it); Luca and Giulio (we shared the same difficulties of being

a PhD student at the same time, and I guess we had a good result); and, last but not least, Francesca, Letizia and Riccardo (new arrivals, but already old friends).

Thanks also to Mario, I had with him a lot of talks at home about my work, and he always gave me good hints.

I spent four months of my life at Northeastern University, and I need to gratefully thank Prof. Alessandro Vespignani, for accepting me and giving me the possibility to have that beautiful experience.

I'm grateful to Walter Fabbri and UNICOOP Tirreno for sharing with my lab their data and their precious suggestions about it.

The final, biggest, "thank you" goes to my family: my mother (Mamma), my father (Papà) and my aunt Ada. They always supported me and constantly make me a better person. Even if I'm not good in showing you my gratitude day by day, I'll always consider me as the luckiest guy in the world for having so special persons around.

This thesis is dedicated to Walter, Nonno Fiore and Nonna Elodia. I wish you were here.

Contents

List of Figures xiii

List of Tables xv

Acknowledgements xvii

Vita, Publications and Presentations xviii

Abstract xxi

1 Introduction 1

2 Related Works 6

I Building a Network over a Retail Market: the Purchase Behavior 13

3 Data 14

4 Discovering the General Pattern of Shopping Behavior 18

4.1 Transactional Data as a Complex System 21

4.2 The Framework 23

4.2.1 Preprocess 24

4.2.2 The f_* Function 27

4.2.3 Null Hypothesis 30

4.2.4 Product and Customer Sophistication 33

4.3 Experiments 35

4.3.1 Data Preprocessing 36
4.3.2 Framework Application 38
4.3.3 Hierarchy of Needs 41
4.3.4 Marketing 44
4.4 Conclusion 47

5 Explaining the Product Range Effect in Purchase Data 49
5.1 Data Selection 51
5.2 The Range Effect 52
5.3 Explaining the Range Effect 57
5.3.1 Sophistication and Range 57
5.3.2 Customer Behavior 59
5.4 Customer Behavior Prediction 61
5.5 Conclusion 65

6 A Complex Network Approach to Marketing Classification and Customer Profiling: Overlap versus Partitioning 66
6.1 Data Preparation 68
6.2 Community Discovery 71
6.2.1 Partition Approach 71
6.2.2 Overlapping Approach 73
6.3 Case Study 75
6.3.1 Community Size 76
6.3.2 Community Entropy 77
6.3.3 Community Extracts 79
6.3.4 Community Interpretation 82
6.4 Conclusion 83

II Exploiting the Social Networks for Applications in Markets 84

7 "You Know Because I Know": a Multidimensional Network Approach to Expert Finding Problem in the Market of Skills 85
7.1 Network-Based Human Resources 88
7.1.1 Problem Definition 88

7.1.2 The Model 88
7.1.3 The Data 91
7.2 The UBIK Algorithm 93
7.2.1 Time Complexity 95
7.3 Experiments 96
7.3.1 Comparison with other methods 97
7.3.2 Rankings 102
7.4 Conclusion 105

8 The Three Dimensions of Social Influence 107
8.1 Leader Detection 110
8.2 Measures 114
8.2.1 Definitions 114
8.2.2 Examples 115
8.3 Experiments 116
8.3.1 Data 116
8.3.2 Characterization of the Measures 120
8.3.3 Case Study 123
8.4 Conclusion 130

9 Conclusion and Future Works 132

References 135

List of Figures

3.1 The location of stores and customers over the Italian territory. 15
3.2 The conceptual data model (star schema) of the data warehouse. 16

4.1 A representation of the bipartite purchase graph, connecting customers to the product they buy. 22
4.2 The cumulative degree distributions of products and customers in the bipartite network. 23
4.3 The M_{cp} purchase matrix. 24
4.4 One instance of the null purchase matrix. 31
4.5 Customer distribution per city in our dataset. 37
4.6 The M_{cp} matrices for Livorno2010-2011 (top) and Lazio2007-2009 (bottom). 38
4.7 Our data-driven pyramid of needs. 42

5.1 The purchase matrix M_{cp}. 53
5.2 Average distance traveled to get a product with a given price. 54
5.3 Average distance traveled to get a product with a given popularity. 56
5.4 Sophistication and customer behavior against the distance from the shop. 56
5.5 The customer-shop distribution for customers buying different products. 60

5.6 Lift charts showing the increase in predicting performance obtained using product sophistication. 62

6.1 Example of the partition approach to community discovery. 73
6.2 Example of the overlap approach to community discovery. 75
6.3 The log binned distribution of the number of nodes per community, both for the partition and the overlapping approach. 76
6.4 The log binned distribution of the entropy per community, both for the partition and the overlapping approach. 78
6.5 Extracts from the non-overlapping communities. 80
6.6 Extracts from the overlapping communities. 81

7.1 Toy example of network and skills of the nodes. 90
7.2 Running times in milliseconds for different random networks with given number of edges, dimensions or skills. . 96
7.3 The q-q plots of various ranking algorithms against the ranking obtained ordering the nodes by degree. 99

8.1 Toy example for leader detection 110
8.2 Log-binned distribution of the nodes' degree, with power law fitting. 117
8.3 Log-binned distribution of number of listeners per artist, with truncated power law fitting. 118
8.4 The runtimes of the EXTRACTLEADERS procedure with growing number of nodes and objects. 119
8.5 Log-binned distribution of the relationship between Depth and Strength. 122
8.6 Log-binned distribution of the size of the Leader's Minimum Diffusion Tree ($T_{l,\psi}$) per tag. 124
8.7 The centroids of our clusters. 125
8.8 Distribution of number of objects (left) and of tags (right) per leader. 127
8.9 Distribution of leaders' Degree, Closeness Centrality, Clustering and Neighbor Degree per tag. 128

List of Tables

4.1 The $N(M_{cp}, f_*)$ for the different f_* tested. 30

4.2 The $N(M_{cp}, f_*)$ for the different views of the dataset. . . . 39

4.3 A selection of the more basic products according to their PS values. 40

4.4 A selection of the more sophisticated products according to their PS values. 40

4.5 The probabilities of buying product p_i in general ($P(p_i)$) and given that a customer already buys product p_{i-1} ($P(p_i|p_{i-1})$). 45

4.6 The comparison between the size of the target customer sets identified by the f_* function against random target customer sets with the same number of customers likely to buy p_i. 46

5.1 A selection of the more basic products according to their PS values. 53

5.2 Average PS values and average customer distance for the shops in our dataset. 60

5.3 The distribution per shop of the filtered dataset for the classifier. 63

6.1 Number of edges in the product network after filtering with minimum absolute support and lift. 70

6.2 Number of nodes in the product network after filtering with minimum absolute support and lift. 70

7.1 The statistics of the extracted networks 92
7.2 The top 15 researchers according to different ranking criteria. 99
7.3 The share of high clustering nodes in the top rankings per algorithm. 100
7.4 The top 10 researchers according to the general UBIK ranking and their rank for 10 different skills. 101
7.5 The top 10 employees according to the UBIK for the weekday and the weekend variants of the ranking. 104
7.6 The top authors for some conferences (taken singularly) comparative rankings. 105

8.1 Pearson correlation coefficient ρ between Width, Depth, Strength and other network statistics for our leaders. . . . 119
8.2 Partial correlation and p-value of Clustering and Closeness Centrality with Width, controlling for Degree values. . . . 122
8.3 The RCA scores of the presence of each tag in each cluster. 124
8.4 Presence of different diffusion patterns per tag. 128

Acknowledgements

Part of the material presented in this thesis has been previously published in several co-authored papers.

In particular, Chapter 4 is based on [101] and [98], joint works with Michele Coscia (Harvard Kennedy School), Dino Pedreschi (University of Pisa), Salvatore Rinzivillo and Fosca Giannotti (ISTI CNR, Pisa). Chapter 5 is instead based on [100], co-authored by Michele Coscia, Salvatore Rinzivillo, Dino Pedreschi, and Fosca Giannotti. Chapter 6 is based on [99], co-authored by Michele Coscia and Dino Pedreschi. Again, Chapter 7 is based on [36], co-authored by Michele Coscia, Fosca Giannotti, Damiano Ceccarelli (Navionics, Lucca), and Giulio Rossetti (ISTI CNR, Pisa). Finally, Chapter 8 is based on [102], co-authored by Michele Coscia, Fosca Giannotti, Dino Pedreschi, Giulio Rossetti, and Luca Pappalardo (University of Pisa).

Vita

August 25, 1982	Born, Lanciano, Italy.
September 2001 - July 2005	Bachelor Degree in Economics, University “G. D’annunzio” of Chieti, Italy,
September 2005 - December 2008	Master Degree in Business Informatics, University of Pisa, Final mark 110/110 cum laude
March 2009 - February 2010	Research Grant, ISTI,KDD Lab, National Research Council, Pisa.
March 2010 - Now	PhD Student, IMT Lucca, Italy.
March 2010 - Now	Research Associate, ISTI,KDD Lab, National Research Council, Pisa.
February 2013 - May 2013	Visiting Student, MOBS Lab, Northeastern University. Boston,MA (USA)

Publications

Conference Papers:

1. Michele Coscia, Giulio Rossetti, Diego Pennacchioli, Damiano Ceccarelli, Fosca Giannotti: *"You Know Because I Know": a Multidimensional Network Approach to Human Resources Problem.* In Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining;
2. Diego Pennacchioli, Giulio Rossetti, Luca Pappalardo, Dino Pedreschi, Fosca Giannotti, Michele Coscia: *The Three Dimensions of Social Prominence*. In Proceedings of Social Informatics - 5th International Conference, SocInfo 2013, Kyoto, Japan
3. Diego Pennacchioli, Michele Coscia, Salvatore Rinzivillo, Dino Pedreschi, and Fosca Giannotti: *Explaining the Product Range Effect in Purchase Data*. In Proceedings of 2013 IEEE International Conference on Big Data (IEEE Big Data 2013)
4. Diego Pennacchioli, Giulio Rossetti, Luca Pappalardo, Dino Pedreschi, Fosca Giannotti, and Michele Coscia: *The patterns of musical influence on the Last.FM social network*. 22nd Italian Symposium on Advanced Database Systems. Discussion Paper

Workshop papers:

5. Diego Pennacchioli, Michele Coscia, and Dino Pedreschi: *Overlap Versus Partition: Marketing Classification and Customer Profiling in Complex Networks of Products*. BDCA 2014 (Big Data Customer Analytics), in conjunction with ICDE 2014;

Technical reports:

6. Diego Pennacchioli, Michele Coscia, Fosca Giannotti, and Dino Pedreschi - *Calculating Product and Customer Sophistication on a Large Transactional Dataset*;
7. Michele Coscia, Giulio Rossetti, Diego Pennacchioli, Damiano Ceccarelli, and Fosca Giannotti - *"You Know Because I Know": a multidimensional network approach to human resources problem*

Poster Abstracts:

8. Diego Pennacchioli, Giulio Rossetti, Michele Coscia, Dino Pedreschi, and Fosca Giannotti: *Exploring the Evolution of the Product Space of Supermarket*. NetSci 2012, Evanston,IL;

Abstract

This thesis is focused on the study of new techniques of analysis coming from diverse fields (Complex Networks Analysis, Data Mining, and Big Data). Main aim is to better understand systems characterized by a high level of complexity. Markets are the chosen application scenario. In these complex systems, to find the right balance between the forces of demand and supply is very challenging, especially considering that they are characterized by imperfect but massive and fast information. In this context, the thesis presents approaches to face several open questions: how to find the general pattern of shopping behavior, how to mine the product space to find the best product/service that meets the demand, what is the role of the social influence between customers, and so on. The methods and techniques, belonging to the field of Complex Networks Analysis, are complementary to the usual ones of Data Mining. While in Data Mining the purpose is to search patterns and special distributions in a large dataset, here the purpose is to give a focus to the relations between entities of the markets, looking more to the whole system than to the single behavior. The thesis, finally, presents results of experiments performed on real world high quality datasets, providing, in addition to the theoretic results, practical application scenarios.

Chapter 1

Introduction

The increasing efficiency and availability of information and communication technologies in the last decade brought all actors in markets to focus their attention in the use and management of information. This task became of primary matter for them in order to protect and develop their business.

There exist a whole discipline, indeed, that study methods and measures to extract useful information from large repositories of data: the data mining. Unfortunately, nowadays the simple extraction of information from "*static*" datasets is not enough. There is the need to have a dynamic framework, that not only captures behaviors and pattern from the past, but that can study relations among entities, in order to early discover how and when these behaviors change. In other words, there is the need to study the network of entities that is established in markets. For this reason, the main aim of this thesis is to apply advanced methods and techniques belonging to *complex networks analysis* on *big data* to discover knowledge useful to better understand *markets*.

First, it is important to define precisely what we mean with big data, complex networks and markets:

- with **big data** we mean the ensemble of huge amount of data coming from different sources (data warehouses, query logs, online social networks, and so on) and optimized algorithms, methods and data structures able to manage it.

- with **complex networks** we refer to all these methods and measures of analysis developed in the recent years with the aim of studying and better understand real complex systems representing them with the mathematical model of graph. In this way we can model and study complex interacting phenomena such as social interactions among human beings, cause-and-effect phenomena in technological systems, biological reactions in organisms and so on.
- with **markets** we mean the actual or nominal place where forces of demand and supply operate, and where buyers and sellers interact to trade goods, services, contracts or instruments, for money or barter. In particular, we are interested in imperfect markets; or, better, in the following two features: (i) the object of the trade (goods, services, and so on) aren't homogeneous, and (ii) there is no single price in the market.

The problem of extracting information from data in order to better understand markets is one of the main application fields in data mining. Problems like churn and retention analysis, market basket analysis and customer segmentation represents natural application of standard methods of the discipline. These three applications are, in a certain sense, the primordial need for market actors to start understanding in a semi-automated way their business. With churn analysis we refer to the activity with the objective to predict when the loyalty of a certain customer is decreasing [8, 62, 124]. The objective of market basket analysis, instead, is to find recurring patterns over the sales data that a significant amount of customers have in common [14]. The customer segmentation, finally, has the objective to split customers in groups with a certain characteristic in common, usually linked to the purchase behavior [63, 70, 125]. The algorithms and methods that the three applications above use are all well known in data mining literature, in particular we refer to frequent [1, 2, 85, 95] and sequential [3, 116] pattern mining, classification

[17, 66, 105], clustering [10, 15, 40, 84], and so on.

This very powerful collection of methods has, unfortunately, one important downside: to perform it the analyst always needs to project over one *dimension*, losing important information about the composition of the other one. More precisely, by projecting on one dimension, one can lose coherence or contingency. For example, we can perform market basket analysis searching for frequent patterns among baskets, but we are not considering the dimension of the customer (we are losing in coherence, splitting the set of baskets belonging to the same customer and considering them as completely unrelated). To solve this problem, we can perform the analysis over the whole set (or an approximation) of purchases of each customer, but in that case we are representing a customer with only one purchase behavior, while he/she can behave in several different ways (in this case, we lose in contingency).

In addition, there is another important factor brought from the exponential increase of availability and cheapness of *ICT*: the enhancing of social interactions and information spreading that makes the above mentioned kind of analysis less useful, at least in these three scenarios:

- due to online social networks, dedicated forums, and special search engines, the spread of information in the market became almost instantaneous, and this phenomenon affects the analysis performed *statically* making the assumption done on the customers' vision of the world quickly obsolete (anyone has access at hundreds of "suggested prices" or at thousand of reviews and opinions for each different product)

- the increased level of social interaction between people enhanced the importance of the "social factor" in the choice function of a product/service (while before the choice function was dominated by the "personal factor"). In addition, automatic recommendation systems give even more priority at what our connections and people similar to us choose. This means on one hand that we can observe flows of changes of behavior (due to the chains of influence that the so called "leaders" have on the other people), and so the analysts need to tackle this continuous change of the customers' purchase behavior; but on the other hand in this scenario the idea of targeted marketing become achievable: by identifying the leaders of any bunch of products, the seller would just needs to advertise few customers in order to spread the information about new or interesting products throughout his set of customers.

- the immediacy with which each person can communicate with his/her "*neighbors*" changed also in a radical way the same concept of person, that is not only a container of personal behaviors and knowledge, but can be seen also as a combination of the previous personal factor plus behaviors and knowledge that its connections transmit to him/her. This phenomenon, in a system where people are the object of the market (e.g. job markets), has a very large impact. The usual mining techniques that just investigate the personal sphere of skills and features are not enough, and there is the need of a methodology to better evaluate the supply in the market.

In some of the above cases, the goals of the analysis can be achieved modeling the interplay between actors in markets and studying the properties of these relations (first two scenarios) or using novel algorithms able to mine information taking into consideration both static features and relations among entities (third scenario).

The thesis is divided in two parts: in part I we focus the attention on retail market, where we try to build relations among customers and products, in order to easily perform analysis considering both of them. After a presentation of the dataset (we have access at the selling data of one of the largest Italian retail company), we talk about the customer behavior. By building a bi-partite graph customer-product (see Chapter 4), it is possible to notice an interesting scheme of the purchases (modeled with the links between customers and products), called *triangular structure*. This structure allows to calculate a new measure called "sophistication degree", that is a new interesting measure that finds ground of application in psychology (Section 4.3.3), targeted marketing (Section 4.3.4) and geomarketing (Section 5). In Chapter 6 we show how, even focusing on just one dimension (in the case the product), a network approach can give interesting results not achievable with usual data mining tools.

In Part II, instead, we study the role of social networks in markets. Here the main differences with respect to the previous Part are that (i) the nodes are always people, and (ii) the links between people are explicit (or, better, explicitly reported by them). As we saw, markets are complex systems where people can have different roles: one can be the object of the trade (e.g. market of jobs) or an actor (e.g. a customer in retail market). In Chapter 7 we discuss the first case, showing a way to better rank and find people (modeled as a container of skills) in an expert finding application, where we take into consideration not only the personal dimension of skills, but also the interconnection that one can have with its personal network (and with skills not directly possessed, but easily accessible by him/her). In Chapter 8, instead, we consider another very important phenomenon in trade of goods: the influence that people have on other people's choice about products and services. More generally, we can say that the experience of a product/service made by a friend (or an acquaintance) we trust can be a very strong added value (sometimes more than the real features/conditions of the good/service). Going up through the chain of influence, we can find the root (or *leader*), opening the doors at the real dream of any communication strategist: exploit the word of mouth to let a new product/service reach all customers focusing the resources just on the leaders. In this section, we provide a real example based on the music listening made on Last.FM social network.

Finally, Chapter 9 concludes the thesis, by presenting the future research directions opened by this study.

Chapter 2

Related Works

One of the purposes of this thesis is to find interesting findings on markets starting from complex networks analysis and big data methods. The most efficient and interesting algorithms in literature belong to data mining field. For example, in Chapter 4 is described a complementary approach to the classical data mining task of the association rule mining. In data mining, association rule mining is a tool developed to find correlations between the appearances of products in shopping carts [1]. Association rule algorithms are able to uncover the most frequent and interesting rules by efficiently cutting the search space (or even without [30]). Recently, many step forwards have been proposed in association rule mining as mining multidimensional rules [92]. When we try to find the general pattern of shopping behavior, we differ from the works presented as we are not focused on finding all the particular rules in a transactional dataset, but in exploring the pattern characterizing it as a whole. This pattern can also be used to design better heuristics for the classical association rule mining algorithm, since it unveils novel relationships among products.

There are also works that aim to use association rule mining to obtain a general picture of the system [24]. However, also in this case our approach is different. In [24], only the associations between products are considered, leaving the customers undescribed. Then, the general picture in [24] is based on the aggregation of the local patterns, while in our paper we employ a complementary approach, creating the general picture by analyzing the entire set of transactions as a complex system, expressing properties at the global level that are not necessarily given by the sum of the properties at the local level.

Other relevant literature dealing with the problem of extracting knowledge from customer behavior can be found in business intelligence. In this field, many data mining and OLAP techniques have been developed, enriching the analytic tools [72, 79], not only for marketing purposes but also to detect frauds [47] or public health surveillance [121]. Data mining and customer behavior has gone also one step forward, by exploiting sentiment analysis as a prediction tool for a product success/failure [20].

Our approach is a combination of the application and the evolution of some tools present in literature. First, for some specific tasks during the whole Part I we make use of the lift measure. The lift (as the conviction, collective strength and many more) is one criterion used in association rule mining to evaluate the interestingness of a rule [48]. Second, we make use of concepts related to ecology literature [11] and macro economics [57, 58]. While in Chapter 4 and 5 we use similar techniques (as the eigenvector factorization of the customer-product matrix to calculate the sophistication levels of both customers and products), our paper differs from the ones presented on two axis: the first is the quality of the data to which we apply our framework (micro purchases against macro world trade or ecosystem presence/absence of animal species); the second is the quantity of data, as we work with matrices with a number of cells $\sim 10^9$ while related works do not scale beyond $\sim 10^5$ and therefore cannot be used with big data.
The customer behavior is not just driven by personal preferences and endogenous needs of the single persons. There are many other factors that affect the choice of products by the customers. One of these exogenous factors is the distance (in terms of space and time) to be traveled by the customer to the shop selling the product she wants to buy (see Section 5). Again, we need to refer to studies of customer behavior and data mining for marketing and for the analysis of spatial data.

The first field has been tackled mainly in the economics literature:

behavioral economics has focused its attention on the rational choices of the entities in the market [37, 41, 60]. Customer behavior study is also a classical marketing problem [117]. However, the recent studies about the customer movements and purchases are more focused on visually inspecting the movements of people inside a shop [89]. This line of research is present also on the computer science side [109]. In computer science, there are also examples of GIS approaches to business intelligence [103], of recommender systems for customer retention [52] and spatial analysis of customer-to-business communications [55]. In computer science [38] is a comprehensive book explaining the relations between economics and a network-based analysis.

Marketing applications are historically one of the most natural testing ground for data mining [1, 14]. The main aim of Part I, understanding the links between customers and products, has been deeply tackled in this discipline: by analyzing them in a multidimensional space [80], by mining frameworks to understand customer behavior [44, 75, 114] and by defining a data-driven customer segmentation [39]. However, these works have in common the aim of the specific description of single customers, rather than finding a broader and general pattern in the data. Data mining has been widely used also in other generic problems related to geographical systems. For example, a network mining approach has been used to detect the borders of human mobility [34], of tweet's topics [61] and trajectory pattern analysis [25, 83].

The bipartite graph customer X product is not the only way to model a market in order to be analyzed with complex networks analysis techniques. We can also think at a easier representation, like customers or products networks, for example in [59], the authors build a network connecting customers based on communication frequency, while one first attempt to build a network of products is in [24] (that is at the basis of the work descibed in Chapter 6), where the authors use a dataset coming from an university store over the time span of a year (660K transactions, 2200 products). Authors build a directed network (where relationships are not symmetrical) connecting product A to product B if B is frequently purchased when A is purchased. Their approach is based on association rules discovery [1]. Anyway, in [24] the authors are interested not at the composition of the communities, but at their overall quality, measured with an aggregation of the confidence attached to the edges.

Moving to the other central aspect of Chapter 6 , one classical problem definition in complex network analysis is how to detect functional modules in the network. This is usually known as "community discovery", borrowing from the social network literature [33]. Given the high relevance of this branch of studies, we examine this problem with higher depth in Section 6.2. In general, community discovery is a very popular research field in complex network analysis, with hundreds of papers on the topic and an almost equal number of developed algorithms [45]. The amount of relevant literature is due to the lack of a proper and unique definition of "community" [127] and the potential high impact of research in the field [42]. Historical approaches such as defining a particular quality function (like modularity) for community discovery or the detection of semi-cliques in the network (the k-clique percolation algorithm) have been widely used but are of no interest here given their theoretical downsides and/or their inability to scale for large networks [33]. Successful modern approaches can be divided in two classes, that we will explore in depth in Sections 6.2.1 and 6.2.2: partition-based algorithms such as infomap [110], agent-based [65] or label propagation based [108]; and overlap-based like DEMON [35], Hierarchical Link Clustering [4] and overlap label propagation [77, 126].

As stated above, markets are not only to be intended as retail markets. In Part II we discuss some application for markets where we can exploit the knowledge of the social relation among agents of the market. In Chapter 7 we present an application for markets of expertises and skills (or, more generally, for job markets), where there's a huge space for improvements in order to help human resources management people. This is not the first work pointing out that social network analysis is a useful tool in this area, as [16] starts from the assumption that the value of a person is not only determined by the extent of what she knows, but also to her position in a social network, that is exactly our starting point. However, [16] only uses a centrality concept, without looking at the skills of the nodes of the network, nor it builds a computer science framework to solve the problem. The social dimension enters in the problem of human resources management not only as an evaluation tool of the skills of a person, but also on how much she can influence the behaviors of other employees [43].

To rank nodes according to their importance in a network structure is a classical problem in complex network analysis [120]. The final aim here is to have multiple rankings over a multidimensional network. To the best of our knowledge, no algorithm is able to perform this task. The current state of the art in this research branch can be divided in four categories.

- In the first category, there are popular ranking algorithms (PageRank, SALSA and related [78, 93, 115]) which provide a single ranking in a monodimensional network.

- In the second category, we have ranking algorithms that can provide multiple rankings, but on a non-multidimensional network. The oldest and best-known approach is HITS [71], which provides only two rankings (hub and authorities). Another example is topic-sensitive PageRank [56], that can provide an arbitrary number of rankings. However, in the topic-sensitive PageRank there is no way to exploit different kinds of relations, so it cannot perform the task requested.

- In the third category, the ranking algorithms deal with multidimensional network, but they provide a simple ranking, much like the PageRank does for monodimensional networks. This branch has been thoroughly tackled in recent years, for example in [81, 91, 119].

- Finally, multidimensional approaches to the multi-ranking problem. There is already one method in literature in this category, namely TOPHITS [74]. However, just like HITS, TOPHITS provides only two different rankings, hubs and authorities, therefore its application to real world scenarios in the evaluation of many skills is questionable. The work described in Chapter 7 belongs to this category.

Another very important factor, talking about markets, is the spreading of information and influence among customers. The study of diffusion processes has reached, in the last decade, a very central role in complex systems analysis field, while before was especially a biological problem. This is due to the increasing complexity and density of personal and social links in society made possible by technology (cheaper and more accessible flights, online social network services, etc.), that provide: a) a very easy way for people to connect and spread information, and b) a collection of huge datasets to be analyzed by the data experts.

Two phenomena related to the networks are tightly linked to the concept of diffusion: the spread of biological [31] or computer [123] viruses, and the spread of ideas and innovation through social networks, the so-called "social contagion" [19]. In both cases, the patterns through which the spreading takes place are determined not just by the properties of the pathogen/idea, but also by the network structures of the population it is affecting.

In the SIR epidemic model [68] each individual transits between three stages in the life cycle of a disease: a node may pass from Susceptible (S) to Infected (I), and from Infected to Recovered (R). Initially, some nodes are assumed to be infected, while the others are susceptible to infection. During the infectious period, people can pass the disease to their neighbors with a certain probability. After a healing time, these nodes become inert in the contact network (recovered state). SIS models [96] allow to model illnesses able to infect people multiple times, with individuals alternating only between susceptible and infectious states. SIRS models [96] confer temporary but not permanent immunity on infected individuals. Indeed, a recovered person can return to a susceptible state after a certain amount of time. Recently, the availability of big data conveying information about human interactions and movements encouraged the production of more accurate data-driven epidemic models.

Colizza *et al.* [31] modeled the spatio-temporal diffusion of the avian pathogen H5N1 through a SIR-like model, exploiting the global air mobility network. For each city covered by an airport, they simulated the epidemic evolution by solving a set of differential equations, revealing how the strong small-world effect of air mobility networks makes air travel limitations useless. Wang *et al.* [123] analyzed a mobile phone dataset to study the spreading patterns that characterize a mobile virus outbreak. They predicted that, if the largest mobile OS reaches a critical point, a virus will be able to reach the entire network of phones.

Christakis and Fowler studied the role of social networks in the spread of obesity [27], smoking [28] and happiness [46]. Their results suggest that these health conditions may exhibit some amount of "contagion" in a social sense: although the dynamics of diffusion are different from the biological virus case, they nonetheless can spread through the underlying social network via the mechanism of social influence.

The main difference between biological and social contagion lies in the fact that while the transmission of a disease obeys mostly to deterministic mechanics, people do make conscious or unconscious decisions to adopt a new idea. Empirical evidences of this observation come from the observed tendency to adopt behaviors of neighbors, detected in several real-life [86, 111] and online [7, 9, 22] contexts. On the basis of this difference, different models from the ones presented before have to be adopted. In threshold models [54] a node decides to adopt an idea if a given fraction of his neighbors adopted it. In cascade models [50] an "active" individual has a single chance to activate or not each currently inactive neighbor. Some authors [67] proposed a general framework that simultaneously includes both kinds of models as special cases. They also provided a greedy solution for the NP-hard optimization problem of finding the best set of individuals for maximizing the cascade of innovation adoptions.

Part I

Building a Network over a Retail Market: the Purchase Behavior

Chapter 3

Data

The dataset we used is the retail market data of Coop, one of the largest Italian retail distribution company.

The whole dataset contains retail market data in a time window that goes from January 1st, 2007 to April, 7th 2013. The active and recognizable customers in that interval are $1,659,028$. 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. The 138 stores of the company cover an extensive part of Italy, selling a total of $2,732,531,951$ items. The geographical distribution of shops and customers is depicted in Figure 3.1.

The conceptual data model of the piece of the data warehouse meaningful for the purposes of this thesis storing the retail data is depicted in Figure 3.2, where the fact table (Sold) contains a row representing each single product scanned in each checkout point of the supermarket chain.

The information contained in the dimensions of the data warehouse is the following:

- **Item**: contains information about items: a natural language description, the brand, and a list of flag for "special" products (for coeliacs, organic, Protected Designation of Origin (PDO), Protected Geographical Indication (PGI), and so on). The table contains information about $439,619$ different products.

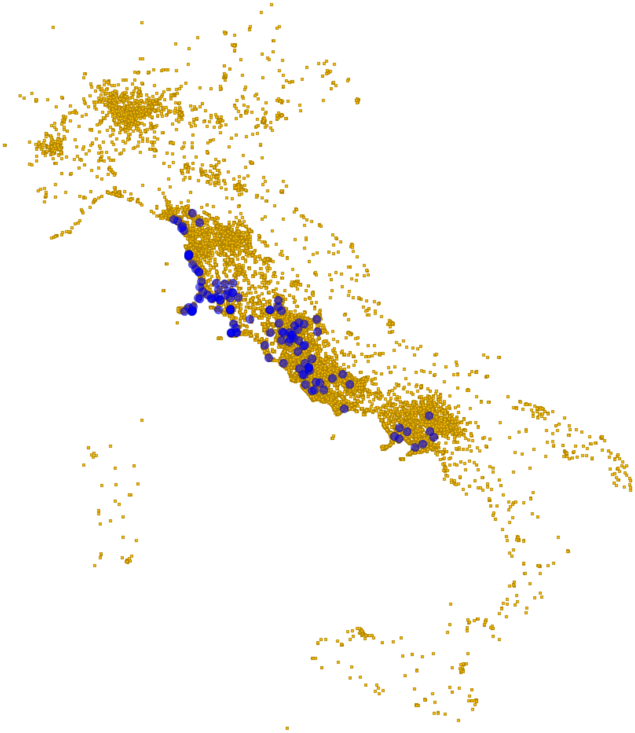


Figure 3.1: The location of stores (blue dots) and customers (yellow dots) over the Italian territory.

- **Customer**: contains information about customers: card number, name, date of birth, and geographical information about the residence (address and coordinates).
- **Store**: contains information about the stores. In particular, geographical informations (address and coordinates) and the kind of the store. A store can be classified as:
  - **Iper**: larger shops: they sell everything (food, cleaning product, but also TVs, scooters, appliances, and so on)
  - **Super**: medium size shops: they have a medium-high assort-

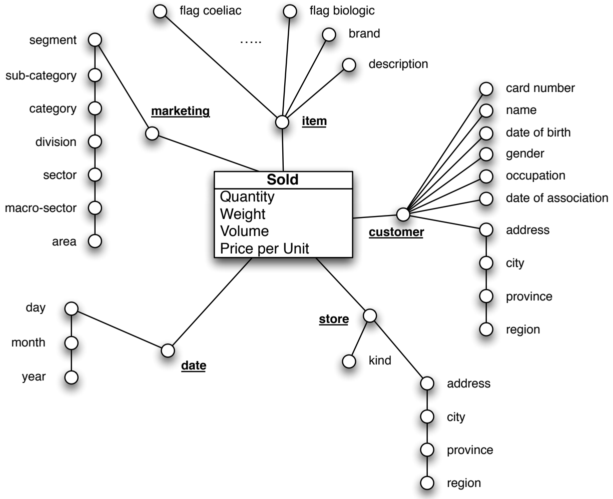


Figure 3.2: The conceptual data model (star schema) of the data warehouse.

ment (w.r.t. Ipers they don't sell appliances, TVs and expensive products)

– **GestIn**: smaller shops: they sell food products and just some not-food (house and body cleaning, cutlery, batteries, and so on)

- **Date**: contains information about the day of the year and the kind of the day (if it is a working day or not)

- **Marketing**: this dimension is used to classify products: it is organized as a tree and it represents a hierarchy built on the product typologies, designed by marketing experts of the company (see Figure 3.2 for a list of hierarchy levels). The top level of this hierarchy is called "Area" and it is split in three fundamental product areas: *Food*, *No Food* and *Other*. The bottom level of the marketing hierarchy, the one directly on top of the leaves of the tree, is called *Segment* and it contains 7,003 different values. Each item has a classification in this hierarchy and, thus, we can exploit such tree to choose the most suitable level of aggregation of products. The main difference between *item* level and *Segment* level consists in packaging, size and brand. For example, the three items: half-liter sugar free coca cola bottle, $6X1.5$ liter sugar free coca cola bottle's box, and two liters pepsi cola bottle, belong all at the same segment (sugar free cola drinks).

Chapter 4

Discovering the General Pattern of Shopping Behavior

This chapter is mainly based on [98] and [101].

Every day, millions of people use supermarkets and shops to fulfill their needs. They buy water, food, products for housework, electronic equipment, cars. Many of these purchases are electronically registered. The stores usually provide loyalty cards to their customers and they are able to connect their customers' purchases with a very high level of detail (i.e. any single item sold separately) to the data the customers had provided when obtaining the card. Thus, not only we have very large amount of data, but also the quality of the data is very rich. We can know everything about the single products, whether they are food or not, the brand, the price, the weight and so on; and we can know many things about the customers, their age, profession, when they usually purchase things.

In such a data rich environment, data mining arose as an useful tool to extract knowledge about customer behavior. Usually, the amount of data is huge. Very large databases and data warehouses are needed to handle this kind of data. In this setting, to apply standard statistical and analytic techniques is hopeless. The need for efficiently extract knowledge from these databases has favored the development of many techniques such as association rule mining [1], fast data clustering [118], OLAP techniques supporting business intelligence tasks [23, 72] and many more.

Many instances of useful knowledge have been extracted with these data mining techniques. However, so far the knowledge extracted was very specific and particular, and it lacked a general big picture about the data. Let us consider association rule mining. In association rule mining the general output is a rule stating that if a customer purchased itemset A (for example diapers) then he/she has also purchased itemset B (for example beer). The correlation between the appearances of the two itemsets is interesting, but it regards only the items included in them and the remaining of the purchase data have been left undescribed. If we describe the dataset with a bipartite network connecting customers to products, the classical association rule mining starts considering the degree of a product, i.e. its support. Then, products are grouped into sets maximizing the number of customers connected to each product of the set. However, rarely these product sets include more than ten products, while the entire bipartite network is composed by thousands products, and millions customers that are reduced to a number (the support value). The output of association rule mining is composed by thousands rules, each describing a single particle of the customer behavior, and selecting the most representative ones is usually a problem [64]. Moreover, usually many products are left undescribed, as they are not frequently purchased, causing this description to be incomplete.

In this section, we propose a methodology aiming at the description of this bipartite structure as a whole, representing a complementary alternative to the classical approach of association rule mining. We want to use the entire set of all customer-product connections to better understand the hidden knowledge governing the interplay between our desires and needs on one side, and the offered goods and products on the other side. Moreover, what we propose is not only a description of interesting product patterns, as in association rule mining, but also of customer patterns.

To do so, we implement a data analysis framework which mainly operates on the characteristics of the customer-product bipartite structure. This framework takes as input the bipartite customer-product network, it considers its adjacency matrix and it operates on this matrix with the aim of extracting its general defining pattern. The framework returns: (i) the characteristic function of the matrix, which divides the adjacency matrix in expressed and not expressed connections; and (ii) a ranking of both customers and products, which describes how much basic or sophisticated is a product, or the needs of a customer. We apply our framework on a unique transaction database, described in Section 4.2.1.

We propose it as a novel analytic paradigm for market basket analysis. This paradigm is challenging conceptually, because we are analyzing a complex system of millions of customers and thousands of products as a whole, in the search of its defining characteristics. Given the cardinality of the data, this novel paradigm is also challenging computationally.

Our results shows that the framework is able to exploit (and quantify) the defining characteristic of the customer-product matrix. Here, we show three possible results that can be extracted from transactional dataset applying our framework. First, we define mathematically the connection pattern between the volume of sales of products and customers. The potential applications of such a general pattern may represent an advancement in many fields: in data mining, as it could provide a new tool to mine product association rules; in economics, as it may represent a better description of economic dynamics; in marketing, as it could be used to define better viral and targeted promotion strategies; in social psychology, as it may constitute a data driven definition of the hierarchy of needs of large groups of people.

Second, we provide one empirical observation of Maslow's hierarchy of needs [87]. We find that we can measure how sophisticated are the shopping behaviors (and, indirectly, the needs) of customers, and the same holds for products. Highest ranked customers, with more sophisticated needs, tend to buy niche products, i.e., low-ranked products; on the other hand, low-ranked, low purchase volume customers tend buy only high-ranked product, very popular products that everyone buys.

Third, we propose a simple marketing application useful for targeted advertising. Given the characteristic function classifying the likelihood of a customer-product connection and a function characterizing how basic/sophisticated are the needs/characteristics of a customer/product, a target marketing campaign may spot more precisely the smallest customer set that is likely to start buying a given product.

To sum up, our contributions are the following. First, we develop an analytic framework able to analyze a transactional dataset as a whole, providing the general picture that association rule mining cannot define. Then, we apply this framework to a vast real world dataset, extracting some useful knowledge from it.

4.1 Transactional Data as a Complex System

Before building our framework we need to justify the choice on which it is founded. Our assumption is that the traditional data approach in association rule mining leaves undescribed a great amount of valuable data. This data can be described by looking at the entire transactional dataset as a complex system, and therefore it can be described as such. A complex system is a system of small units who influence each other. The complex system as a whole expresses properties and characteristics that are not embedded in its parts, but emerge only when looking at the global picture.

In Figure 4.1 we have a possible representation of the purchase data. It is a bipartite graph with two types of nodes: customers and products. Customers are connected to the products they buy. What association rule mining does is to count the in-degree of each product and remove the products with a degree lower than a given threshold. It then combines the remaining products in set of two (and more) products and removes the couples with insufficient in-degree until no further combination of product with sufficient degree can be found. At the end of the procedure, the result consists usually of thousands of sets of products.

However, this set of patterns cannot describe the entire dataset. Firstly,

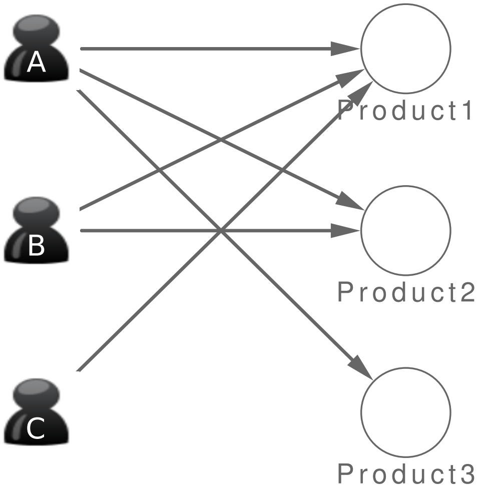


Figure 4.1: A representation of the bipartite purchase graph, connecting customers to the product they buy.

it excludes customers, that are used only for counting the support of products. Secondly, as many natural phenomena, also purchasing behavior is characterized by uneven distributions.

In Figure 4.2 we depict the two cumulative degree distributions of a transactional dataset, one for each node type: the products (left) and the customers (right). The data comes from one of the geographical areas of the supermarket dataset (namely Lazio), which will be described in details in Section 4.3. At the left we have a plot describing the probability (y axis) of a product being bought by at least a given number of customers (x axis), while at the right we have the probability (y axis) of a customer of buying at least a given number of products (x axis). We can see that the distribution in both cases is log-normal (please note the logarithmic x axis): the 20% of the least popular products are being bought only by at most 10 customers while the 20% most popular products are being bought by more than 100k customers (almost a million); the 20% of the customers bought 10 or less distinct products while 10% of the customers bought 1,000 distinct products or more.

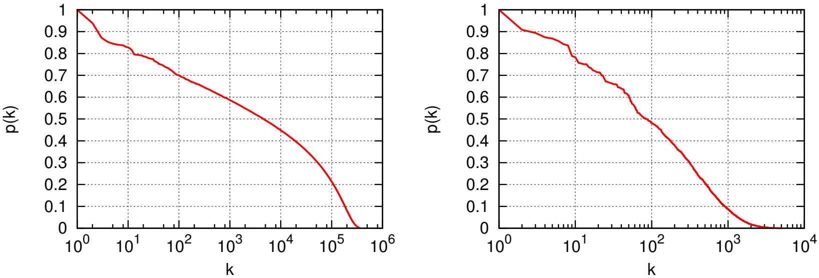


Figure 4.2: The cumulative degree distributions of products (left) and customers (right) in the bipartite network.

The consequence is that a large amount of products are not considered in association rule mining, simply because they fail to meet the support threshold requirement. Also, the customer degree distribution gives us a further intuition that will be confirmed in the following sections: the number of products that are really bought by almost everybody is very low (in the order of 10, we need more evidences of this as the distribution is an aggregation and the 10 products bought by the bottom 20% of the customers may be different, but we will see that they are not). In other words, the connections between the most popular products are not randomly distributed into the dataset, but they tend to be connected to the same set of customers, the ones that buy everything. This is an additional limitation of association rule mining: not only it ignores many products, but the products considered are always being bought by the same set of big buyers, leaving out different purchase behaviors. We can conclude that a methodology able to include customers and less popular products into a global picture can be useful as a complementary part of association rule mining.

4.2 The Framework

In this section we describe our framework. The input of the framework is the weighted bipartite graph $G = (C, P, E)$ connecting the customers

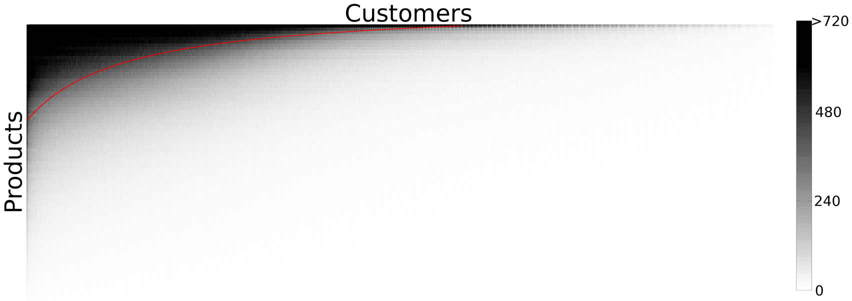


Figure 4.3: The M_{cp} purchase matrix. For layout purposes, the matrix has been transposed, thus we have customers as columns and products as rows. The red line is the isocline of the matrix (see Section 4.2.2). Color image.

$c \in C$ to the products $p \in P$ they buy. The weight w_i on the edge $(c_i, p_i, w_i) \in E$ is the number of times customer c_i bought product p_i. The output of the framework is a description of the global and the local properties of the bipartite structure. The global description is a function f_* connecting the volume of sales of products with the set of customers buying them and the volume of purchases of customers with the set of products they are buying. The local description is an evaluation of how much a product, and a customer's need, is basic or sophisticated and we call it product (or customer) sophistication. To go from the input to the output we need to apply a three-step process: (i) pre-process the data, by filtering out the not significant purchases (Section 4.2.1); (ii) evaluate the consistency of the data through a null model check (Section 4.2.3); (iii) apply two different algorithms to obtain f_* (Section 4.2.2) and the product/customer sophistication (Section 4.2.4).

4.2.1 Preprocess

To analyze the bipartite customer-product structure, we decide to deal with the adjacency matrix representing its connections. Since we want to analyze the aggregate behavior of customers and verify whether some patterns emerge on the relation between customers and products, we need to arrange the rows and the columns of the adjacency matrix in a logical way. Hence we sorted the matrix with the following criterion: fixing the top-left corner of the matrix M as the origin, we sorted the customers on the basis of the sum of the items purchased in descending order (the top buying customer at the first row and so on), and the products with the same criteria from left to right (the top sold product at the first column and so on). In this way, at the cell $(0, 0)$ we can find the quantity of top sold product purchased by the top buying customer. Using this criterion, we exploit the log-normal degree distributions of the bipartite structure, showing that the best sold products are bought by all kinds of customers, while products with a low market share are bought exclusively by customers who buy everything. This consideration is at the basis of the f_* function estimation in Section 4.2.2.

The final step of data preparation is to binarize the matrix, by identifying what purchases are significant and what are not. We cannot simply binarize the matrix considering the purchase presence/absence of a customer for a product. A matrix with a 1 if the customer c_j purchased the product p_i and 0 otherwise will result in a certain amount of noise: it takes only a single purchase to connect a customer to a product, even if generally the customer buys large amounts of everything else and the product is generally purchased in larger amount by every other customer.

We need a mechanism to evaluate how meaningful is a purchase quantity for each product p_i for each customer c_j. This evaluation is done using the concept of lift [1], that is related to association rule mining. Given a couple of itemsets (X, Y), the lift of the couple is defined as follows:

$$\text{lift}(X, Y) = \frac{\text{supp}(X, Y)}{\text{supp}(Y) \times \text{supp}(X)},$$

where $\text{supp}(I)$ is the relative support of the itemset I. The relative support of itemset I is the number of times all $i \in I$ are purchased together over all the transactions present in the dataset.

In our case, we force a particular condition: the itemset X always contains one item (the customer c_j); the itemset Y always contains one element (the product p) and the support of (c_j, p_i) is given by the corresponding entry in the matrix. In other words, $\mathrm{supp}(c_j, p_i)$ is the relative amount of product p_i bought by customer c_j, $\mathrm{supp}(p_i)$ is the relative amount sold of product p_i to all customers and $\mathrm{supp}(c_j)$ is the relative amount of products bought by customer c_j.

Lift takes values from 0 (when $\mathrm{supp}(c_j, p_i) = 0$, i.e. customer c_j never bought a single instance of product p_i) to $+\infty$. When $\mathrm{lift}(c_j, p_i) = 1$, it means that $\mathrm{supp}(c_j, p_i)$ is exactly the expected value, i.e. the connection between customer c_j and product p_i has the expected weight. If $\mathrm{lift}(c_j, p_i) < 1$ it means that the customer c_j purchased the product p_i less than expected, and viceversa. Therefore, the value of 1 for the lift indicator is a reasonable threshold to discern the meaningfulness of the quantity purchased: if it is strictly higher, then the purchases are meaningful and the corresponding cell in the binary matrix is 1; otherwise the purchases are not meaningful, even if some purchases are actually made, and the corresponding cell in the binary matrix is 0. The M_{cp} matrix is built accordingly to this rule:

$$M_{cp} = \begin{cases} 1 & \text{if } \mathrm{lift}(c_j, p_i) > 1; \\ 0 & \text{otherwise.} \end{cases}$$

This is the final output of the preprocess phase, hence from now on it will be referred as the purchase matrix and $M_{cp}(c_j, p_i)$ is the entry of M_{cp} of row j and column i. We provide an example of an M_{cp} matrix in Figure 4.3, that is the M_{cp} matrix extracted in the Livorno2007-2009 dataset (see Chapter 3). In Figure 4.3, the columns of the matrix are the $317,269$ customers and the rows are the $4,817$ products. We depicted a compressed view of the matrix, where each data dot represent a 50×50 square of the original matrix and the gray gradient represents how many $1s$ are present in that section of the matrix, for space constraints.

We can observe in Figure 4.3 the phenomenon we described in Section 4.1: only a small amount of popular products are bought by everyone, but a smaller and smaller set of customer purchases the rest of the products (going from the right to the left columns) and it is always composed by the same set of big buyers. This particular distribution of ones is used in Section 4.2.2 to define f_*. Before doing that, we need to verify if this particular distribution is telling something interesting or it is expected, and we tackle this problem in the next stage of the framework.

4.2.2 The f_* Function

Customers behavior is not random: as we have seen in Chapter 2 there are many studies dealing with the problem of finding correlations between products frequently bought together [1]. However, our framework is based on a stronger assumption than the presented consideration: it requires that these correlations are actually organized following a general law that regulates retail purchases. In other words, we are not dealing with a set of correlations limiting their effects on two or three products. There exists a general pattern, stating that it is possible to define the set of products bought by a customer as a function of the amount of products he/she buys.

In other words, the assortment of products bought by any given customer c_j is determined by c_j's volume of purchase, and the population of customers that buy any given product p_i is determined by p_i's volume of sales. More precisely, the f_* function that is being extracted relates the *rank* of products with the *rank* of customers, where the rank i of a product p_i (or j for customer c_j) stands for the fact that p_i is the i-th highest sold product (or c_j is the j-th customer with the largest volume of purchases). The function f_* is a systematic map from the rank j of a customer c_j to the rank $i = f_*(j)$ of a product p_i, such that the assortment of products bought by c_j is $\{p_1, \ldots, p_i\}$ with high probability. The mapping can be inverted, so we can map the rank i of any product p_i into the rank $j = f_*^{-1}(i)$ of a customer c_j.

We help to understand the intuition behind the f_* function using the graphic argument provided by Figure 4.3. The M_{cp} purchase matrix presents a very particular shape, i.e. it has a triangular structure. The first rows and columns present a very high density of points while, in comparison, the rest of the matrix is almost empty. The f_* function is the equation of the line dividing the black area of Figure 4.3 (the one with the high density of ones) from the white area. After generating a null model, based on the null hypothesis that this structure is expected given the distribution of products bought by customers, we can say that this structure is not expected and therefore it is carrying some meaning.

Figure 4.3 suggests what is the shape of f_*. For any customer c_j we denote $assortment(c_j) = \{p_1, \ldots, p_{f_*(j)}\}$ and, for any product p_i, $customer\_base(p_i) = \{c_1, \ldots, c_{f_*^{-1}(i)}\}$. The mathematical shape of the f_* map appears to be, from Figure 4.3, anti-monotonic, i.e., $i_1 < i_2$ implies that $f_*(i_1) > f_*(i_2)$, which in turn implies that $assortment(c_2) \subseteq assortment(c_1)$. In other words, if c_1 is customer purchasing more in terms of product quantities than c_2, then it is very likely that c_1 buys the same set of products c_2 buys, plus something more.

Matrices with triangular structures have been already studied in ecology literature. In ecosystems, simpler organisms are ubiquitous and more complex organisms appear *iff* simpler organisms are already present [11]. In these works, authors define nestedness as a measure to understand how much triangular is the structure of the matrix representing the connections between species and ecosystems. The nestedness is calculated by identifying the border dividing the matrix in two areas containing respectively most ones and most zeroes, that is exactly the role of the f_* function we want to define. This function is known as *isocline*.

In literature there are several algorithms tackling the problem of computing the isocline of a matrix [6]. The general approach is usually made in two steps: a reordering of the rows and columns of the matrix, such that the ones tend to be clustered in the upper-left corner of the matrix; and an estimate of the isocline function on the reordered matrix.

In our framework we are implementing an alternative way to calculate the isocline. We have chosen to do so for two reasons. Firstly, all algorithms explicitly reorder the matrix. We do not want to reorder our matrix, since the order we defined in Section 4.2.1 is a fundamental prerequisite for the f_* function, as it has been defined above to connect the ranks of customers and products calculated on their volumes of purchases and sales, respectively. These ranks are obtained by the matrix ordering during the preprocessing stage and cannot be modified. Secondly, the state-of-the-art algorithms are designed to deal with ecology data, with a number of cells in the order of 10^4 or 10^5. Since our cells are $\sim 10^9$, we need to define a new procedure, enabling the application of our framework to large datasets.

We need an evaluation measure to understand if a proposed isocline is good or not. We use the following formulation:

$$N(M_{cp}, f_*) = \frac{1}{2}\left(\frac{f_l(M_{cp}, 1)}{f_l(M_{cp}, *)} + \frac{f_r(M_{cp}, 0)}{f_r(M_{cp}, *)}\right),$$

where $f_l(M_{cp}, *)$ counts the number of cells at the left of the isocline in M_{cp} where we expect to find the ones (and $f_l(M_{cp}, 1)$ counts the ones) and where $f_r(M_{cp}, *)$ counts the number of cells at the right of the isocline in M_{cp} where we expect to find the zeroes (and $f_r(M_{cp}, 0)$ counts the zeroes). In practice, we take the average of the one-density at the left and zero-density at the right of the isocline. In an ideal case, where all the ones are at the left of the isocline and all the zeroes are at the right, $N(M_{cp}, f_*) = 1$, while in the worst case (all zeroes at the left and all ones at the right), $N(M_{cp}, f_*) = 0$. Therefore, the more $N(M_{cp}, f_*)$ is close to 1, the more the matrix is nested. We used this measure because simply counting unexpected presences and absences of ones at the right/left of the isocline is not a fair measure, being our matrix very sparse.

We now need to find the isocline. To find it, we estimate where the isocline should pass to maximize the division of ones at the left and zeroes at the right. We consider our matrix as a Cartesian space. For each discrete x axis value (customer) we get an estimate of where the isocline should pass (y axis). We do so by summing the ones of the corresponding matrix row ($k_{c,0} = \sum_p M_{cp}(c, p)$). Then, for each discrete y axis value (product) we get an estimate of where the isocline should pass (x axis). We do so by summing the ones of the corresponding matrix column ($k_{0,p} = \sum_c M_{cp}(c, p)$). We average these two values and we obtain a pair of coordinates. This procedure is linear in the number of customers and products and therefore it can scale with very big matrices. We fit these coordinates using a non-linear least squares optimization with the Levenberg-Marquardt algorithm to obtain the best function able to represent the isocline and, therefore, the f_* function. For the Livorno2007-

f_*	$N(M_{cp}, f_*)$
$ax + b$	0.616106811666
$ax^2 + bx + c$	0.628533747603
$a\log(x) + b$	0.623911138356
ax^b	0.572996769269
$\frac{a}{x}$	0.609588181022
$-\frac{ax+b}{cx+d}$	0.632547410976

Table 4.1: The $N(M_{cp}, f_*)$ for the different f_* tested.

2009 dataset ($317,269$ customers and $4,817$ products) the whole procedure took seconds.

To fit a function with the non-linear least squares optimization, it is needed the shape of the function. Our framework tries several different shapes, memorizing the $N(M_{cp}, f_*)$ value and then choosing the best performing one. We chose the functions that, at the best of our knowledge, could be a good approximation of the isocline (line, parabola, hyperbola, logarithmic function, and so on). The results of each shape we tried for the Livorno2007-2009 dataset is reported in Table 4.1. The simple non-rectangular hyperbola (last line of Table 4.1) is the best option for our f_* in this dataset. Then, the relation linking the volume sale rank i of a product p_i to the volume purchase j of the last customer of the $customer\_base(p_i)$ set is hyperbolic. The two relations are $i = -\dfrac{\alpha j + \delta}{\gamma j + \beta}$ and $j = -\dfrac{\beta i + \delta}{\gamma i + \alpha}$.

4.2.3 Null Hypothesis

The triangular structure of the matrix in Figure 4.3 gives an important information: a customer that purchased few products is expected to have bought just products that are best seller (base-products as bread, fruit and pasta). This disagrees with the expected presence of "cherry pickers", i.e. customers that are particularly sensible and responsive to sales, especially if the sales are placed on expensive goods. Instead, looking at Figure 4.3, it seems that the customers follow a general pattern.

Starting from this consideration, we need to validate the model, in particular we want to control that the triangular structure is meaningful.

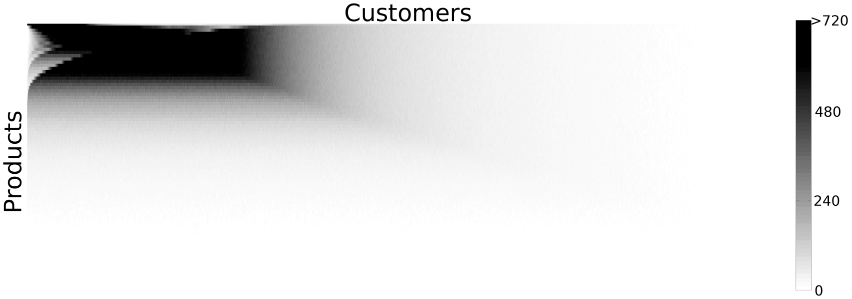


Figure 4.4: One instance of the null purchase matrix. To be consistent with Figure 4.3, also the null matrix has been transposed.

We need, thus, a null model definition with which compare the real world data. We identify three important features that our null model must hold: (1) the purchases are distributed randomly; (2) customers must preserve the total amount of their purchases; and (3) each product must preserve its sale volume on the market.

Given these assumptions, we need to generate a random matrix where the observed sums of rows and columns are preserved. In literature there is an algorithm providing this feature [97], but it is not designed to work on very large matrices. Therefore, we proceed to define our null model with the following relaxed formulation.

We use two sets ($PLeft$ and $CLeft$) to keep trace of the rows and columns that are not yet full (customers that have not yet reached their sum of products bought and products that have not yet reached their diffusion among the customers). Vector R (C), instead, keeps trace in each cell of the respective residual in the row (column). The integer $NItemsLeft$ contains the total number of purchases.

We start from an implicit empty matrix, with the same dimensions of our real data matrix and with all cells initialized at 0. We iterate until we have a product left to place. At each iteration we extract randomly a position from the set of cells that are still increasable (stored in $CLeft$ and $PLeft$). At this point, we just increase by 1 the value of the cell extracted, we decrease the residual of the row and the column selected and of the total number of products. Finally, we check if the column (the row) selected has become full and, in this case, we remove the column (row) index from the set $Pleft$ ($CLeft$). After building this null adjacency matrix, we calculate the lift for each cell and then we apply the same binarization strategy described in Section 4.2.1. We obtain a null M_{cp} matrix and we can then confront it with the original one to understand if they are similar or not (and therefore if the shape of the original matrix is meaningful or not).

To make our procedure more scalable in terms of memory and disk occupation, we used temporary data files storing the temporal null matrix in a sparse format.

We depict in Figure 4.4 one of the null models generated for the Livorno2007-2009 dataset M_{cp} matrix. We can see that Figure 4.4 still presents some of the characteristic of the original M_{cp} matrix. However, in Figure 4.4 popular customers/products tend to have randomly distributed lifts (therefore their columns/rows appear white in the compressed view) and, while preserving some triangularity, the null model matrix have a tendency to display more ones on the top-left to bottom-right diagonal than the original M_{cp} matrix (look at Figure 4.4).

Anyway, we need to show a formal proof that helps us to reject the Null Hypothesis. We evaluate the nestedness of the null models using the very same procedure described in Section 4.2.2 for the original matrix (estimate of the isocline coordinates, fit of different function and calculation of the nestedness value). We found that the average nestedness value for 30 null models was 0.5892564877, with a very low standard deviation (around 2×10^{-5}) and a normal distribution. The absolute value of the nestedness is still high (we can see that the matrix in Figure 4.4 is somewhat nested) so the null hypothesis explain part of the nestedness. But given the standard deviation value, the null models present a nestedness value that is remarkably lower than the observed value in the original matrix (more than 2,000 standard deviations are needed to get to the observed values), in fact, taking the nestedness value of our function from Table 4.1, we get:

$$\frac{0.632547410976 - 0.5892564877}{2 \times 10^{-5}} = 2164.5211638$$

We then reject the null hypothesis stating that the observed nestedness can be only partly explained by the null hypothesis, but it is intrinsically more complex than what is generated by the distributions of volume of sales for the products and volume of purchases for the customers.

4.2.4 Product and Customer Sophistication

In this step of the framework, we want to quantify the sophistication level of the products sold and of the customers buying products. The basic intuition is that more sophisticated products are by definition less needed, as they are expression of a more complex need. One may be tempted to answer to this question by trivially returning the products in descending order of their popularity: the more a product is sold, the more basic it is. However, this is not considering an important aspect of the problem: to be sold to a large set of costumers is a necessary condition to be considered "basic", but it is not sufficient. Another necessary condition is that the set of customers buying the product should include the set of costumers with the lowest level of sophistication of their needs. The conjunction of the two properties is now sufficient to define a product as "basic".

This conjunction is not trivial and it is made possible by the triangular structure of the adjacency matrix. Consider Figure 4.3: the columns in the right part of the matrix are those customers buying only few products. Those products are more or less bought by everyone. In a world where our null hypothesis of Section 4.2.3 holds, instead of buying the products at the top row of the matrix they would buy random products (it is not possible to spot their purchases in Figure 4.4 due to the image compression as they are very few).

For this reason, we need to evaluate at the same time the level of sophistication of a product and of the needs of a customer using the data in the purchase matrix, and recursively correct the one with the other. We adapt the procedure of [58], adjusting it for our big data.

We calculated the sums of the purchase matrix for each customer ($k_{c,0} = \sum_p M_{cp}(c,p)$) and product ($k_{0,p} = \sum_c M_{cp}(c,p)$) to estimate the isocline in Section 4.2.2. To generate a more accurate measure of the sophistication of a product we need to correct the sums recursively: this requires us to calculate the average level of sophistication of the customers' needs by looking at the average sophistication of the products that they buy, and then use it to update the average sophistication of these products, and so forth. This can be expressed as follows: $k_{N,p} = \dfrac{1}{k_{0,p}} \sum_c M_{cp} k_{c,N-1}$. We then insert $k_{c,N-1}$ into $k_{N,p}$ obtaining:

$$k_{N,p} = \frac{1}{k_{0,p}} \sum_c M_{cp} \frac{1}{k_{c,0}} \sum_{p'} M_{cp'} k_{N-2,p'}$$

$$k_{N,p} = \sum_{p'} k_{N-2,p'} \sum_c \frac{M_{cp} M_{cp'}}{k_{0,p} k_{c,0}}$$

and rewrite this as:

$$k_{N,p} = \sum_{p'} \widetilde{M}_{pp'} k_{N-2,p'},$$

where:

$$\widetilde{M}_{pp'} = \sum_c \frac{M_{cp} M_{cp'}}{k_{0,p} k_{c,0}}.$$

We note in the last formulation $k_{N,p}$ is satisfied when $k_{N,p} = k_{N-2,p}$ and this is equal to a certain constant a. This is the eigenvector of which is associated with the largest eigenvalue (that is equal to one). Since this eigenvector is a vector composed by the same constant, it is not informative. We look, instead, for the eigenvector associated with the second largest eigenvalue. This is the eigenvector associated with the variance in the system and thus it is the correct estimate of product sophistication.

However, this formulation is very sensitive to noise, i.e. products that are bought only by a very narrow set of customers. To calculate the eigenvector on the entire set of products generates a small amount of products whose sophistication level is seven orders of magnitude larger than the rest of the products. This variance provokes the other sophistication estimates to be flattened down to the same values and therefore not meaningful. However, we do not want to simply cut the least sold products, as we aim to create a full product hierarchy, including (especially) also the least sold products. To normalize this, we employ a three step strategy. First, we calculate the eigenvector on a restricted number of more popular products (purchased by at least a given threshold δ of customers). Then we use the estimate of the sophistication of these products to estimate the sophistication of the entire set of customers (that is, as defined before, the average sophistication of the restricted set of products they buy). Finally, we use the estimated sophistication of the customers to have the final sophistication of the entire set of products, again by averaging the sophistication of the customers buying them. Hence, we define the product sophistication index (PS) as:

$$PS = -\frac{\mathbf{K} - \mu(\mathbf{K})}{\sigma(\mathbf{K})},$$

where $\mathbf{K}$ is the eigenvector of $\widetilde{M}_{pp'}$ associated to the second largest eigenvalue, normalized as described above; $\mu(\mathbf{K})$ is its average and $\sigma(\mathbf{K})$ its standard deviation. The customer sophistication CS is calculated using the very same procedure, by estimating $k_{c,N}$ instead of $k_{N,p}$.

4.3 Experiments

In the previous section we have defined a framework employing a strategy whose aim is to extract the general pattern governing user behavior, by analyzing the adjacency matrix of the bipartite structure connecting the customers to the products they are buying. In this section, we apply our framework to real world data. We firstly describe the data selection process in Section 4.3.1. Then, we apply the framework, obtaining the f_* function, that is the general pattern of our data, and the sophistication levels of the data we selected, in Section 4.3.2. Finally, we show the usefulness of the framework's outputs providing an empirical observation of Maslow's theory in Section 4.3.3 and a possible marketing application in Section 4.3.4.

All the analysis presented in this section are performed with regular user-end computers. No mainframes or parallel computing techniques have been used. The f_* function fitting of Section 4.2, Sections 4.3.4 analysis and the eigenvector computing for Section 4.3.3 have been performed each one in less than one hour on a Dual Core Intel i7 64 bits @ 2.8 GHz laptop, equipped with 8 GB of RAM and with a kernel Linux 3.0.0-12-generic (Ubuntu 11.10), using a combination of Octave, Numpy and Scipy Python libraries. Data preparation pipeline (Section 4.3.1) and null model generation and evaluation (Section 4.2.3) have been computed on a Quad Core Intel Pentium III Xeon @ 2 GHz, equipped with 8 GB of RAM and with Windows Server 2003, using Java 1.6. The most memory and time consuming operation was the null model generation: each null model required 6 GB of memory and 4 hours of computing. The conclusion is that our framework is able to scale and to analyze large data quantities.

4.3.1 Data Preprocessing

Considering that the dataset contains more than $1,5$ million customers and almost 440k items, to build a matrix "customers × items" would generate ~ 660 billions of cells, that is redundant for our purposes; hence we need a sort of reduction on both the dimensions (customers and items). There are two main criteria to select the customers: on the basis of their purchase behavior (e.g. excluding from the analysis all the people that did not purchased at least a total number x items) or geographically (e.g. considering just the customers of an area). We decided to apply the latter filter, since we do not want to exclude any customer behavior apriori. We select a subset of shops in the dataset belonging to the same areas of Italy. The number of customers per area is presented in Figure 4.5. We generated different views of the dataset for different purposes. Our main dataset is Livorno2007-2009, that is including all the purchases of the customers located in the city of Livorno during the period from 2007 to 2009. We use only this view for the applications of the framework's

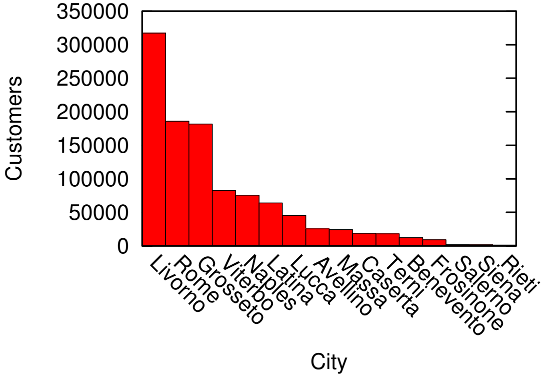


Figure 4.5: Customer distribution per city in our dataset.

output. We also generated the dataset Lazio2007-2009 (same period, different geographical location, the sum of the cities of Rome, Viterbo, Latina, Rieti and Frosinone) and Livorno2010-2011 (different period, same geographical location). These two views are generated to prove that the fundamental properties of the adjacency matrix needed for our framework are not bounded to a particular place or time. The following steps of data preparation are applied equally to the different datasets extracted.

The second issue, as introduced above, regards the cardinality of products. There is a conceptual problem in using the level of detail of "item": the granularity is too fine, making the analysis impractical as it would consider a very low detail level. The distinction between different packages of the same product, e.g. different sizes of bottles containing the same liquid, is not interesting here. A natural way to solve this problem is to use the marketing hierarchy on the products, substituting the item with the value of the marketing Segment. In this way, we reduce the cardinality of the dimension of the product by 98% (from $439,619$ to $7,004$), aggregating at the same time products that are equivalents.

The last step in data selection is to exclude from the analysis all the products (segments) that are either too frequent (e.g. the shopper) or meaningless for the purchasing analysis (e.g. discount vouchers, errors,

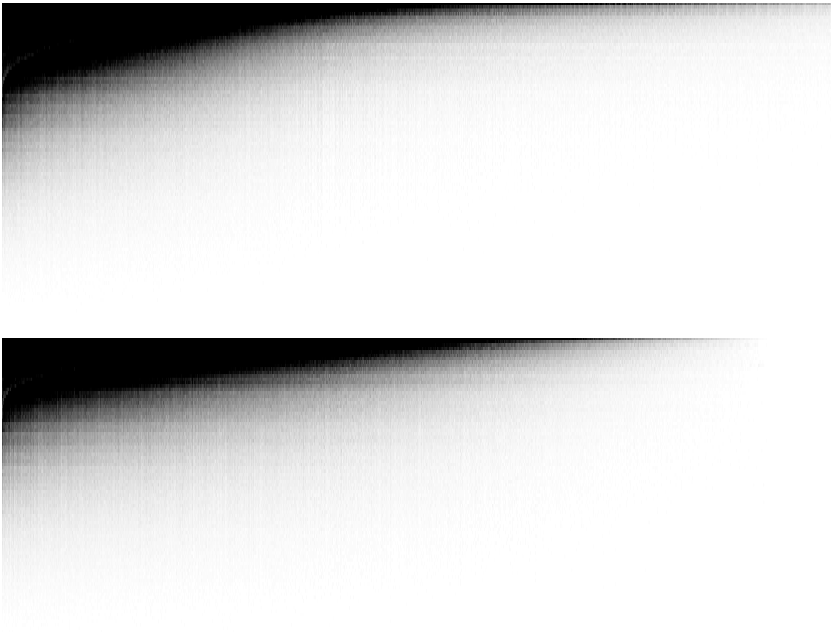


Figure 4.6: The M_{cp} matrices for Livorno2010-2011 (top) and Lazio2007-2009 (bottom).

segments never sold, etc.). After this last filter (and consequently the discharge of the customers that bought exclusively products classified under the removed segments), we got our adjacency matrix, ready to be provided as input of our framework. Livorno2007-2009 matrix has $317,269$ customers and $4,817$ segments, with $182,821,943$ purchases; Livorno2010-2011 has $326,010$ customers and $4,807$ segments, with $183,679,550$ purchases; and Lazio2007-2011 has $278,154$ customers and $4,641$ segments, with $135,517,300$ purchases.

4.3.2 Framework Application

In this section, we apply our framework on the three views of the dataset extracted. We report the application of each step.

M_{cp}	$N(M_{cp}, f_*)$
Livorno2007-2009	0.632547410976
Lazio2007-2009	0.622983174602
Livorno2010-2011	0.615276275848
Null Model Average	0.5892564877

Table 4.2: The $N(M_{cp}, f_*)$ for the different views of the dataset.

1) *Calculation of the M_{cp} matrices from the adjacency matrices.* The results is three M_{cp} matrices for Livorno2007-2009, Livorno2010-2011 and Lazio2007-2009. The number of rows and columns of the matrices are not changed, and the total number of ones (i.e. significant purchases according to the lift) are $37,338,591$ for Livorno2007-2009, $43,982,774$ for Livorno2010-2011 and $45,410,992$ for Lazio2007-2009. Livorno2007-2009 matrix is depicted in Figure 4.3, Livorno2010-2011 and Lazio2007-2009 matrices are depicted in Figure 4.6, top and bottom respectively (the legend for both Figures is the same as Figure 4.3 legend).

2) *Calculation of the null model.* We already depicted one null matrix for Livorno2007-2009 in Figure 4.4, that is an accurate depiction also of the typical null matrix for Livorno2010-2011 and Lazio2007-2009.

3) *Calculation of the f_* function.* We obtained a simple hyperbola in all the three cases in exam. The evaluation via the $N(M_{cp}, f_*)$ function of the goodness of the division operated by the isocline is provided in Table 4.2. In the last row of Table 4.2 we also provided the average isocline evaluation for the null models that were generated (around 30). As we can see, the average value for null model is lower, and given a standard deviation of the order of 10^{-5}, we can conclude that the difference is also significant. The three views provide a proper input for our framework and we are able to extract knowledge through its application. For the Livorno2007-2009 dataset, the value of the parameters has been estimated as: $\alpha = 11318.559$, $\beta = 94.2526$, $\gamma = 0.2834$, $\delta = -16866558$, and the corresponding hyperbola has been drawn in red in Figure 4.3 (please remember that the matrix in this Figure has been rotated clockwise of 90 degrees). We do not report the values of the parameters for the other datasets as we are not using them in the rest of the section.

p_i	PS
Regular Bread	-4.41
Natural Still Water	-4.19
Yellow Nectarines (Peaches)	-3.84
Semi-Skimmed Fresh Milk	-3.81
Bananas	-3.53

Table 4.3: A selection of the more basic products according to their PS values.

p_i	PS
LCD 28″/30″ Televisions	2.91
DVD Music Compilations	2.86
Sauna clothing	2.66
Jewelry Bracelets	2.53
RAM Memories	2.33

Table 4.4: A selection of the more sophisticated products according to their PS values.

4) *Calculation of the product and customer sophistication*. We present the most and least sophisticated products only for the Livorno2007-2009 dataset, for the same reason of the previous point. Also we do not report the customer sophistication for privacy concerns. In table 4.3 we report a selection of the least sophisticated products, i.e. to ones with the lowest PS values, in the purchase matrix. The less sophisticated products should be intuitively the ones covering the most basic human needs, and this intuition is confirmed by the reported products: bread, water, fruits and milk. On the other hand, table 4.4 reports the most sophisticated products, i.e. the ones with the largest PS values, that intuitively should be products satisfying high-level non-necessary, probably luxury, needs. In fact, what we find in Table 4.4 are hi-tech products (LCD televisions, DVD compilations, computer accessories), jewelry and very specific clothing.

4.3.3 Hierarchy of Needs

In this section we want to use the information provided by the product sophistication index to reconstruct the hierarchy of needs of the supermarket customers, and therefore provide an empirical observation of the theory of Maslow [87].

Two caveats need to be specified. First: we are not claiming that this hierarchy of needs is universal. The result we are presenting in this section has been reached with data from one city of Italy (Livorno) and therefore it describe the hierarchy of needs of that particular city. However we showed that the triangular structure of the purchase matrix is present even in different areas of Italy (Figure 4.6) and therefore our framework probably may be applied to different world regions and it may help to create a picture of different hierarchies of needs. The confront of hierarchies of needs of different cities and the evaluation of different cultural perspectives of customers over their needs is left as future development. Further, this hierarchy is a valuable marketing tool for that particular city: products at the basis of the hierarchy are more needed, thus no marketing strategies are required for them as they will be sold anyway.

The second caveat is that we built the hierarchy of needs using the product category classification defined by the supermarket owners (see Section 4.3.1). To use this classification introduces the bias of a precise set of people, with a given culture and marketing aims. We plan to use for future developments standard product classifications.

With this intuition in mind, we now build the hierarchy. To build the hierarchy we need to divide products in classes according to their PS value. Formally, we need to do a segmentation of the PS sorted values. We decided to perform a one-dimensional clustering using the ck-means algorithm (an evolution of the k-means algorithm which guarantees the optimality of clustering [122]). We set $k = 5$, as we follow Maslow's hierarchy of needs classification [87] and we want to obtain roughly the following classes of products: fundamental for survival, basic needs, complementary needs, accessory needs and luxury needs. The results of the ck-means clustering have been depicted in Figure 4.7. In Figure 4.7 for

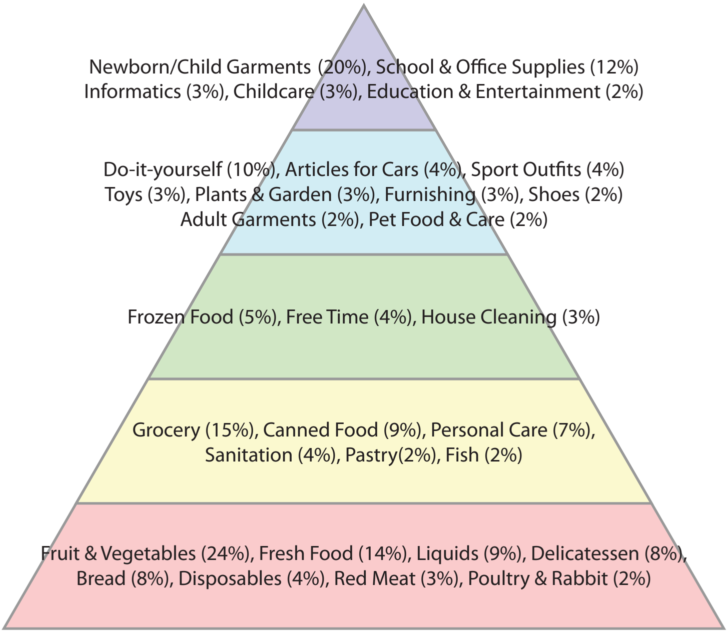

Newborn/Child Garments (20%), School & Office Supplies (12%)
Informatics (3%), Childcare (3%), Education & Entertainment (2%)
Do-it-yourself (10%), Articles for Cars (4%), Sport Outfits (4%)
Toys (3%), Plants & Garden (3%), Furnishing (3%), Shoes (2%)
Adult Garments (2%), Pet Food & Care (2%)
Frozen Food (5%), Free Time (4%), House Cleaning (3%)
Grocery (15%), Canned Food (9%), Personal Care (7%),
Sanitation (4%), Pastry(2%), Fish (2%)
Fruit & Vegetables (24%), Fresh Food (14%), Liquids (9%), Delicatessen (8%),
Bread (8%), Disposables (4%), Red Meat (3%), Poultry & Rabbit (2%)

Figure 4.7: Our data-driven pyramid of needs with the most basic products at the bottom and the most sophisticated at the top. For each product category we report its share among all purchase at that level of the hierarchy.

each level of the hierarchy we report its main composition according to the product categories. The share values between the parenthesis tells, given the total amount of products purchases at that level of the hierarchy, how many of those belong to that particular category. For instance, skimmed and semi-skimmed milk may belong to two different hierarchy levels, say 0 and 1, and they have respectively been sold 4,000 and 2,000 times. Let us say that the total amount of products sold in hierarchy levels 0 and 1 are respectively 4,000,000 and 1,500,000. Then, skimmed milk contributes to level 0 as 0.1%, while semi-skimmed milk contributes to level 1 as 0.13%. We report only categories representing at least 2% of the hierarchy level. We did not report the single product segment, as they are too specific and too many: for instance apples, pears, bananas, tomatoes, potatoes and so on have been aggregate in the product category "Fruits & Vegetables". Of course, products in the same category may fall in different hierarchy levels: in Figure 4.7 we chose to put the category where it occupies the largest share of the level purchases.

Figure 4.7 is clear depiction of what are the priorities in the mind of the customers of Livorno. Figure 4.7 is telling some expected and some unexpected things. First there are the basic needs: drinking and eating, particularly fruit, vegetables, bread and meat. Then, there are more sophisticated eating products and what is needed to take care of the body hygiene. At the middle of the hierarchy we start to have product not strictly necessary for survival: house cleaning and simple products for the free time. The two most sophisticated needs are schooling, entertainment (both for children and adults), more complex garnishment; and, climbing at the top of the pyramid, newborn childcare and unnecessary equipment. The basis of the pyramid is expected: most basic needs are food and personal hygiene. Up until now we have basic confirmation about human needs. The top of the pyramid, instead, contains information that could be biased by the fact that we have only information coming from one retail company. Some specific (highly sophisticated) product could be bought in other contexts, just because there exist very specialized shops (or because can be bought at a lower price). Anyway, the products at the top of the pyramid seems to be good candidates for "high sophisticated products in a retail shop", but we leave a better and deep analysis of this aspect as a future work.

4.3.4 Marketing

We now describe a possible targeted marketing strategy based on the outputs of our framework. Suppose the supermarket wants to promote a product p_i and it wants to limit its target to the smallest subset with the highest probability of actually buying the product advertised. The f_* function can be used in the following way: given the amount of products bought by customer c_j we use its index j to obtain the index $f_*(j) = i$ of the most sophisticated product p_i that c_j is buying, and therefore the entire set of products he/she is expected to buy, that is $assortment(c_j)$, defined as all the products that have an index $i' \leq i$. The same applies considering as input a product p_i, we obtain the index delimiting the set of customers buying it (for which $j' \leq f_*^{-1}(i)$).

One concern needs to be addressed before continuing: how well is the f_* function dividing the ones from the zeros w.r.t. what we expect? How much is a customer more likely to buy a product following the f_* function evaluated on our real world data (P_f) over any random product (P)?

As presented in Section 4.3.2, the purchase matrix Livorno2007-2009 has ∼37 millions ones out of ∼1.5 billions cells, thus given a random product p_i and a random customer c_j the baseline probability $P(p_i, c_j)$ that customer c_j is buying product p_i in a significant amount (i.e. $\text{lift}(c_j, p_i) > 1$) is the ratio of these two numbers, or $P(p_i, c_j) = 2.44\%$. If we consider only the portion of the matrix at the left of the calculated isocline, i.e. the area of the matrix for which the f_* function tells us that the customers are very likely to buy exactly that products, we count 16,748,048 ones and 60,025,000 total cells. Thus, the probability $P_f(p_i, c_j)$ for a customer c_j to buy significant amounts of a product p_i for which $i \leq -\dfrac{\alpha j + \delta}{\gamma j + \beta}$ (i.e. $p_i \in assortment(c_j)$) is 27.9%. Using the f_* function, we can narrow of two orders of magnitude the set of combinations of products and customers to analyze and still capturing almost half of the significant purchases. In other words, customers are 11.43 times more likely to buy a

p_i	p_{i-1}	$P(p_i)$	$P(p_i \mid p_{i-1})$
Dishwasher Salt	Dishwasher Soap	8.39%	30.41%
Asparagus	Olive	8.00%	26.12%
Peppers	Chicory	7.31%	23.73%
Canned Soup	Preserved Anchovies	9.96%	32.23%
Wafers	Sugar Candies	11.30%	21.67%

Table 4.5: The probabilities of buying product p_i in general ($P(p_i)$) and given that a customer already buys product p_{i-1} ($P(p_i|p_{i-1})$).

product p_i if i is lower than, or equal to, the index limit predicted by the f_* function. We refer to this ratio as $\frac{P_f(p_i,c_j)}{P(p_i,c_j)}$, i.e. the f_* function based probability of connecting customer c_j with product p_i over the baseline probability. We also calculated the same ratio, this time by counting at the right side of the isocline, where we expect to find many zeros. The number of ones is 37 millions minus 16 millions, and it is divided by the number of cells, 1.5 billions minus 60 millions. The probability of obtaining a one is 1.39%, less than one twentieth of the left side of the isocline.

Now that we have addressed the main concern about the f_* function, we can safely assign to product p_i a corresponding customer index $j = -\dfrac{\beta i + \delta}{\gamma i + \alpha}$ that is its current "border": all indexes $j' \leq j$ represents customers who buy product p_i (i.e. $\forall j' \leq j, c_{j'} \in customer\_base(p_i)$), while the indexes $j'' > j$ are customers not buying p_i. By definition, the higher the value of j'', the more unlikely is the customer buying p_i. Thus, the set of customers the law is suggesting to target is the one immediately after index j. Since the f_* function is an interpolation, it is safe to define a threshold ϵ_1. Then, we define the set TC, the target customers set, as the set of all customers for which, given their index j', it holds: $j - \epsilon_1 \leq j' \leq j + \epsilon_1$ and $M_{cp}(c_j, p_i) \neq 1$ (the last condition is necessary to exclude from TC all customers who are already buying large quantities of product p_i, as it is useless to advertise p_i to them).

How can we evaluate how many elements of TC will be likely to start buying p_i? Given our considerations in the previous paragraph, it

p_i	$\|TC^*\|$	$\|TC\|$	$\frac{\|TC_r\|}{\|TC\|}$
Tomino Cheese	58	137	7.51095
Raw Ham	78	144	5.81250
Apricot Jam	66	127	4.66142
Anchovies	83	144	4.06250

Table 4.6: The comparison between the size of the target customer sets identified by the f_* function against random target customer sets with the same number of customers likely to buy p_i.

holds that having a 1 in the product of index $i-1$ makes the customer very likely to buy the next more sophisticated product p_i, i.e. to have purchased large amounts of the product immediately to the left in the matrix to p_i increase to probability of purchase this product. For instance, customers buying "Dishwasher Soap" have 30.41% probability of buying product "Dishwasher Salt" against a baseline probability of 8.39%, some instances of this are provided in Table 4.5. On average, the $\dfrac{P(p_i|p_{i-1})}{P(p_i)}$ ratio is 1.993 for the 500 most sold product, and no single product has a ratio lower than 1 (the lowest is 1.05 for Fresh Bread). Therefore, for each $tc \in TC$ element we check if $\exists x, M_{cp}(tc, p_x) = 1$, with $i - \epsilon_2 \leq x < i$, thus looking not only at the direct left neighbor of product p_i, but at his ϵ_2 left neighbors. If the condition holds, we have identified TC^* as the subset of TC composed by those customers who are likely to buy p_i.

The question now is: how large should be a TC_r set to obtain an equally large TC^*_r set if TC_r has been populated without knowledge about the f_* function, i.e. at random by picking customers who are not already buying product p_i? We address this question by looking at several different products. For each of them we identified the TC set using the f_* function and then we calculated 500 random TC_r sets. In Table 4.6 we report, for each product p_i, the following statistics: the number of customers likely to purchase p_i ($|TC^*|$ column), the total number of targeted customers ($|TC|$ column) and the average ratio between the targeted customers without and with using the f_* function ($\frac{|TC_r|}{|TC|}$), by fixing $\epsilon_1 = 100$ and $\epsilon_2 = 2$. As we can see, the knowledge provided by the f_* function reduces the number of customers to be targeted by a marketing campaign of four or more times, with the same return of investment (as our procedure fixes $|TC^*| = |TC_r^*|$). Table 4.6 reports only a few products, but we tested these 500 random sets for 800 different products and the average of the averages of the $\frac{|TC_r|}{|TC|}$ ratio is 3.55594, i.e. on average using the f_* function the marketing campaign can target three times less customers or less. For none of the 800 products the average of the ratio was less than 1.

4.4 Conclusion

In this section we analyzed large quantities of data extracted from the retail activity of the customer subset of an Italian supermarket chain. Our aim was to build a framework able to take advantages of some properties of the data which undermines the completeness of association rule mining results, thus providing an alternative and complementary methodology to mine purchase data. These properties are the uneven distributions of connections in the customer-product bipartite structure and the triangular structure of its adjacency matrix. We found that customers usually start buying the same set of basic products and the more sophisticated products are only bought by customers buying everything, providing a triangular adjacency matrix for the bipartite structure. Our framework is able to analyze this structure as a whole, instead of looking at the local patterns like classical rule mining, returning the general pattern of shopping behavior. From this consideration, we were able to define a function that can identify the set of customers buying a specific product by looking simply at how much the product is sold (and vice versa); and a way to rank the sophistication level of both products and customer needs. We showed some possible applications of these results: a data driven empirical observation of Maslow's theory of needs and an efficient way to identify a small set of potentially very interested customers for a given product p_i.

Our paper opens the way to several different future developments.

The first one concerns the validation of our observation of the hierarchy of needs, as it is based on a narrow geographical set of people and on a non-standard product category classification. Also, with more data we can extend our pyramid of needs to fully cover the entire spectrum of human needs. Another interesting track of research may be to investigate what is the minimum time window needed to observe the prerequisites of the f_* function, maybe linked with the cyclic behavior of customers [113] and/or with the stability of customer and product ranking order in the matrix [112]. Another application scenario may be to fully exploit the purchase matrix as a complex system: to analyze products not only based on their product sophistication index, but by looking at the product-product relationship level; or to try to find the way of controlling the complex system [82].

Chapter 5

Explaining the Product Range Effect in Purchase Data

This chapter is mainly based on [100].

In the economic literature, market society is considered driven by rationality and the expression of this rationality is the price system. According to this view customers are rational beings: they try to minimize the amount of money they are spending, while at the same time maximizing the amount of goods they are purchasing [60]. Therefore, price is a generic utility function that each customer tries to minimize, and it is the same for everybody. However, customers are also driven by their own personal needs and desires [41]. Many of these needs are shared with other customers, such as the basic needs for survival, but many others are intimately bound to each individual and possibly different from the ones of everybody else. A customer is driven both by a generic utility function (cost minimization) and by a personal utility function (fulfillment of unique desires).

If we are able to quantify the personal utility function for each customer, then we can address a question with repercussions on a seller's market strategy: which function will win the arms race in influencing the purchase behavior of a customer, the generic one or the personal one? If the generic one is stronger, then a seller is forced to compete mostly on the price; while if a customer's needs are more important, then it is the quality of the choice that matters the most.

Here, we develop an analytic framework based on mining big customer transaction data, aimed to quantify the strength of both utility functions. We test the customer behavior in terms of distance traveled, under the assumption that customers want to minimize their travel length. We observe that customers do not always go to the closest supermarket: there is a *range effect* for each product, due to the intrinsic characteristics of the product. To explain and predict the range effect we propose a method to compare the strength of the generic and the personal utility function in the customer's mind. This comparison boils down to the question: given that customers travel on average x meters to buy product p, are they doing that because p is expensive or because p satisfies very particular needs?

While the price is an explicit information of the product, the needs the product itself is satisfying are not. We quantify them by evaluating the *sophistication* of each product and customer, following Chapter 4. We find that the sophistication of a product is better than the price in explaining a customer's behavior.

We provide empirical evidence of these claims with real world data about customer behavior. We analyze digital traces of customers purchases in our sales dataset considering a single Italian city.

We show how our proposed product sophistication index is a better explanatory variable of a product range than the price. The more sophisticated is the need a product satisfies, the longer a customer will travel to purchase it on average, almost regardless of its price. Intuitively, this means that to buy bread people will just settle with the closest shop where it is available, while to buy blank DVDs, with roughly the same price and available in all the supermarkets of the chain, a customer will travel a significantly longer distance. While the product range concept may be quite intuitive, in this section we provide a system able to quantify it better than just assuming that it is proportional to the product price.

There are many consequences for sellers from the ability of predicting a product's range. For instance, to know the range of all the products of a supermarket implies that the supermarket's marketing strategies can be tailored according to the distance of a customer from the nearby points of sales. Customers far away from a point of sale need to be stimulated on more sophisticated needs, while nearby customers may be more susceptible to more basic needs.

A second application is in point of sale placement, as we can use our methodology in conjunction with the central place theory [29]. Besides the construction costs, each point in the city space is altering the minimum distance between a customer and a product. Therefore, given the range effect, each point in the city space has one optimum in its product assortment. Here, we provide the proof that this problem can be formally addressed to find a good approximate solution.

The final contribution is to show how to accurately predict how long a customer will travel (or which shop she will choose) to buy a given product, as a function of the product's sophistication. In other words, product sophistication reveals as a powerful predictor feature for a challenging predictive task, because most people shop preferably at the closest store for most products, so it is difficult to accurately characterize for which products a customer will travel more.

These applications have to cope with the enormous amount of data flowing every day in the real world. For this reason, we create a scalable framework, using a data mining approach similar to the one at the basis of the PageRank [93], that is able to analyze networks with hundreds of million of nodes. To increase the interpretability of our results, we narrow our questions to the customer base of a city, making possible to let emerge from the data the knowledge about the actual range of each product.

5.1 Data Selection

Since our analysis is based on distances between customers and stores, we focus our presentation only on one metropolitan area, to be able to interpret more easily the results and avoid the problem of people in the border of more cities. Again, it is important to notice that our framework can easily scale for larger data collections [98]. We chose a city motivated by the strong penetration of customers for our retail distribution company. The customer base is not only large, but very committed to the brand, being "members" of COOP, and not just getting discounts: we can fairly assume that the people we are studying make most of their purchases in this chain. For each of the five stores we selected all the customers within a radius of $5km$ from each store.

The resulting dataset contains $60,366$ customers and $4,567$ segments, with $107,371,973$ total purchases[1].

Again, we used the same method based on lift described in Section 4.2.1.

The purchase matrix M_{cp} is depicted in Figure 5.1, where rows and columns are sorted according to the volume of sales: from left to right customers are sorted according to how many products they bought and from top to bottom products are sorted according to how much they are sold.

In Figure 5.1, the columns of the matrix are the $60,366$ customers and the rows are the $4,567$ products. We depicted a compressed view of the matrix, where each data dot represent a 30×30 square of the original matrix and the gray gradient represents how many $1s$ are present in that section of the matrix, for space constraints.

5.2 The Range Effect

The assumption is that customers modify their shopping behavior according to their relative position w.r.t the shop they are going to. A customer may decide to buy or not buy a given product because it is close enough or too far away from the shop. We call this phenomenon the *range effect* of a product. Table 5.1 reports the average distance traveled for purchasing

[1] This dataset has been made available along with all the framework coding at `http://www.michelecoscia.com/?page_id=379`. Customer and product IDs are obfuscated for privacy and business protection reasons.

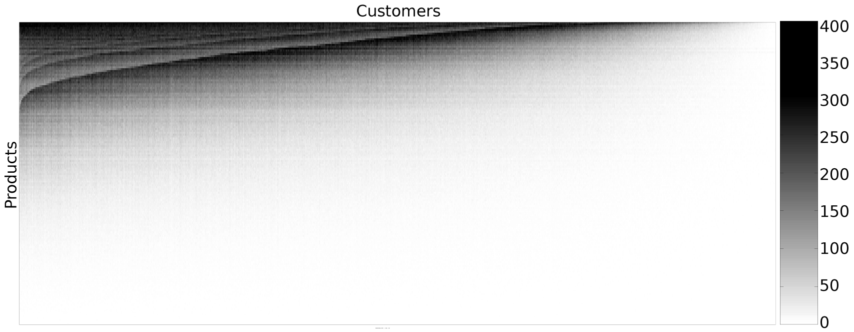


Figure 5.1: The purchase matrix M_{cp}.

Product	AVG Distance (in meters)
Pizza	809
Packed Salads	$1,576$
Frozen Side Dishes	$2,437$
School Notebooks	$3,511$
Travel Books	$5,523$

Table 5.1: A selection of the more basic products according to their PS values.

a product. We can find products for which customers traveled more than 5 kilometers on average, other products for which the average distance is less than 1 kilometer and many other products generated a variety of average distances. There are two trivial explanations of this fact: it is driven by price and/or by the frequency with which a product needs to be purchased.

We expect that customers will travel more to purchase products that are more expensive, for many possible reasons (they require higher quality, they may be not available around them, and so on). We check this hypothesis by plotting for each purchase the price of an item against the average distance that a customer traveled to get the product. This plot is depicted in Figure 5.2: the price is on the x axis (in logarithmic scale),

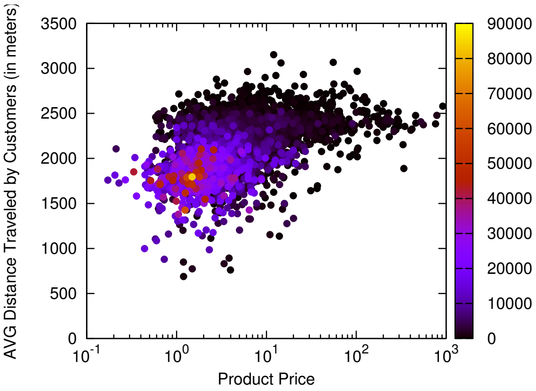


Figure 5.2: Average distance traveled to get a product with a given price.

while the distance traveled is on the y axis. The price is recorded in Euros. Each dot is a purchase and we color it accordingly to how many purchases are represented by the same price and by the same distance.

Intuitively, it would make sense to plot just one point per product, as we want to know the average distance traveled by customers given a product price. However, this would disproportionately weigh the purchases of products sold less frequently, or the purchases made by customers who buy only a handful of products. Filtering out these purchases also would not make sense, as our purchase matrix is triangular: there are many products sold in small quantities and many customers who purchase only few products. Therefore the behavior of these shoppers is important. By plotting each single purchase, we know that we are weighing each customer behavior by its fair proportion of purchases.

The connection of a customer to a product is created with the procedure described in Chapter 4, therefore we are only considering connections generated when the quantity of product p bought by customer c_i is significant. A customer c_i may have bought product p_j in different shops, say s_1, s_2, s_3, s_4. In this case, we weigh each distance traveled with the amount of purchases made using the following formula:

$$d(c_i, p_j) = \sum_{\forall s \in S} \frac{p_j(c_i, s) \times d(c_i, s)}{p_j(c_i, *)},$$

where S is the set of all shops, $d(c_i, s)$ is the distance between customer c_i and shop s, $p_j(c_i, s)$ and $p_j(c_i, *)$ is the amount of purchases of product p_j made by customer c_i in shop s and in general, respectively. This procedure has been done for the plots depicted in Figures 5.2, 5.3, 5.4(a) and 5.4(b).

Products with the same price are bought by customers placed at different distances w.r.t the shop. Given a price, we average the distance traveled by the customers buying the products with that exact price. By averaging, we lose the ability of describing each single customer and we just describe the behavior of the system in its entirety. We do so because the single customer is bounded by the place where she lives, thus each single customer carries a noisy information, and we can make sense of it only by looking at the global level.

From Figure 5.2 we can conclude that price plays a role in driving customer decisions of traveling a given distance for a product. The correlation here looks weak, but positive: customers travel more if they need to buy a more expensive product. We calculate a log-normal regression[2] using the function $f(x) = a \log x + b$. In this regression, $R^2 = 17.25\%$, meaning that we can explain 17.25% of the variance in the distance traveled using the price.

To check if the frequency of purchase can explain the distance traveled by customers, we repeated the same analysis, using the number of purchases of a product instead of the price. We depicted the plot in Figure 5.3. The correlation here is negative: the more frequently a product needs to be bought, the smaller the distance a customer will travel for it. We calculate a regression with the function $f(x) = a \log x + b$ and we obtained $R^2 = 32.38\%$.

As a conclusion of this section, we can state that the price plays a small role in predicting the distance a customer will travel for purchasing a

[2] This and all other regressions have been calculated with the *leastsq* function of the *SciPy* module for Python.

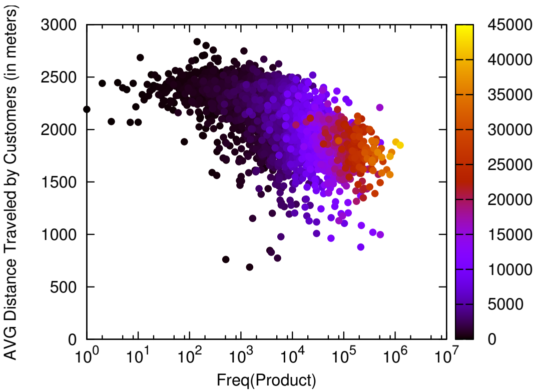
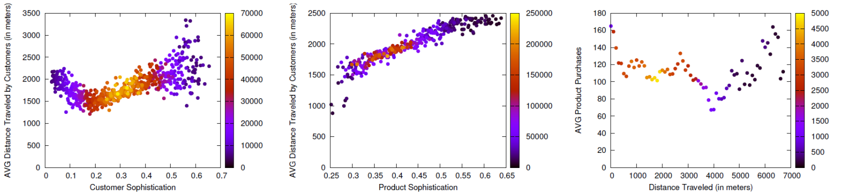


Figure 5.3: Average distance traveled to get a product with a given popularity.



(a) Average distance traveled to get a product with a given sophistication index

(b) Average distance traveled by a customer with a given sophistication index

(c) Average number of purchases by customers at a given distance from the shop

Figure 5.4: Sophistication and customer behavior against the distance from the shop.

product, by increasing it. If a product is needed more frequently then it drives (down) the distance a customer will travel to buy it, regardless of the price. However, there is a large amount of variance that remains unexplained. In the next section, we provide one possible explanation.

5.3 Explaining the Range Effect

In this Section we tackle the problem of explaining the *range effect* for products. Our theory states that customers travel more to buy a product if the product can satisfy a more sophisticated need and/or they have sophisticated needs in general. To do so, we use the formal definition of what exactly product and customer sophistication are, as explained in Chapter 4. Then, we provide evidences that the product and the customer sophistication are variables able to better explain the distance traveled by customers, in Section 5.3.1. Finally, in Section 5.3.2 we provide explanations of why our product sophistication index is better predictor of customer behavior.

5.3.1 Sophistication and Range

To understand if the product and/or the customer sophistication is influencing the distance a customer will travel to purchase the product she needs, we generate the same plots shown in Section 5.2. The plots are depicted in Figure 5.4(a-b). We recall that in these plots each data point is a purchase.

In Figure 5.4(a) we test the relationship between the distance traveled and the customer sophistication: we calculate the average distance traveled by customers (y axis) to get to the shop against their sophistication value (x axis). In this case, the x axis has not a logarithmic scale, as the relationship is linear.

We can see that the relationship between distance traveled and customer sophistication looks non-linear. From a value of sophistication of 0 to around 0.2 the relationship is negative, while it is clearly positive afterwards. The sole conclusion we have is that there is some kind of relation, but we do not have an explanation for it.

For this reason, we move on in depicting the product sophistication (x axis) against the average distance traveled by the customers to purchase the given product (y axis) in Figure 5.4(b). In this case, the relationship is clear: the more a product is sophisticated, the more customers will travel to buy them. The product sophistication has a normal distribution, but less sophisticated products are more sold, given the triangular shape of the matrix. This fact explains why most of the data points are in the left part of the plot: most purchases are generated for low sophistication products. We calculated a linear regression, for which $R^2 = 85.72\%$. This R^2 is more than twice higher than the R^2 obtained with the purchase frequency, explaining much better the variance in the distances traveled by customer.

A possible objection is that the distance is influencing the number of products purchased by a particular customer, and this would invalidate the explanatory power of the product sophistication index. We already saw in Section 5.2 that the distance and the frequency of purchase are somewhat related, but this relationship cannot fully explain what we see in Figure 5.4(b). However, this objection is focused on the customer, not on the product: it states that the customer-shop distance may have a strong positive or negative correlation with the number of items purchased on average by he customer.

We depict this relationship in Figure 5.4(c): the x axis is the distance of a customer from the shop and on the y axis we have its average number of products purchased. Customers at the same distance may have purchased different quantities of products, so we average them and we color the data point accordingly to how many customers it represents. As we can see, there is no relationship at all between distance and number of products purchased. For this reason, we can reasonably state that customers tend to travel more to purchase more sophisticated products.

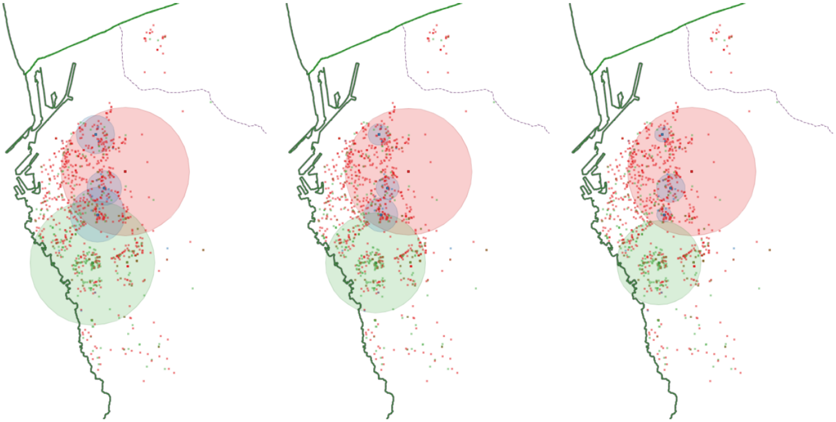
5.3.2 Customer Behavior

We now provide a possible explanation of customer behavior. Customers tend to buy products with a low sophistication level in the closest possible shop. However, when they are in need to buy a more sophisticated product, they do not choose the closest shop even if the shop has the product they look for (and, given the fact that we are considering shops of the same chain, the quality level of the products is identical).

We provide a visual argument for this explanation. In Figures 5.5(a-c) we generated three maps representing the purchases of three different products. We chose products with different sophistication level: in Figure 5.5(a) we focus on a very low sophisticated product (pasta), in Figure 5.5(b) we focus on a medium-low sophisticated product (breaded frozen fish) and in Figure 5.5(c) we focus on a medium-high sophisticated product (health testers, like pregnancy or insulin indicators). Each dot in the map is a location in which we found one or more customers that has bought the given product. The color of the dot represent the shop type in which the customer went for her purchase. The colored circles are centered on the position of the given shop and their radius is the median distance traveled by customers to purchase the product in that shop.

As we saw in Chapter 3, shops have an attribute "type", that encodes the category of the shop, a proxy of its size. In Figures 5.5(a-c), the customers in red went to the "iper" shop (the largest in our data), the customers in green went to the "super" shop (smaller than the "iper"), while customers in blue went to one of the three "gestin" shops (smaller than a "super"). As we can see, the smaller shops have quite some range in attracting customers who need the lowest sophisticated product. However, as the sophistication of the product increases, the number of customers going at those shops becomes lower and lower. The red circle keeps its radius, while the green and blue circles tend to shrink.

Instead of relying on three examples out of the $4,567$ products, we report this trend in Table 5.2. For each shop, we record the average product sophistication of the products sold in that shop in significant quantities (column "AVG PS"). We also record the average distance traveled by



(a) Pasta (b) Breaded Frozen Fish (c) Health Testers

Figure 5.5: The customer-shop distribution for customers buying different products.

Shop Type	AVG PS	AVG Distance
Iper	0.49	$2,392$
Super	0.46	$1,721$
Gestin	0.43	869

Table 5.2: Average PS values and average customer distance for the shops in our dataset.

customers to buy products in significant quantities in that shop (column “AVG Distance”). The “shop type” column refers to the shop classification explained in Chapter 3.

We can see that indeed there is a difference between the complexity of the “iper” (red) shop with the “super” (green) shop, and another significant difference between the “super” shop and the rest of the “gestin” shops. This significant difference is also reflected in the average distance traveled by customers: almost 2.4 kilometers to get to the “iper” shop, more than 1.7 kilometers for the “super” shop and less than 900 meters for the “gestin” shops. Table 5.2 proves that large shops are objectively more sophisticated than smaller ones and suggests that are also subjectively considered so by customers.

The conclusion we draw is that the average sophistication of the products in a shop is influencing customers’ decisions: when they need a more sophisticated product they are prone to decide to go to a larger shop with higher sophistication even if that product is also present in the smaller shops.

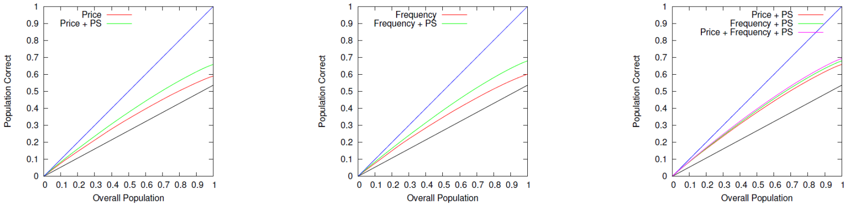
In the next section, we put our finding into practice.

5.4 Customer Behavior Prediction

In the previous sections we showed how the product sophistication can be used to describe the average customer behavior better than the price of a product or how frequently the product is needed. However, as noted, the average customer and all the separate individual customers are different entities. If the knowledge about the average customer cannot be used for predictions, then the product sophistication cannot be used in practice. Aim of this section is to show that predictions based on the product sophistication can achieve a significant improvement over the ones based on price and frequency of purchase.

To do so, we provide the following problem definition:

Definition 5.1 *Let D be a set of triplets (c, p, s). Each triplet represent the purchases of product p made by customer c, and $s \in \{s_1, s_2, s_3, s_4, s_5\}$ is the*



(a) Price (b) Frequency (c) Global

Figure 5.6: Lift charts showing the increase in predicting performance obtained using product sophistication.

target shop. We want to build a classifier that, given some features of c and p, returns the value of s.

In other words, for each customer c and product p we know the shop s in which c usually goes to buy p, and we want to predict s using information about c and p. For each customer, the feature we calculated is the weighted average distance that c usually travels to buy all the products she needs. We used the formula:

$$\bar{d}(c_i) = \frac{1}{W(c_i)} \sum_{p \in P(c_i)} w_p \times d(c_i, s),$$

where w_p is the weight of product p, $P(c_i)$ is the set of products bought by customer c_i and $W(c_i) = \sum_{p \in P(c_i)} w_p$. We used three different classes of weights, based on the product price, quantity and sophistication, thus generating three attributes for each customer. We repeated the same procedure for the products, by using the same weighted average distance, using $C(p_j)$ (the set of customers buying product p_j) instead of $P(c_i)$. In the end, we obtained three features also for the product, based again on price, frequency of purchase and product sophistication.

Our dataset is heavily unbalanced on the larger shop, that attracts most of the purchases and contains products that are not present in any other shop. We then filter the data, to focus only on those cases where the prediction task is harder. For this reason, we defined three constraints that each entry in our test data has to satisfy.

Shop ID	Shop Type	Row Share
s_1	Iper	53.67%
s_2	Super	32.08%
s_3	Gestin	7.62%
s_4	Gestin	2.91%
s_5	Gestin	3.72%

Table 5.3: The distribution per shop of the filtered dataset for the classifier.

1. If the product is sold only in one shop, the prediction task is trivial. Thus, we want to consider only the products that are sold at least once in each of the five shops.

2. We consider only customers with a diversified shopping behavior. If the customers always went to the same shop, the prediction of its movements is trivial. For this reason, we select only the customers who purchase significant quantities of products in at least two different shops.

3. If a customer purchased the same product in two different shops we only kept the entry corresponding to the shop where he purchased the largest quantity of the product, as the classifier will output only one shop and therefore could not achieve a perfect accuracy.

The entries in our dataset, the triplets (customer, product, shop), satisfying all three constraints are $10,412,391$. In Table 5.3 we report the share of the rows of our filtered dataset whose target variable takes one of the possible five values, corresponding to the five shops. From Table 5.3 we know that we can build a naive classifier that always returns "s_1" as a result, and we would get an accuracy of 53.67%.

Given the size of the dataset, we extracted samples containing 5% of the entries (around $500,000$) and we performed our prediction tasks on these samples. The results we show are consistent in our samples. We created our classifier using the c4.5 algorithm [106]. To validate our results, we used the k-fold cross-validation method, by setting $k = 10$. We divided our data sample by putting two thirds of the data in the training set and the remaining data in the test set.

We depicted the lift charts of our classifiers in Figures 5.6(a-c). In the lift chart, on the x axis we have to total population fraction and on the y axis we have the population fraction that has been classified correctly. Since the correctly classified population has as upper bound the population itself, the perfect predictor that achieves a 100% accuracy is the bisector that goes from $(0, 0)$ to $(1, 1)$, and we depict it in all Figures with a blue line. The naive classifier, that always returns s_1 as result is depicted with a black line. The area between the blue line and the black line is where a model that improves over the baseline should lie.

In Figure 5.6(a) we consider as first model a classifier based on the product price. The red line shows that this classifier makes only an incremental improvement over the baseline, with an overall accuracy of 59.03%. We added the product sophistication information to this classifier (green line) showing a further accuracy improvement, ending up with an overall value of 65.87%.

We repeated the same analysis, this time using a classifier based on the frequency of purchase of a product. We can see that Figure 5.6(b) looks very similar to Figure 5.6(a): again the classifier based on on the frequency (red line) improves to an overall accuracy of 60.09%, while adding the product sophistication information (green line) bring to an overall accuracy of 67.91%.

We also point out in Figure 5.6(c) that a classifier including all the available information does not significantly improve the accuracy. Especially comparing to the frequency of purchase and product sophistication classifier (green line in Figure 5.6(c)), the increased accuracy of the classifier including also the price information (purple line) is very low. The overall accuracy of this model is 69.33%.

As a conclusion, we saw that the product sophistication adds significantly to the accuracy of the predictions based on price ($+6.84\%$) and on the frequency of purchase ($+7.82\%$). Adding the price information to this last classifier provides too a marginal improvement, but lower than the one provided by the product sophistication itself ($+1.42\%$). Therefore, the product sophistication is a very strong factor that not only explains on average customer movements, but can be effectively used to increase customer behavior predictions at the level of the single customers.

5.5 Conclusion

In this Section we addressed the problem of explaining and predicting customer behavior when shopping to large retailers. We showed that products have what we call a *range effect*: for some products, customers travel long distances, while for other products they settle down with the closest shop. We ruled out as possible explanations of this phenomenon the price of a product and the frequency of purchase. We introduced a new measure, namely the product sophistication, that is able to better explain customer movements: it is because products satisfy more complex needs, not because they are more expensive or they are needed less often, that customers travels more. We also showed as this additional information provides a significant boost of the accuracy in predicting in which shop a given customer will go buying a given product.

In this context there's place for many future developments. First, our prediction accuracy is good, but it may be improved, by using more sophisticated measures such as the radius of gyration [51, 94] of customers and products. Second, we analyzed a static snapshot of retail, but it would be interesting to analyze the evolution of customer behavior. Finally, following [58], to create a network of products based on the customers buying them may lead to further insights.

Chapter 6

A Complex Network Approach to Marketing Classification and Customer Profiling: Overlap versus Partitioning

This chapter is mainly based on [99].

What we presented in the previous two Chapters is a useful set of techniques to build a very complex structure over a retail market chain data (the bipartite graph customer-product). Actually, we can find very interesting insights from the data also using a less complex (but still complex) structure. For example two products can be connected if they are frequently co-purchased by the same customers [107]. The topological properties of this complex structure are informative about how customers perceive product relations, just like the collaborative filtering of Amazon and Netflix, but on a broader product typology set. For instance, sets of products may be very densely connected the one with the other, because customers always buy them together.

The task of finding sets of nodes densely connected in a complex network is known as “community discovery” [33]. Community discovery is one among the most prolific sub-branches of complex network analysis. Hundreds of papers have been written on the subject, and dozens of algorithms have been proposed to solve it. As a result, the scientific community has come to agree that there is no unique solution to community discovery, given the many different possible definitions of “community” that can be accepted in different applications.

In particular, one of the most important distinction between community discovery algorithms is about a node’s membership to a community [45]. In some methods, a node is forced to belong only to the community it is closest to. This is a partition approach to community discovery (often called “hard clustering”, or “disjoint community discovery”). In other methods, a node may be free to join as many communities as necessary. If an algorithm allows this type of output, then it is said to return overlapping communities (also known as “soft clustering” or “community coverage”).

Historically, overlapping algorithms were developed as a critique to the partition approach. The theory was that “actual communities” are overlapping, and therefore the partition approach was obsolete. Our point is that the two approaches are not mutually exclusive. In the same context, they yield different results because the problem they address is different and there is no “better” or “worse” method.

The aim is to investigate the typology of results that different approaches to community discovery can achieve while analyzing a complex network of products. When applied to a network created connecting products if they are co-purchased by the same customers, community discovery will return groups of products that are “related” to each other. Our aim is to understand what “related” means under different community definitions, in particular when our aim is to find a community partition *vis-à-vis* when our aim is to find an overlapping community coverage.

To prove our point, we collected data about more that 24 thousand products, co-purchased by a million customers in more than 80 millions shopping sessions from four regions in the center of Italy. Using their purchases, we created a product-product network connecting products if they were co-purchased during the same shopping session. We then applied two state-of-the-art community discovery algorithms on this structure: one yielding a disjoint community partition (Infomap [110]), the other yielding an overlapping community coverage (Hierarchical Link Clustering [4]).

Our results confirmed that the different community definitions returned two very different sets of results. We observed that there is no clear ranking in the quality of these results, i.e. there is no clear way to determine which algorithm performed "better". On the other hand, both the partition and the overlapping approach returned results that can be utilized for different tasks. The disjoint communities proved to be useful for the redefinition of the product marketing classification. The overlapping communities, instead, represent specific customer behaviors, and therefore provide useful data for the task of customer profiling.

To sum up, the contributions of this Section can be summarized as follows:

- To the best of our knowledge, this is the first empirical test able to provide an insight about the practical usefulness of different community discovery approaches in a real-world analytic scenario, in particular about the difference between a partition approach versus an overlapping approach. As a consequence,
- We showed how partition-based community discovery is useful as a novel approach to the construction of marketing, and possibly general purpose, classifications;
- We provided a novel approach to customer profiling, via overlapping community discovery of product co-purchase networks.

6.1 Data Preparation

As claimed in Chapter 3, the dataset contains information about $439,619$ different objects sold in the shops. The main bias introduced by this huge cardinality of products are two:

1. we have information about products that are meaningless for our purposes (e.g. shoppers, discount coupons, etc.), and
2. due to some exception, we have distinct products that have the very same semantic (e.g. 6-bottles regular coca cola box and 6-bottles regular coca cola box with Santa Claus in the package for Christmas time).

We solved (1) by filtering data using semantic information in the marketing hierarchy. We solved (2) by including in our analysis dataset at most the top 5 sold products (or less, if there are not enough products) for each marketing Segment. Notice that, for each item exception, there always is a product that is top seller over the others with the same semantic. After this filtering phase, the dataset contains $26,862$ different items, that are the nodes of our network, belonging to $5,510$ marketing Segments.

We now want to connect these nodes, to create the product complex networks. Given the big amount of data considered, almost all products have been sold at least once with all the other products. We need to filter out these connections, to focus only on relevant and significant relationships. To this end, we discovered all possible pairs of products sold together (using Apriori [1]), and we calculated, for each of them, the *lift* measure (in a very similar way than in Chapter 4). We recall the definition:

$$\text{lift}(X, Y) = \frac{\text{supp}(X, Y)}{\text{supp}(Y) \times \text{supp}(X)}$$

where X and Y are products, and $\text{supp}(i)$ is the number of baskets containing the item i divided the total number of the baskets in the dataset. $\text{supp}(i)$ is the "Relative Support" of i, i.e. the observed likelihood of having i in a basket. Lift measures how much a pair of items is interesting, calculating how its distribution is related with the distribution of the single items. If lift is equal to 1 we are under the hypothesis of stochastic independence, and the greater lift is, the greater the occurrence rate of the pair is significant.

Since lift is a relative measure, we need also to take under control the popularity of the products composing the pair. In fact, the supports of

	1	2	5	8	10
10	20,042,602	11,784,927	1,962,699	825,644	577,954
50	9,141,753	5,356,930	640,933	264,156	187,341
100	5,874,000	3,433,601	376,367	160,049	115,033

Table 6.1: Number of edges in the product network after filtering with minimum absolute support (rows) and lift (columns).

	1	2	5	8	10
10	16,910	16,769	16,152	15,402	14,949
50	12,268	12,035	11,401	10,797	10,392
100	10,578	10,347	9,734	9,158	8,784

Table 6.2: Number of nodes in the product network after filtering with minimum absolute support (rows) and lift (columns).

the single items composing the pairs are in the denominator of the lift formula, and multiplying each other. This implies that the smaller the supports are the greater the lift is inflated, by exaggerating the relevance of products rarely sold and thus not really meaningful. For this reason, we also use the "Absolute Support" of the pair, measuring how many are the occurrences of the couple in the dataset, i.e. the number of baskets containing both products.

To sum up, lift tells us how interesting the pair occurrence is, the absolute support tells us how relevant the pair occurrence is. A pair to be included in our network has to be interesting and relevant at the same time. In tables 6.1 and 6.2 we show the cardinalities of the edge set and the node set of different product networks, built using different thresholds on absolute support (in rows) and lift measure (in columns) of the pairs.

To obtain a manageable network, with edges representing associations not very infrequent but strongly reliable, we chose to set the minimum absolute support at 10 and the minimum lift at 10. The resulting product network contains $14,949$ nodes and $577,954$ edges, and that is the network we use hereafter, for our case study in Section 6.3.

6.2 Community Discovery

The branch of Community Discovery in network science is very prolific, with hundreds of papers proposing new approaches to the detection of network communities [33]. Given the extensive attention on the subject, two issues have been deeply studied. The first is the notion that there is not a single best method to extract communities from complex networks. It is possible to define "communities" in different ways and different approaches are more or less efficient for a particular community definition [33].

The second issue is the one at the center of investigation of this part of the thesis. The assumption that communities are dense subsets of nodes isolated from the rest of the network has been questioned. There is a growing evidence that communities are not really isolated from the rest of the network, but rather overlap the one with the other, sharing nodes. Given this issue, two mutually exclusive approaches to community discovery can be implemented: the partition approach, that follows the main assumption here presented; and the overlap approach, that allows nodes to be classified in more than one community.

It is one of the assumption here that both approaches can yield enlightening, and different, results even in the very same network. For this reason, we briefly present the two community discovery algorithms, that we use in our case study of analysis of a co-purchase product network. In Section 6.2.1 we present Infomap, a community discovery algorithm employing the partition approach; and in Section 6.2.2 we describe the Hierarchical Link Clustering (HLC), an overlap community detector.

6.2.1 Partition Approach

As we discussed, in the partition approach the main assumption is that densely connected nodes are separated from the rest of the network by nodes with sparser connections. We show a simplified example of this assumption in Figure 6.1(a). In Figure 6.1(a) we clearly need a partition of the graph, separating nodes 0 to 4 from nodes 5 to 9 and from nodes 10 to 14. As a consequence, algorithms seeking a node partition have to minimize the number of edges between communities, while maximizing the number of edges inside communities. Many non-trivial measures have been proposed. One of the proved most successful is the compression factor allowed by the partition, and it has been proposed in the Infomap algorithm.

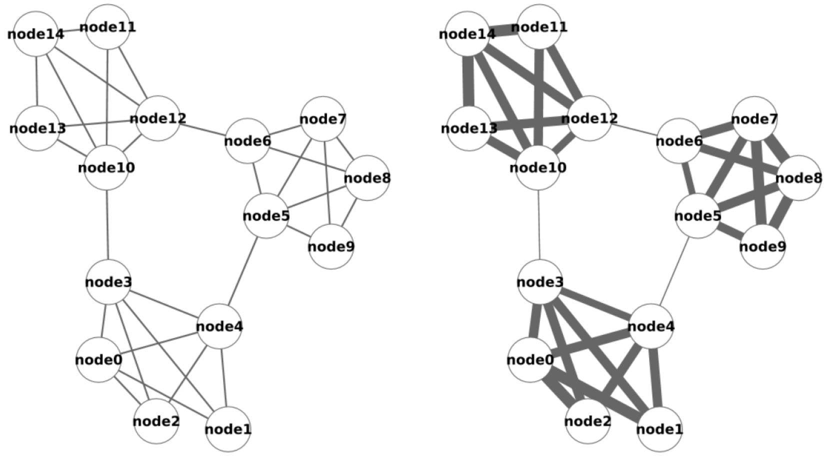
The Infomap algorithm [110] is based on a combination of information theoretic techniques and random walks. It uses the probability flow of random walks on a graph as a proxy for information flows in the real system and decomposes the network into clusters by compressing a description of the probability flow. The algorithm looks for a cluster partition M into m clusters so as to minimize the expected description length of a random walk.

In Figure 6.1(b) we have depicted the same example of Figure 6.1(a) where the edge width is proportional to the amount of redundant information shared by the two connected nodes.

The intuition behind the Infomap approach for the random walk compression is the following. The best way to compress the paths is to describe them with a prefix and a suffix. Each node that is part of the same cluster M of the previous node is described only with its suffix, otherwise with prefix and suffix. Then, the suffixes are reused in all prefixes, just like the street names are reused in different cities. The optimal division in different prefixes represent the optimal community partition. We can now formally present the theory behind Infomap. The expected description length, given a partition M, is given by:

$$L(M) = qH(Q) + \sum_{i=1}^{m} p_i H(P_i).$$

$L(M)$ is made up of two terms: the first is the entropy of the movements between clusters and the second is entropy of movements within clusters. The entropy associated to the description of the n states of a random variable X that occur with probabilities p_i is $H(X) = -\sum_1^n p_i \log_2 p_i$. In (1) entropy is weighted by the probabilities with which they occur in the particular partitioning. More precisely, q is the probability that the random walk jumps from a cluster to another on any given step and p_i is



(a) Toy example of the fundamental assumption of a partition algorithm.

(b) How Infomap sees the network structure using random walks.

Figure 6.1: Example of the partition approach to community discovery.

the fraction of within-community movements that occur in community i plus the probability of exiting module i. Accordingly, $H(Q)$ is the the entropy of clusters names, or city names in our intuition presented before, and $H(P_i)$ the entropy of movements within cluster i, the street names in our example, including the exit from it. Since trying any possible partition in order to minimize $L(M)$ is inefficient and intractable, the algorithm uses a deterministic greedy search and then refines the results with a simulated annealing approach.

6.2.2 Overlapping Approach

The overlap class of algorithms rejects the fundamental assumption of the partition approach. Here, nodes are allowed to be in multiple communities, therefore they are densely connected also to nodes that are not part of the community, removing the sparser areas of the network outside the community. A simplified example representing this concept is depicted in Figure 6.2(a). Here, nodes are grouped in cliques of 6 nodes, connected the one with the other by cliques of three nodes. So the nodes $\{0, 1, 2\}$ form a 3-clique with each other and two separated 6-cliques with nodes $\{3, 4, 5\}$ and $\{9, 10, 11\}$. Similar structures are generated by all other 3-node groups on the diagonals.

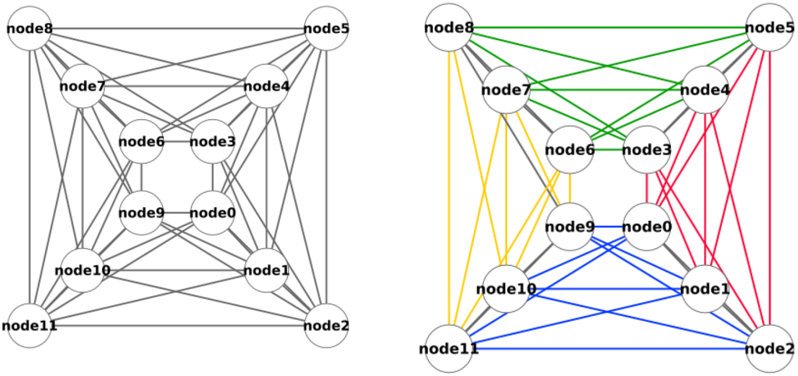
It appears clear that even in this simple case there is no reasonable partition of the graph. There is no reason for which we should prefer clique $\{0, 1, 2, 3, 4, 5\}$ over clique $\{0, 1, 2, 9, 10, 11\}$, and we cannot merge them either, ignoring the fact that the other nodes are densely connected to them too. That is when an overlapping approach like HLC [4] proves its usefulness.

HLC assumes that communities should group together edges, not nodes. The relationship is part of a community and the node is part of all the communities its relationships are part of. In the case of a social network, a person knows other people for one main reason (work together, study together, spend together the free time, and so on) and therefore she is part of a different community for each "relationship environment". As a consequence, these communities overlap. Figure 6.2(b) depicts an example of HLC output for the graph presented in 6.2(a): each link is colored according to the link cluster it belongs to, and therefore we obtain as communities both $\{0, 1, 2, 3, 4, 5\}$ and $\{0, 1, 2, 9, 10, 11\}$.

For an undirected, unweighted network, we denote the set of node i and its neighbors as $n_+(i)$. HLC considers only link pairs that share a node, under the assumption that they are more similar than disconnected pairs. The similarity S between links e_{ik} and e_{jk} in the set E of all links in the network is computed as:

$$S(e_{ik}, e_{jk}) = \frac{n_+(i) \cap n_+(j)}{n_+(i) \cup n_+(j)}.$$

Shared node k does not appear in S because it provides no additional information and introduces bias. This is basically the jaccard index of the set of nodes one step away from edges e_{ik} and e_{jk}. HLC then builds a link dendrogram from the presented equation (ties in S are agglomerated simultaneously). The dendrogram is cut at a S threshold that maximizes a quality function called "partition density". For each community, the



(a) Toy example of the fundamental assumption of a overlap algorithm.

(b) How HLC sees the network structure using node Jaccard.

Figure 6.2: Example of the overlap approach to community discovery.

partition density is defined as:

$$D_c = \frac{m_c - (n_c - 1)}{n_c(n_c - 1)/2 - (n_c - 1)},$$

where m_c is the number of links in the community c and n_c is the number of induced nodes in the community ($n_c = \bigcup_{e_{ij} \in c} \{i, j\}$). The overall partition density of a given set of link partition is the average of all partition densities, normalized over the total number of edges in the network:

$$D = \frac{2}{|E|} \sum_c m_c \frac{m_c - (n_c - 1)}{(n_c - 2)(n_c - 1)}.$$

6.3 Case Study

We now take a look at the characteristics of the results provided both by the partition and by the overlap approach. We consider the following list of characteristics:

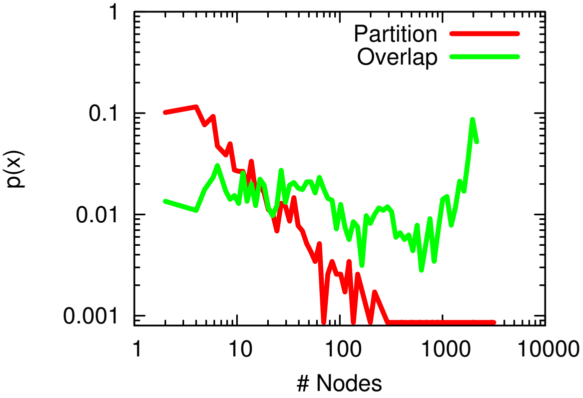


Figure 6.3: The log binned distribution of the number of nodes per community, both for the partition and the overlapping approach.

- The distribution of community size, to understand if one method privileges larger or smaller communities, in Section 6.3.1;
- The community entropy w.r.t. the marketing classification, i.e. if the results of an algorithm are substantially overlapping with the known product classes, in Section 6.3.2;
- Some community extracts, to provide examples of the typical communities returned by an algorithm, in Section 6.3.3.

We then put together the discovered differences of community results in Section 6.3.4.

6.3.1 Community Size

We start by providing one of the most basic information about a community coverage: the distribution of the community size, i.e. the number of nodes per community. The distribution is depicted in Figure 6.3. On the x axis we report the number of nodes and on the y axis the probability that a community contains the given number of nodes, both for Infomap (red line) and HLC (green line). The distribution is log binned, i.e. each x axis value is grouped in bins of increasing size.

As we can see, the two distributions have different asymptotic behavior. Infomap provides a community size distribution that resembles a power-law, with more than 30% communities containing 4 nodes or less, and one community containing more than 3000 nodes. On the other hand, HLC communities have a very different size distribution: just above 3% of communities have 4 nodes or less, and 10% of them have around 2000 nodes. So the first difference between the two approaches can be summarized as: “The overlap approach returns larger communities than the partition approach”.

6.3.2 Community Entropy

We now want to describe what is the actual content of these communities. In particular, we are interested in how homogeneous the communities are w.r.t. the marketing classification of the supermarket. In practice, we want to know if in a given community we grouped the products that belong to the same marketing classification. For the marketing classification, we use the segment level, as presented in Chapter 3. A good measure to do this is to calculate their information entropy. The information entropy is formally defined as the average unpredictability in a random variable, which is equivalent to its information content. In our case, a community c of $|c|$ nodes is viewed as $|c|$ outcomes of a random selection of a marketing classification. The possible outcomes of the extraction are $|M|$, the number of marketing classifications. The information entropy of a community $c \in C$ is then calculated as:

$$H(c) = -\sum_{m \in M} p(c_m) \log_2 p(c_m),$$

where $p(c_m)$ is the number of nodes in the community c that belongs to the marketing category m, over the total number of nodes inside community c. The average entropy value, calculated for all communities in C that is the community set, for Infomap is 1.8302, while the average for HLC is equal to 5.66305. The average entropy could not be an accurate

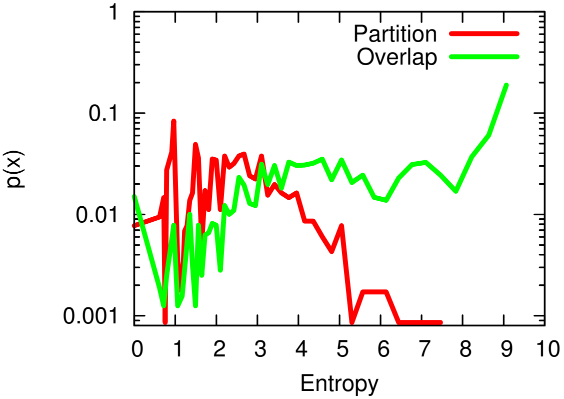


Figure 6.4: The log binned distribution of the entropy per community, both for the partition and the overlapping approach.

information, as it may be driven by extreme values. For this reason, we depict in Figure 6.4 the probability (y axis) that a given community takes a given entropy value (x axis) for the communities extracted by Infomap (red line) and by HLC (green line). Again, the distributions are log binned.

Also in this case, the entropy distribution for Infomap and HLC look different. Most communities returned by Infomap have entropy lower than 3, and the number of communities with entropy larger than 6 is not significant. On the other hand, the majority of HLC communities have entropy larger than 3, with 20% of the communities having an entropy around 9.

One could think that the higher entropy of the HLC communities is due exclusively to the fact that HLC communities are larger on average. To disprove the objection, for each community of size $|c|$ we normalize the obtained entropy value over the description length required to code a random community, that is $\log_2 |c|$. We then sum up all the normalized values and take the community average, as:

$$\bar{H}(C) = \frac{1}{|C|} \sum_{c \in C} \frac{H(c)}{\log_2 |c|}.$$

The $\bar{H}(C)$ values are interpreted in the following way. If $\bar{H}(C) = 1$, then it means that, on average, for each community c it holds $H(c) = \log_2 |c|$, i.e. the distribution of marketing classifications in the community is practically random. If $\bar{H}(C) = 1$, then for each community c we have $H(c) > \log_2 |c|$, then the communities separate products of the same marketing category even if it would be expected to find them in the same community. If $\bar{H}(C) < 1$, then on average we find products of the same marketing category in the same community.

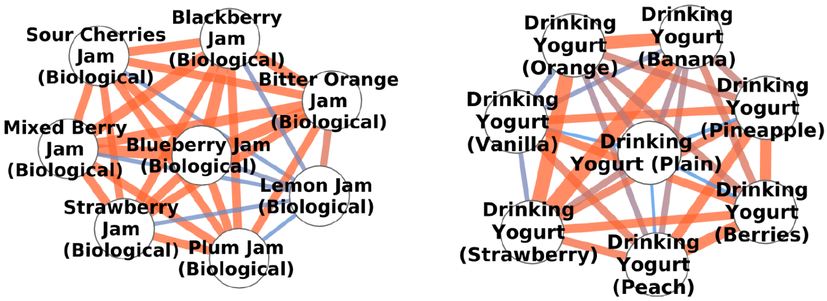
The lower the $\bar{H}(C)$ value, the more homogeneous the communities are on the marketing classification, independently on their number of nodes. We found that in Infomap $\bar{H}(C) = 0.60180796763$, while for HLC $\bar{H}(C) = 0.814926005465$. We can conclude that HLC communities have a 20% higher entropy than Infomap, independently on community size. So the second difference between the two approaches can be summarized as: "The overlap approach returns communities that contains more diverse typologies of products than the partition approach".

6.3.3 Community Extracts

The aim of this section is to provide some concrete instances of the findings described in the previous subsections. We provide two examples of small communities extracted using the Infomap partition method and two examples of communities extracted with the HLC overlap community detection.

Figures 6.5(a) and 6.5(b) are the two extracts from the Infomap community partition. Given that most Infomap communities are small (see Figure 6.3) it is easy to find representative communities of limited size. In this case, we limit ourselves to communities containing 8 nodes. These two communities are identified as community #37 and #80 respectively.

In Figures 6.5(a) and 6.5(b), the node color refers to the node marketing Segment (see Chapter 3). Colors are not consistent across Figures, i.e. even if nodes from different Figures have the same color it does not mean they are in the same segment. Edge width and color are proportional to edge weight, that is the number of times the two products were bought in


Blackberry Jam (Biological)
Sour Cherries Jam (Biological)
Bitter Orange Jam (Biological)
Mixed Berry Jam (Biological)
Blueberry Jam (Biological)
Lemon Jam (Biological)
Strawberry Jam (Biological)
Plum Jam (Biological)
Drinking Yogurt (Orange)
Drinking Yogurt (Banana)
Drinking Yogurt (Pineapple)
Drinking Yogurt (Vanilla)
Drinking Yogurt (Plain)
Drinking Yogurt (Berries)
Drinking Yogurt (Strawberry)
Drinking Yogurt (Peach)

(a) Community #37.

(b) Community #80.

Figure 6.5: Extracts from the non-overlapping communities.

the same shopping session. In Section 6.1 we refer to this quantity as "Absolute Support" of the pair and the minimum value is equal to 10, i.e. each connected pair of products in the community has been sold at least 10 times. In the edge color map, orange indicates high weight, blue indicates low weight.

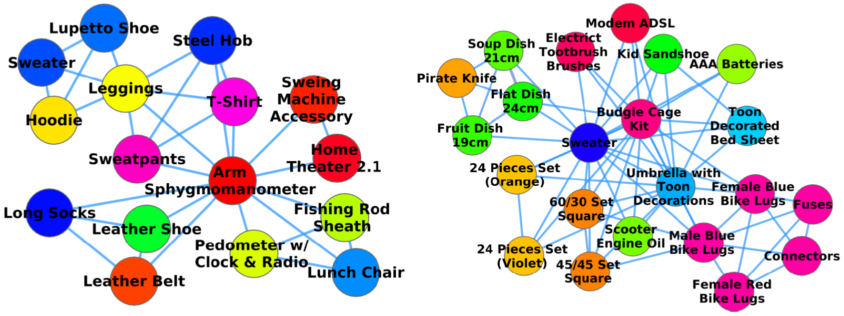
By inspecting communities #37 and #80 we can observe the following characteristics:

- The communities are very dense: in fact, they are cliques, where every node is connected to every other node, meaning that different customers have bought all possible combinations of these products at least once;

- Most links have high weight, meaning that the amount of customers buying these products in the same shopping session is high;

- All products in the communities are part of a very homogeneous class of products: in community #37 we have only biological jams, while in community #80 we have only liquid yogurts.

We now turn to examine communities extracted with the overlap HLC approach. They are depicted in Figures 6.6(a) and 6.6(b) and they are identified with IDs #239 and #590. Again, the edge width and color is



(a) Community #239.

(b) Community #590.

Figure 6.6: Extracts from the overlapping communities.

representative of edge weight in the same scale of Figures 6.5(a) and 6.5(b), while node color indicates the marketing segment and it is not consistent across Figures, due to the high amount of segments present in each community. We had to choose communities with a larger number of nodes, given the relative scarcity of small communities returned by HLC (see Figure 6.3).

By inspecting communities #239 and #590 we can observe the opposite characteristics we observed for the Infomap communities:

- The communities are dense but they are not cliques: they are rather cliques joint together by some products (for example, the arm sphygmomanometer plays a central role in community #239);
- Almost all links have low weight, meaning that the amount of customers buying these products in the same shopping session is low;
- Almost all products in the communities are part of a different marketing segment.

So we can summarize the results of this section by saying that: "examples shows that characteristics and topology of communities returned by the overlap approach are very different from the results of the partition approach”.

6.3.4 Community Interpretation

In this section we wrap up the results we presented in the previous sections, providing a tentative explanations that takes into account all of them. The conclusions of each section were:

- The overlap approach returns larger communities than the partition approach;
- The overlap approach returns communities that contains more diverse typologies of products than the partition approach;
- Examples shows that characteristics and topology of communities returned by the overlap approach are very different from the results of the partition approach.

Our explanation is then that the overlap approach mostly reflect customer behaviors and possible expansions of them, while the partition approach returns a refined marketing classification. We support our explanation by noticing that customers usually buy products for different marketing segments because they have to satisfy different needs, this also implies that a community grouping together a “customer profile” should be larger and more diverse on marketing classifications. Being less dense, overlap communities also put together products that some customers bought together and some others, who bought similar products, did not buy together, identifying possible customized product suggestions to the marketing department.

On the other hand, the partition approach is more homogeneous on the existing marketing classification, but the disagreement points may be interesting to explore to refine it. The small communities suggest a fine grained marketing description and the high density and edge weight of them implies that the products have really something to do with each other. It is worthwhile to notice that Infomap can also be used with a hierarchical approach, by merging related communities at different levels. In this way, it is possible to reconstruct a full marketing hierarchy. Also HLC by nature returns a hierarchy, but of a different kind: it is a hierarchy of extended customer profiles, with different, but nevertheless useful, classification of customers' behaviors and sub-behaviors.

6.4 Conclusion

In this Section, we investigated the different application scenarios that it is possible to tackle with community discovery using different community definitions. We focused on the analysis of networks of products co-purchased in a supermarket. In particular, we have showed that there is not a quantitative difference in how good or bad are the results obtained by searching for a disjoint community partition and the results obtained from an overlapping coverage search. There is rather a qualitative difference, i.e. different problem definitions. A partition approach has proven to be useful as an approach to a marketing product classification. An overlap approach, instead, can shed some light over a novel technique for customer profiling.

The present paper can be extended along several other lines of research. First and foremost, the distinction between partition and overlap based approaches is just one of the many in the field of community discovery. Some review works [33] have come as far as to identifying more than seven macro definitions of communities in complex networks. It is possible that each of these definitions is going to provide answers for additional problem definitions. As a second point, the empirical study about the different practical applications of alternative methods of community discovery can be separated from the application scenario we considered in this paper. Instead of focusing on product networks, we can consider many different typologies of networks, covering a wider set of kind of markets, human activities and natural phenomena.

Part II

Exploiting the Social Networks for Applications in Markets

Chapter 7

"You Know Because I Know": a Multidimensional Network Approach to Expert Finding Problem in the Market of Skills

This chapter is mainly based on [36].

Finding talents is one among the most difficult challenges for organizations. Hiring talents means performing better, get more revenue and evolve the business. Where hiring talents is relatively easy, the biggest challenge for organizations today is to find talents they have already hired: finding and creating knowledge is important, but so it is to be able to search and mine knowledge that is already owned. These complex requirements (that are essential for companies in order to *"stay in the market"*) create an interplay between demand and offer where as quick a company can find her own "expert", as more chanches she's gonna have to survive and/or succeed. The social networking revolution allowed the creation of tools to evaluate competencies, expertise and skills, like Linkedin[1]. Before social networks, talent management activities were restricted to a marketing oriented approach: to discover talents among their employees, organizations promote internal contests or invest into assessment activities and campaigns. However, during the last few years, organizations started also to put efforts on social talent management, often connected to wider social related initiatives (social CRM, enterprise social networking, etc.). Social talent management is nowadays based on dedicated pages or applications whose aim is to discover interesting professionals. Social networks are also used by organizations to get unofficial information about their employees or candidates, to understand what they do and who they really are.

To understand where and how an employee is positioned on a skill network will enable organizations to find previously hidden sources of knowledge, innovation and know-how. Resumes can provide structured information about studies and working experiences, but they are not useful to understand skills and experiences that do not belong directly to the employee, but to his friends and colleagues. If a candidate does not master a topic but has a strong relationship with somebody who does, then he is an important gateway in that topic, although different relationships allow for different gateway values (i.e. two friends in the same company represents a stronger connection than two friends in competing companies).

If an employee is involved on specific tasks, and she has always been involved only on those tasks, this does not mean she could not have strong competencies and skills on completely different subjects: hobbies, passions, interests can be worth some, sometimes a lot of, value in the *prosumers* age. People are digitally involved throughout their life and there is not a clear separation between personal and professional network. Understanding the position of somebody among different skills and knowledge networks or rankings can let this value emerge. This data

[1] `http://www.linkedin.com/`

richness and hidden knowledge demands for a multidimensional and multiskill approach to the employee ranking problem: the definition of a ranking algorithm on networks, able to capture the role of different kinds of relations and the importance of different skill sets.

Given a person with a set of skills and a neighborhood of friends in a social network, the skills of the friends to some extent are accessible through that person, and therefore they should be considered when evaluating her. More formally, each node n in a network has some skills S each with a given intensity, and it is connected, with different kinds of edges, to other nodes ($n_2, n_3, ...$) with their own skill set ($S_2, S_3, ...$). The real value of node n is then defined by a function f that takes as input not only S, but also $S_2, S_3, ...$, by accounting for the different dimensions connecting n to $n_2, n_3, ...$. This idea has been proven in the economic field at the macro level: in [88] the author proved that the social return of higher skill levels is higher than the personal return, i.e. higher skilled people make their colleagues to be more valuable as well.

Current ranking algorithms can only provide multiple rankings on monodimensional networks, or simple rankings in multidimensional networks. Herer we propose an algorithm whose aim is to provide multiple rankings (one for each skill) for nodes in a multidimensional network, a network with multiple types of edges. Our approach is called "you (U) know Because I Know", or UBIK. We test UBIK on real world networks, showing that its ranking is less trivial and more flexible than the current state-of-the-art methods.

Our contribution can be then summarized as follows. We introduce a novel ranking algorithm for multidimensional networks with multiple node attributes, able to provide a different ranking for each skill. We provide a fast implementation able to scale linearly in the number of edges of the network. We provide a new ranking for real world scenarios, including co-authorship in computer science and corporate emails.

7.1 Network-Based Human Resources

7.1.1 Problem Definition

Our problem definition is the following:

Definition 7.1 *Let $G = (V, E, D, S)$ be a multigraph where each node $v \in V$ is connected to its neighbors through multiple edges $e \in E$, each carrying a label $d \in D$; and S be a skill set, such that each node v is labeled with one or more skills $s \in S$, each with a given weight $w \in R^+$. Given a query q containing a set of skills $S_q \subseteq S$ and the importance $r(d) \forall d \in D$, we want to rank nodes accordingly to the weight of each $s \in S_q$ they posses directly or indirectly through their connections.* ▮

The intuition behind our idea is the following. Suppose we have a set of people, each with her own skills and acquaintances, and a task to be performed. In a world without social knowledge interaction, the best way to perform the task is to assign it to the person, or to a set of people (i.e. a team), possessing the highest value of the related skill. However, each person can access to the external knowledge of their acquaintances, thus possibly modifying her skill set value, and therefore the decision of the composition of the team. Each person can also access to the acquaintances' acquaintances skills, but with an increasing cost at each degree of separation, causing at some point the external skill to be useless.

Now, we need to define the social connections, and their different types. We need to formally define the skills carried by each individual as the initial state of the system and how the expertise propagates through social connections.

7.1.2 The Model

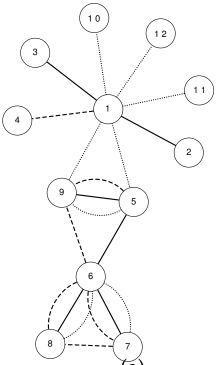
Following [13] we model our problem with a multidimensional network (also studied in [18]). We now present the basics of multidimensional networks and then the extensions we apply to the basic model.

In the case of a multidimensional setting, a convenient way to model a network is a *labeled multigraph*. Intuitively, a labeled multigraph is a graph where both nodes and edges are labeled, and where there can exist two or more edges between two nodes. Just as any regular labeled graph, also labeled multigraphs may be directed and undirected, thus we allow edges to be both directed and undirected, when the analytic aim requires it. Such a graph is denoted by a triple $G = (V, E, D)$ where: V is a set of nodes; D is a set of labels representing our dimensions; E is a set of labeled edges, i.e., it is a set of triples of the form (u, v, d) where $u, v \in V$ are nodes and $d \in D$ is a label. We assume that given a pair of nodes $u, v \in V$ and a label $d \in D$ it may exist only one edge (u, v, d).

In [13], the previous paragraph fully describes a multidimensional network. We need to add several things in order to fit into our problem definition. First, we need to introduce weighted node labels. In other words, each node $v \in V$ is a collection of couples in the form (s, w) where s is the label and $w \in R^+$ is the value of label s for node v. Therefore, $v = \{(s_1, w_1), (s_2, w_2), \dots, (s_n, w_n)\}$. The set of node labels describes what in our data are the skills of the node, along with their value. The set of all possible skills is fixed for the network, and we refer to it as S.

In [13], dimensions are considered distinct but equal. In our case, each dimension can have a different importance: in the real world a friendship tie may be more or less strong than a working collaboration, given the social environment where this tie may play its role. Therefore, each dimension $d \in D$ is represented not only by its label, but also by a value $r(d) \in R^+$, quantifying how much relevant is a relation expressed in dimension d, according to the query requested.

At this point we have all the building bricks of the static part of our model. Next, we define how the knowledge exchange dynamics takes place in the model itself, generating the flow that allows us to rank the nodes in the network. The idea is that each node passes the entire set of its skills to each one of its neighbors. This procedure is similar to the one employed by the classical PageRank formulation, but with a few distinctions. First, our method is not based on random walks, but on the percolation of the various skills without random jumps. Second, the amount of skill value passed to the neighborhood of a node is not equally divided and assigned to each neighbors. The amount of value that node u passes to the neighbor v is proportional to the importance of



(a)

Node	Skill-Value
1	$(a, 115), (b, 0), (c, 0), (d, 0)$
2	$(a, 100), (b, 0), (c, 0), (d, 0)$
3	$(a, 100), (b, 0), (c, 0), (d, 0)$
4	$(a, 100), (b, 0), (c, 0), (d, 0)$
5	$(a, 100), (b, 0), (c, 0), (d, 0)$
6	$(a, 0), (b, 0), (c, 0), (d, 80)$
7	$(a, 0), (b, 90), (c, 20), (d, 0)$
8	$(a, 0), (b, 20), (c, 90), (d, 0)$
9	$(a, 100), (b, 0), (c, 0), (d, 0)$
10	$(a, 100), (b, 0), (c, 0), (d, 0)$
11	$(a, 100), (b, 0), (c, 0), (d, 0)$
12	$(a, 100), (b, 0), (c, 0), (d, 0)$

(b)

Figure 7.1: Our toy example. (a) The multidimensional network structure (each line style represents a different network dimension; (b) The skill table, recording the values of each skill (a, b, c and d) for each node.

the dimensions connecting u and v. It is also inversely proportional not only to the degree of the node passing the skill, but also to the degree of the node receiving the skill. Third, the knowledge exchange may be or may be not mutual, i.e. we are not narrowing our model only to directed graphs, but to general graphs.

We use also the range parameter α, commonly used for centrality scores, handling the following situation. If u and v are connected through a node v_2, then the amount of knowledge they exchange is lower than a direct connection, just like the resistance loss in an electric circuit. If x is the amount of value passed by a direct connection, the amount of skills received from nodes ℓ degrees away is corrected as $x^{\frac{1}{\ell\alpha}}$. Traditionally, directly connected nodes should be at zero degrees of separation, because no other *nodes* should be crossed to reach them. For practical purposes, in this paper we assume $\ell = 1$ for neighboring nodes, as we need to cross one *edge* to reach them. The introduction of α is due both to logical and practical reasons. Logically, it makes our model more realistic, as a person is a gateway of her friends' skills, thus she is to some extent also a gateway for her friends' friends' skills, but of a much lower importance. Without the α parameter the skill percolation could potentially continue indefinitely, and the computation may not be able to stop.

In Figure 7.1 we represented a simple toy example. The network structure of social connections is depicted in Figure 7.1(a), while Figure 7.1(b) is the skill table associated with the structure. From the skill table we know that all nodes start with some global skill value (not equal for everybody, as it happens in reality), distributed along four skills (a, b, c and d). The social connections do not have all the same value: solid line has a 50% efficiency in the knowledge transfer, dashed line has a 33% efficiency, while the dotted line has only a 17% efficiency. By looking only at the network structure, node 1 is the most central node. It also has the highest global skill value. According to the closeness centrality, also nodes 5 and 9 are more central than 6. However, our algorithm will propagate skills a, b and c to 6 with the maximum efficiency, while 6 will retain also its unique d skill. At the end of the process, node 6 ends up as the most valuable node in general in the network, while node 1 can only specialize in skill a. With relaxed values for the α parameter (like $\alpha = 1$) node 1 can still get some parts of skill d ($\sqrt{\frac{1}{6}\frac{1}{2}80}$ from node 5 and $\sqrt{\frac{1}{6}\frac{1}{3}80}$ from node 9, that is ~ 4.69017) and even less of b and c. If $\alpha = 3$ then the contributions to node 1 of skills different from a is negligible.

7.1.3 The Data

Our model is describing, according to our hypothesis, how knowledge flows in a face-to-face social environment, following the proven macro level mechanism of the social effect of schooling [88]. However, the data about the face-to-face interactions are usually part of the tacit realm of knowledge. If we cannot find direct or proxy data sources about these interactions, any algorithm solving the problem of evaluating people on the basis of our hypothesis is practically useless. In this section, we present how we use two real-world datasets, adapting them to our model and problem definition, and providing an interpretation of the knowledge that our model can unveil. Of course in both cases we are in front of an approximation. Table 7.1 provides the general statistics about the extracted networks for each dataset.

Network	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert D\rvert$	$\lvert S\rvert$	$\lvert D_r\rvert$
DBLP	38,942	100,983	16	50	1,187
Enron	5,913	49,058	7	8	0

Table 7.1: The statistics of the extracted networks.

DBLP[2] is an online bibliography containing information about scientific publications in the field of computer science. Using the data from this dataset, our problem definition may be adapted as follows: we want to evaluate the actual knowledge possessed by scientific authors in different topics and sub-topics, focusing on different branches of their disciplines. Being our aim to rank authors, they should be the nodes of our network. The link is the co-authorship relation: two authors are connected if they have written together a paper. The dimension of the connection should represent the "quality" of their relation, in this case the venue where the publication appeared (we chose 16 top-tier conferences in computer science, including VLDB, SIGKDD, CIKM, ACL, SIGGRAPH and others). The set of skills should describe the expertise of the author, therefore we chose to represent them as the keywords used in their publications. We eliminated stopwords, we applied a stemming algorithm [104] on the remaining words and then we selected the 50 most commonly used keywords in a paper title. The number of times author u used the keyword s is used to evaluate how much the author considers himself an expert over s, i.e. it is used as its w value for s.

Enron dataset[3] is a collection of publicly available emails exchanged by the employees of the energy company, distributed after the well known bankruptcy case. We are interested in ranking the employees, that are the nodes of our network. With this network we are able to unveil who are the real knowledge gateways in an organization, by looking at the internal communication even in the absence of more structured social information (see Section 7.3.2 for the results). Therefore, to apply our algorithm is not necessary for an organization to actually create a social media platform for their employees (or to download information from other social media). We

[2] http://www.dblp.org/db/

[3] http://www.cs.cmu.edu/~enron/

Algorithm 1 The pseudo-code of UBIK.

```
Require: $G = (V, E, D, S); \alpha \in [0 \dots \infty]; r(D)$
Ensure: Node set $V$ with updated skill values.
$\ell \leftarrow 1$
while $\ell < \delta$ do
    for all $u \in V$ do
        for all $v \in N(u)$ do
            for all $s \in S$ do
                $w'_{u,s} \leftarrow w_{u,s} + \sum_{d \in D} \frac{(f(v,s) \times r(d))^{\frac{1}{\ell\alpha}}}{|N(u)|+|N(v)|}$
            end for
        end for
    end for
    $UPDATE(V, u'(s))$
    $\ell \leftarrow \ell + 1$
end while
$NORMALIZE(V)$
return $V$
```

took only the email addresses ending with "@enron.com". We connected two employees if they wrote to each other at least once. Then we used as dimensions the day of the week when the communication took place (ending up with seven dimensions from Monday to Sunday). For the set of skills, we considered the 8 most used keywords in the subject field of the emails (again eliminating stopwords and stemming the remaining words and directly evaluating the relation between an employee and the keyword by the number of times she used the word in an email subject).

7.2 The UBIK Algorithm

In this section we discuss the implementation details of our algorithm. We called it UBIK ("you (U) know Because I Know"). UBIK requires the following input: a network $G = (V, E, D, S)$ with the characteristics presented in Section 7.1.2; a range parameter α regulating how much information is lost after each degree of separation; and a set specifying, for each $d \in D$, what is the relevance $r(d)$ of d (defined by the analyst accordingly to the ranking aims).

Aim of UBIK is to update $\forall s \in S$ and $\forall u \in V$ the value w, i.e. how much node u possesses of skill s. The pseudocode of UBIK is Algorithm 1. UBIK cycles for each node, using the following master equation:

$$f(u,s) = \sum_{d \in D} \sum_{v \in N(u,d)} \frac{(f(v,s) \times r(d))^{\frac{1}{\ell \alpha}}}{|N(u)| + |N(v)|}$$

where $N(u,d)$ is a function returning all the neighbors of u that are reachable through dimension d (if a dimension is not specified, it returns the entire neighborhood). Notice that the contribution of each d is different, corrected with the value $r(d)$, i.e. the relevance of dimension d. Also note that, at the first iteration, $f(v,s)$ (i.e. how much node v possesses of skills s) is equal to just the weight $w_{v,s}$, but at the second iteration it will be updated with the master equation.

One important caveat must be discussed about the ℓ parameter. In our model (Section 7.1.2) we said that ℓ represent the degrees of separation of the nodes u and v, exchanging their skill values. Therefore, the exact implementation of our model would require to scan for each u each node of the network, calculate the shortest path between the two, and then update the contribution accordingly to the ℓ value. However, this implementation is inefficient, as it is an equivalent of finding all the shortest paths in the network (that is a cubic problem in terms of the number of nodes, or $|V|^3$ [5]) and then apply our calculation. Instead, Algorithm 1 provides an approximation of the result. The approximation reduces the main loop time complexity as linear in terms of number of edges, usually approximated as $|V| \log |V|$.

We set $\ell = 1$ and we apply the master function to every node and every skill. Then, we increase ℓ by 1 and we apply the master function again, using not the original skill values of nodes, but the ones updated at the first iteration. In this way, all the neighbors of u are passing to u also the skills that they have inherited from their neighbors. We avoid to pass back to u the skills that u itself passed to its neighbors at the previous iteration. At the n-th iteration, the neighbors of u pass to u the skill values obtained by the nodes $n - 1$ degrees away.

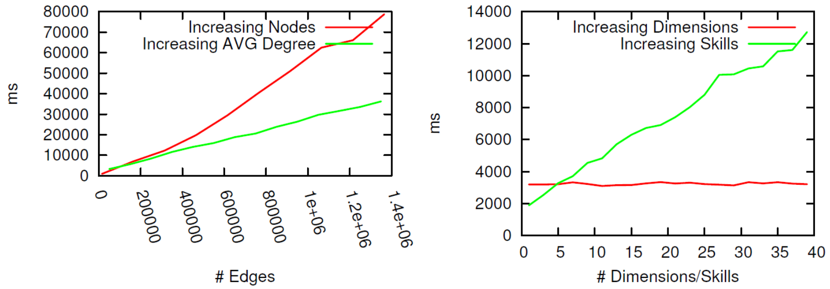
The stop criterion is dependent on the ℓ value. On average, nodes that are beyond three or four degrees of separation cannot influence significantly the skills accessible from one node. Therefore, in Algorithm 1, at step 2 we stop if $\ell \geq \delta$, with $3 \leq \delta \leq 6$, dependent on the application.

When we calculate the new skill values at step 6, we store the result in a temporal variable for each node. Then, we apply the $UPDATE$ function at step 10 to update the value of skill s for node u in the node set V. Each element of the master equation is either fixed (α, D, $r(d)$, $N(u)$ and $N(v)$ are always the same) or it only depends on the previous iteration (ℓ, $u(s)$ and $v(s)$). By forcing this condition, UBIK becomes order-independent: the computed value of each $u(s)$ at a particular iteration is always the same, regardless if u was considered as the first node of the iteration or as the last.

The $NORMALIZE$ function at step 13 scales for each skill the values obtained for each node in the $[0, 1]$ interval. Moreover, it combines all the skill values in a general network ranking. The general value of node u is evaluated by simply extending the $u(s)$ function in the following way: $u(*) = \sum_{s \in S} u(s)$, where $*$ symbolizes the sum of all $s_1, s_2, \ldots, s_n \in S$. This function is run at the end of the main UBIK loop and does not add any complexity to it.

7.2.1 Time Complexity

Algorithm 1 has five nested loops. From the inner to the outer, they cycle over: the set of dimensions (the sum at step 6), the set of skills (step 5), the neighbors of a node (step 4), the nodes of the network (step 3) and until the $\ell < \delta$. Since cycling over the nodes and their neighbors (steps 3-4) is equivalent to cycle twice over the edges, the complexity of those two loops is $\mathcal{O}(|E|)$. Steps 5-6 have complexity of $\mathcal{O}(|S|)$, while the outer loop generally terminates after very few iterations: in real world networks, usually $\delta = \log |V|$. The final estimate for the time complexity is then $\mathcal{O}(\log |V| \times |E| \times |S|)$. We also report that usually for real world networks, the number of skills and dimensions (both in the order of 10^1 or 10^2) is usually much lower than the number of edges (usually ranging from 10^5 to 10^8 and more), making the average case complexity in the order of



(a) First series.

(b) Second series.

Figure 7.2: Running times in milliseconds for different random networks with given number of edges, dimensions or skills.

$\Theta(\log |V| \times |E|)$.

7.3 Experiments

We tested our Java implementation of UBIK[4], on a Dual Core Intel i7 64 bits @ 2.8 GHz, 8 GB of RAM and a kernel Linux 3.2.0-23-generic, using as virtual machine the Java OpenJDK version 1.6.0_24. Our implementation took on average 22 seconds on DBLP and less than 2 seconds on Enron. Our networks are small in scale, thus we created some benchmark networks to show how UBIK scales in terms of number of nodes, average degree, dimensions and skills. The results are depicted in Figures 7.2a and 7.2b.

First, we fixed the number of dimensions and of skills at 5. Then for the “increasing nodes” series we fixed the average degree at 3 and we increased the number of nodes; while for the “increased AVG degree” series we fixed the number of nodes at 50k and we increased the average degree of the nodes. Both techniques increase the number of edges: UBIK is able to scale linearly in this dimension. UBIK is able to analyze a network with 455k nodes and 1.3M total edges over all dimensions in less

[4] Freely available with our test datasets at `http://www.michelecoscia.com/?page_id=480`

than a minute and a half, or with 50k nodes and the same number of edges in less than 40 seconds. The difference in the linear slope between the two is given by the increasing of both nodes and edges. We can conclude that UBIK is scalable and applicable to large scale networks, as it is linear on the number of edges. In Figure 7.2b we fixed the number of nodes at 25,000 and the average degree at 3. Then for the "increasing dimensions" series we increased the number of dimensions from 1 to 40, while for the series "increasing skills" we increased the number of skills from 1 to 40. Our implementation, paying a preprocessing phase, is independent from the number of dimensions, the runtime increases linearly with the number of skills[5].

We now proceed to evaluate the results of UBIK, comparing its results with some of the state-of-the-art node rank approaches (in Section 7.3.1) and presenting some knowledge extraction from real world networks (in Section 7.3.2).

7.3.1 Comparison with other methods

We compare the rankings provided by UBIK with some state-of-the-art algorithms. The algorithms used for comparison are the personalized PageRank [56] and TOPHITS, a tensor eigenvector-based approach to ranking. Personalized PageRank is implemented in the R statistical software, TOPHITS is part of the Tensor Toolbox for MatLab [74], freely available for download[6]. For our comparison, we used the DBLP network.

We used UBIK without giving to any dimension any particular value of $r(d)$ and we took the global ranking of the nodes without selecting any particular skill. In this way, the comparison with PageRank and TOPHITS is fair, because we are evaluating the general rank of our nodes without using anything else than the network structure, that both the Personalized PageRank and TOPHITS can handle. Also, we set $\alpha = 2$ and $\delta = 6$.

The task of confronting different ranking methods is not easy, as it is

[5] To assure repeatability, also the random network generator is provided at the same page of the algorithm and the networks

[6] `http://www.sandia.gov/~tgkolda/TensorToolbox/index-2.5.html`

not explicit why a ranked list is better than a different one on absolute terms. However, there are several properties that we would like to have in the results of a ranking algorithm. We evaluate the results of the algorithm based on quantitative tests on the following properties:

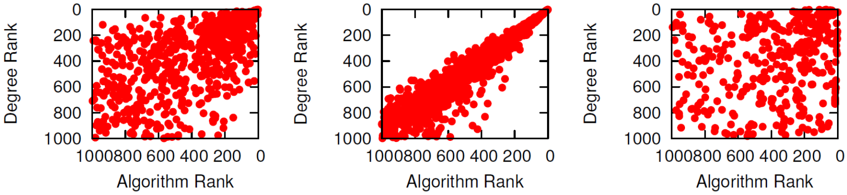
1. Ranking results should not be trivial: if the results are highly correlated with a trivial ranking method, then the algorithm is not telling us something interesting.

2. Ranking results should not be trivially boosted: if there is a simple mechanism to increase one's rank, the flaw of the algorithm makes its results less important.

3. Given a gold standard calculated in an independent way, a ranking algorithm is good if it is quantitatively similar, to some extend, to the gold standard.

Let us start from the first element of the list. One of the most trivial criterion for ranking nodes in a network is to check their degree: the more edges are connected to a node, the more important it is. Of course, this ranking method is not optimal, as it takes only an artificial creation of many edges centered on a node to obtain the maximum rank (as it happens in the World Wide Web). Therefore, the most similar to the degree ranking are the results of the algorithm, the less interesting they are. This is only the first test to be satisfied, but it is necessary to satisfy also the other two. For example, a ranking method that uses the inverse of the degree to rank node will pass this test, as it anti-correlates with the trivial degree ranking method, but it will not satisfy the other two conditions.

Figure 7.3 depicts the q-q plots of UBIK, PageRank and TOPHITS against the degree ranking for the 1,000 nodes with highest degree. Each point $x(i, j)$ of a q-q plot corresponds to some node x. The coordinates of the point (i, j) mean that the node is ranked at the i-th position by the first algorithm (x-axis) and at the j-th position by the second algorithm (y-axis). In Figure 7.3, the y axis is the degree rank, while on the x axis we have UBIK (Figure 7.3a), PageRank (Figure 7.3b) and TOPHITS (Figure 7.3c).



(a) UBIK.

(b) PageRank.

(c) TOPHITS.

Figure 7.3: The q-q plots of various ranking algorithms against the ranking obtained ordering the nodes by degree.

R	Degree	PageRank	UBIK
1	Jiawei Han	Jiawei Han	Philip S. Yu
2	Philip S. Yu	Philip S. Yu	Jiawei Han
3	Christos Faloutsos	Christos Faloutsos	Qiang Yang
4	Qiang Yang	Qiang Yang	Hans-Peter Kriegel
5	Divesh Srivastava	Divesh Srivastava	Gerhard Weikum
6	Zheng Chen	Jian Pei	Divesh Srivastava
7	Jian Pei	Zheng Chen	Zheng Chen
8	Raghu Ramakrishnan	Hector Garcia-Molina	Elke A. Rundensteiner
9	Beng Chin Ooi	Beng Chin Ooi	C. Lee Giles
10	Hector Garcia-Molina	Gerhard Weikum	Christos Faloutsos
11	Haixun Wang	Raghu Ramakrishnan	Wei-Ying Ma
12	Wei-Ying Ma	Haixun Wang	Yong Yu
13	Gerhard Weikum	Wei-Ying Ma	Tao Li
14	Michael J. Carey	Michael Stonebraker	Ming-Syan Chen
15	Jeffrey Xu Yu	Rakesh Agrawal	Jian Pei

Table 7.2: The top 15 researchers according to different ranking criteria.

Algorithm	(1)	(2)	(3)	(4)	(5)
UBIK	0.0204	1%	1.2%	4%	5.5%
PageRank	0.0166	0%	0.8%	2%	5.3%
TOPHITS	0.0857	39%	33.6%	34.8%	37.5%

Table 7.3: The share of high clustering nodes in the top rankings per algorithm.

The interpretation of the picture is clear: especially for the 300 highest ranked nodes, having a high degree implies having a high PageRank, while this consideration does not hold for UBIK and TOPHITS results. We also report the top 15 researchers in Table 7.2 for UBIK and PageRank (TOPHITS omitted due to lack of space, but the interesting confront of the table is with the PageRank algorithm, as TOPHITS does not show the rank-degree correlation). Again, we can easily see the correlation between the Degree and the PageRank column. The rank-degree correlation for PageRank is not our finding, as it has already been studied in literature [49]. In practice, the logic of the degree centrality is “The more collaborators a researcher has, the more important he is”. Both PageRank and UBIK modify this philosohpy in “The more important collaborators a researcher has, the more important he is”. However the “important” in PageRank is still a quantitative measure on the number of collaborations, while UBIK uses more qualitative information. For the first criterion, we conclude that UBIK rankings are less trivial than the ones returned by PageRank.

So far we have shown the main defect of PageRank, i.e. it provides a trivial ranking. What is the main defect of TOPHITS? In literature, it is studied as the Tightly-Knit Community (TKC) effect: being the TOPHITS rank self-enforced through eigenvector calculation, if we highly clustered nodes in the network, it may happen that all members of this group are ranked high by TOPHITS, even though the nodes are not particularly central. This is the second point we want to prove: UBIK rankings are not prone to these easily applicable ranking boost strategies.

Table 7.3 reports the share of high clustering nodes for the top k ranked nodes, accordingly to UBIK, PageRank and TOPHITS. The column (2) reports the percentage of nodes in the top 100 ranked by the algorithm

	App.	Search	Stream	Obj.	Analysis	Sys.	User	Model	Network	Context
P. S. Yu	1	3	4	7	4	7	5	6	2	6
J. Han	5	1	3	5	3	6	4	3	3	4
Q. Yang	11	15	1	13	5	14	8	18	7	10
H-P. Kriegel	3	2	5	1	7	12	6	10	16	2
G. Weikum	7	9	15	2	1	17	2	11	5	5
D. Srivastava	18	5	24	21	11	1	12	2	11	7
Z. Chen	17	17	2	20	6	22	1	14	20	17
Rundensteiner	2	7	11	14	9	15	14	1	8	11
C. Lee Giles	14	16	16	16	10	19	7	23	1	14
C. Faloutsos	21	6	34	30	15	28	15	20	30	1

Table 7.4: The top 10 researchers according to the general UBIK ranking and their rank for 10 different skills.

that have a local clustering $k > .1$, the column (3) reports the same statistic for the top 250 nodes, and so on. The local clustering $k(i)$ of a node i is defined as:

$$k(i) = \frac{2|\{(u, v) \mid u, v \in N(i) \wedge (u, v) \in E\}|}{|N(i)| \times (|N(i)| - 1)}$$

(note that we calculate the monodimensional clustering, without specifying a d for $N(i)$). We can see that both UBIK and PageRank tend not to return high ranks for nodes with a high local clustering value. On the other hand, in the TOPHITS ranks 39 nodes out of the most important 100 have high clustering values. Table 7.3 also reports the average local clustering value for the top ranked 100 nodes in column (1) and again this value is $> 4\times$ higher for TOPHITS. We conclude that both UBIK and PageRank are not affected by the TKC, and that UBIK is the only example that is not dependent both on degree and local clustering.

Let us now address the third important feature that a ranking algorithm should have: the comparison with a quantitative gold standard. In scientific publishing, a useful indicator about the quality and the impact of a researcher is quantified using several different indexes. One of them, the h-index, measures both the amount of publications and citations of an author, and it is logically not related to the co-authorship network. A researcher has an h-index of h if he has at least h publications cited at least h times. We use the h-index as our ground truth.

For this comparison we could use a q-q plot, but we need a more quantitative and objective measure than simply looking at the plot. Therefore, we follow [115] and we use a function computing the distance of the points in a q-q plot from the line $y = x$ that represent identical rankings. The distance of point (i, j) from the line $y = x$ is equal to $\frac{|i-j|}{\sqrt{2}}$. Thus, the distance measure D of two rankings r_1 and r_2 is:

$$D(r_1, r_2) = \frac{1}{|V|} \sum_{\forall v \in V} \frac{|r_1(v) - r_2(v)|}{|V|}$$

where $r_x(v)$ is the rank of v according to the ranking r_x.

We calculate this function for UBIK, PageRank and TOPHITS against h-index ranking. We obtained the h-index values from an updated webpage who is collecting data from Google Scholar[7]. From that list, we removed the authors that have not published a single paper in the set of conferences with which we have built our multidimensional network, because they are not part of the network structure at all. UBIK's ranking is closer to the ground truth, computed independently from the network structure, provided by the h-index ranking, with a score of 22.43. Therefore, not only UBIK is not affected by the biases of PageRank and TOPHITS, but it also yields results closer to an independent ground truth, as they scored 23.50 and 37.54 respectively. We can now take a look to the actual multiskill and multidimensional rankings provided by the algorithm, as the comparison section is over and we can use the features, not handled by PageRank nor TOPHITS.

7.3.2 Rankings

We now report some rankings extracted with UBIK. We already saw in Table 7.2 the top 15 researcher setting no particular dimension weight (i.e. $r(d)$ is equal for all $d \in D$). However, now we want to take advantage of the fact that UBIK is able to return different rankings for each skill. In Table 7.4 we report the list of researchers who can master some skills, and their ranking for the other skills. As we can see, no researcher dominates

[7] http://www.cs.ucla.edu/~palsberg/h-number.html

over all the skills, and the different rankings can enlighten us about different leaders in different sectors. We remind that we took only authors of a very specific set of conferences, thus a possible specialist in one or more reported skills may not be part of the rankings because she never published in one of the selected conferences. Also, the skill name is the substantive of the stemmed form, thus it includes all the possible declination of the term (e.g. “Stream”, “Streaming”, “Streamed”, and so on).

UBIK is able to customize the ranking even further, in an additional degree of freedom. Instead of looking at some skills taken separately, we can populate our set of dimension relevance functions with different importance $r(d)$ values for different dimensions. A proper definition of the dimension importance set results in a very specific ranking analysis. To show this feature, we decide to create two different definition classes for the Enron network. We recall that in the Enron network each dimension is the day in the week when the email was sent. In the first variant, we populate our set of rules with a $10\times$ multiplier for the dimensions of Saturday and Sunday; in the second variant we apply the same $10\times$ multiplier, but this time to each of the weekday, and nothing to the weekend days. The choice of the $10\times$ multiplier is *ad hoc*, to represent a query that focuses on the relations established mainly during weekdays (first case) or during weekends (second case).

Table 7.5 reports the top 10 employees according to both criteria (please note that some employees are part of both top 10 and they are not repeated). As we can see, the two rankings are quite distinct. There are three employees very important in both criteria (Jeff Dasovich, Michael Kass and Chris Germany). We also observe one expected phenomenon: the important employees during the weekdays are also somewhat important during the weekends, while the vice versa is not true. It is expected that important employees receive emails during the entire week, while the communications during the weekend may follow a different logic and promote unexpectedly low ranked employees (maybe because they perform a weekend shift or due to particular emergencies outside office hours).

Employee	Weekday Rank	Weekend Rank
Victor Lamadrid	1	34
Jeff Dasovich	2	4
Lisa Jacobson	3	39
Michael Kass	4	2
Joannie Williamson	5	37
Bob Ambrocik	6	68
Chris Germany	7	8
Kay Chapman	8	27
Tana Jones	9	23
Drew Fossum	10	18
Forrest Lane	238	1
Jennifer Blevins	1760	3
Shubh Shrivastava	857	5
Kay Mann	14	6
Scott Neal	50	7
Vince Kaminski	18	9
Rosalee Fleming	21	10

Table 7.5: The top 10 employees according to the UBIK for the weekday and the weekend variants of the ranking.

The most notorious elements of the top management of Enron are not present in either rankings. Kenneth Lay is ranked 617th in weekdays and 224th in weekends, while Jeffrey Skilling is ranked 725th in weekdays and 458th in weekends. Joannie Williamson, who worked as a secretary for both of them[8], is instead present in Table 7.5, and highly ranked. This is an expected result of UBIK, able to unveil who is a knowledge gateway in an organization.

The same multidimensional ranking can be done for the DBLP network. In this case, we are able to spot the collaboration hubs in several different conferences. We applied the $10\times$ multiplier to a collection of conferences. The results for some of our conferences are provided in Table 7.6, where for each conference we record the top ranked author and then we report also his ranking for the other conferences. We can see that UBIK is able to identify specialists who are highly ranked only in one conference. On the other hand, as expected, the top authors in the general

[8] `http://money.cnn.com/2006/04/03/news/newsmakers/enron_defense/index.htm`

Author	ACL	CIKM	ICDE	KDD	GRAPH	VLDB	WWW
D. Marcu	1	3893	3511	2606	2309	1608	3327
Boughanem	2768	1	3892	2966	2643	1946	3658
T. Ichikawa	3070	4549	1	3266	2982	2260	3994
Nakhaeizadeh	2606	4036	3709	1	2484	1783	3507
D. Salesin	2101	3538	3149	2304	1	1300	2980
P. Dubey	3093	4570	4198	3274	2988	1	3999
J. Nieh	2976	4453	4134	3185	2865	2158	1
P. S. Yu	574	32	27	10	860	10	684
J. Han	689	106	60	12	999	29	708
Q. Yang	933	634	967	384	1258	232	930
H-P. Kriegel	1024	517	66	330	1304	146	1476
G. Weikum	1100	472	91	888	1348	51	1355

Table 7.6: The top authors for some conferences (taken singularly) comparative rankings.

ranking score average high rank in all conference, but they are rarely in the top 10 of a specific conference, as their impact is more broad and it spreads over many different venues (the bottom rows of Table 7.6 records the ranking for each single conference for the top 5 general authors in the network taken without any dimension multiplier).

7.4 Conclusion

In this Section we addressed a problem of human resources: the ranking of employees according to their skills. We did so following an intuition about the intellectual value of a person: her evaluation should not be based only on her set of skills, but also on her friends' set of skills. We created an algorithm, UBIK, to tackle this problem: UBIK is able to rank nodes in a multidimensional network, with weighted labels referring to their set of skills. We applied UBIK to two real world networks, confronting its output with popular ranking algorithms, showing that our results are less trivial and more significant. Our contributions are: the creation of a feasible tool to address a problem in organizations in the real world; the address of the multiple ranking over multidimensional networks problem, not tackled in literature; and increased ranking performances over the current most used methodologies.

Our paper opens the way to a number of future works. First, following [91], we can extend UBIK to rank also the relations. UBIK can be applied to several different datasets with different semantics and properties, from

Linkedin to evaluate people's expertise; to networks of international organizations, to detect the most important organizations given a set of topics. Lastly, we can create a more efficient implementation, confront with other algorithms and evaluation measures.

Chapter 8

The Three Dimensions of Social Influence

This chapter is mainly based on [102].

One of the classic problems in social network analysis is the comprehension of percolation and diffusion effects in networks. Modeling diffusion processes on complex networks enables us to tackle problems like preventing epidemic outbreaks [31] or favoring the adoption of new technologies or behaviors by designing an effective word-of-mouth communication strategy, like condom usage in Africa [26]. In this Section, we are focused particularly on the social influence aspect of the diffusion problem in networks.

In the setting of favoring social influence, most of the attention of researchers has been put on how to maximize the number of nodes subject to the spreading process. Usually, this is done by choosing appropriate seeds in critical parts of the network, such that their likelihood of influence diffusion is maximum, in order to achieve as large cascades as possible. While influencing as many nodes as possible is obviously part of this problem definition, we argue that the size of cascades is not the unique dimension of the social influence problem. Three other dimensions are relevant: the *width*, the *depth* and the *strength* of the social influence of any given node in a network. The width of a node is the ability of influencing its immediate neighbors; the depth is its ability to cause long cascades, i.e. of influencing nodes that are socially distant; the strength is the ability to influence with intensity, so to bring the nodes in the cascade to achieve a desired result at the highest possible degree.

Indeed, many real-world scenarios focus on a more specific diffusion pattern that requires a more fine grained understanding of the influence mechanics at play, along the three mentioned dimensions. Some examples are:

1. An analyst needs information from the personal acquaintances of a subject. The important aspect is that many among the subject's direct connections respond, ignoring people two steps away or more.

2. A person is interested in finding another person with a given object. The important aspect is that some people are able to pass her message through a long chain that ends with the desired target (what it is today known as "crowdsourcing" the search).

3. A music artist wants to influence people in a social network to listen to her new album. The important aspect is that some people are influenced above the threshold that will make them actually buy the album.

These are different scenarios requiring significantly different diffusion patterns. Scenario 1 requires a broad diffusion in the first degree of separation. Scenario 2 requires a targeted diffusion similar to a Depth First Search algorithm on graphs. Scenario 3 requires a high-intensity diffusion. Different scenarios may require any combination of the three.

Here, we make use of three measures to capture the defining characteristics of these three scenarios: the Width, Depth and Strength of social influence. The Width measures the ratio of the neighbors of a node that are influenced by the node's actions. The Depth measures how many degrees of separation there are between a node and the other nodes that are influenced by its actions. The Strength measures the intensity of the social influence that a node has over other nodes.

We study what the relations are between these three measures to understand if we are truly capturing three orthogonal dimensions of social influence. We also study the relations between the Width, Depth and Strength measures and different node properties. In this way, we may be able to predict the typical diffusion patterns of different events given the characteristics of the nodes that lead their diffusion.

To empirically validate our concepts, we constructed a social network from the popular music platform Last.Fm[1], along with the listening data of artists for each node of the network, i.e. how many times and when each user listens to a song performed by a given artist. We create a procedure to detect who are the "leaders" for each artist, i.e. the users who start listening to an artist before any of their neighbors. We calculate for each leader their values of Width, Depth and Strength, along with their network statistics such as the degree and the betweenness centrality, looking for associations between them. We then create a case study to understand what are the different dynamics in the spread of artists belonging to different music genres, by using the tags users attach to the songs in the social platform.

To sum up, the contributions are:

- A proof that social diffusion indeed follows at least these three dimensions, which are uncorrelated or anti-correlated;
- A fast and efficient way to identify leaders in social network and to calculate their Width, Depth and Strength values;
- The discovery of some significant associations between the three dimensions of social influence and some traditional network measures;
- The ability to predict the patterns of diffusion of particular influence events by looking at the characteristics of the leaders spreading them.

[1] `http://www.last.fm/`

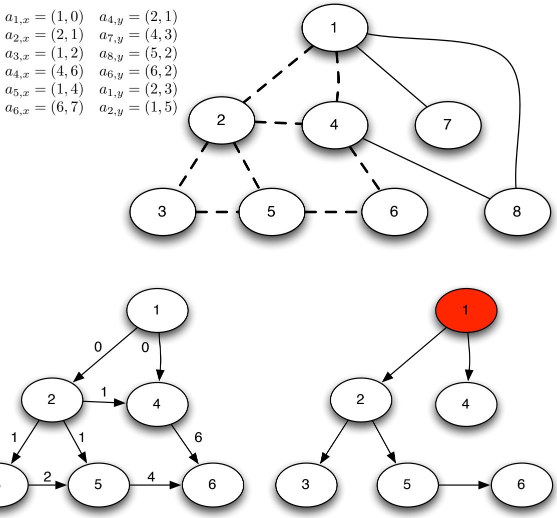


Figure 8.1: Toy Example. On the *top* the social graph $\mathcal{G}$ and action set $\mathcal{A}$, where $x, y \in \Psi$ are the objects of the actions; in the *bottom-left corner* the induced subgraph for the action x; on the *bottom-right corner* the diffusion tree for x. In red we highlighted the leader (root) for the given tree.

8.1 Leader Detection

Each diffusion process has its starting points. Any idea, disease or trend, in order to successfully propagate, needs to be adopted by particular kinds of actors. Such actors are not like every other actor: they are the first to perform an action, nobody directly influences them and, due their network position and influence, they are able to set in motion the diffusion process. We call such actors *leaders*. Given a graph, several interesting problems arise regarding how information spreads over its topology: can we identify the *leaders*? Can we characterize them? What kind of knowledge should we expect to extract from their analysis?

Our approach aims to detect *leaders* through the analysis of two strictly correlated entities: the topology of the social graph and the set of actions performed by the actors (nodes). When discussing the roles of those entities, we refer respectively to the following definitions:

Definition 8.1 (Social Graph) *A social graph $\mathcal{G}$ is composed by a set of actors (nodes) V connected by their social relationships (edges) E. Each edge $e \in E$ is defined as a couple (u, v) with $u, v \in V$ and, where not otherwise specified, has to be considered undirected. With $\Gamma(u)$ we identify the neighbor set of a node u.*

Definition 8.2 (Action) *An action $a_{u,\psi} = (w, t)$ defines the adoption by an actor $u \in V$, at a certain time t, of a specific object ψ with a weight $w \in \mathcal{R}$. The set of all the actions of nodes belonging to a social graph $\mathcal{G}$ will be identified by $\mathcal{A}$, while the object set will be called Ψ.*

We identify with $\mathcal{G}_\psi = (V_\psi, E_\psi)$, where $V_\psi \subset V$ and $E_\psi \subset E$, the induced subgraph on $\mathcal{G}$ representing respectively the set of all the actors that have performed an action on ψ, and the edges connecting them. We depict an example of the social graph and the set of actions in Figure 8.1 *(left)*, where the induced subgraph for the object x is highlighted with a dashed line.

Given the nature of a diffusion process, we would expect that each *leader* will influence its neighbors starting a cascade event that follows some rigid temporal constraints. Our constraint is that a node u can influence a neighbor v iff given $t_{u,\psi} \in a_{u,\psi}$ and $t_{v,\psi} \in a_{v,\psi}$ is verified that $t_{v,\psi} > t_{u,\psi}$ and $t_{v,\psi} - t_{u,\psi} \leq \delta$. Here, δ is a temporal resolution parameter that limits the influence spread: if $t_{v,\psi} - t_{u,\psi} > \delta$, we say that v executed action $a_{v,\psi}$ independently from u, as u's influence interval is over.

We transform each undirected subgraph $\mathcal{G}_\psi$ in a directed one imposing that the source node of an edge must have performed its action before the target node. After that, each edge (u, v) will be labeled with $min(t_{u,\psi}, t_{v,\psi})$ in order to identify the starting time of the influence between the nodes. The directed version of $\mathcal{G}_\psi$ represent all the possible diffusion paths that connect leaders with their "tribes" (Figure 8.1 *(center)* an example for the object $x \in \Psi$).

From now on, for a given object ψ, we will refer to the corresponding leader set as $\mathcal{L}_\psi$: when no action is specified the set $\mathcal{L}$ will be used to describe the union of all the $\mathcal{L}_\psi$ for the graph $\mathcal{G}$. In order to be defined a *leader* an actor should not have any incoming edges in $\mathcal{G}_\psi$. This is because a *leader* cannot be influenced by other nodes (they are, in their surroundings, innovators), and is a direct consequence to the adoption of a directed graph to express diffusion patterns. Note that, given this definition, for each directed connected component $\mathcal{C}_\psi \subset \mathcal{G}_\psi$ multiple nodes can belong to $\mathcal{L}_\psi$.

One caveat of our problem definition comes from the fact that a leader may be influenced by exogenous events. Therefore what we are studying is only the influence endogenous to the social network. Also any other node influenced by the leader may have had an exogenous stimulus, but for sure the endogenous stimulus coming from the leader should have played some kind of role.

In order to study the real path of diffusion given an action a and a leader l we build a minimum diffusion tree:

Definition 8.3 (Leader's Minimum Diffusion Tree) *Given an action a_ψ, a directed connected component $\mathcal{C}_\psi$ and a leader $l \in \mathcal{L}_\psi$, the minimum diffusion tree $T_{l,\psi} \subset \mathcal{C}_\psi$ is the* Minimum spanning tree *(MST) having its root in l and built minimizing the temporal label assigned at the edges.*

An example of minimum diffusion tree for the node 1 and object x is shown in Figure 8.1 *(right)*. For each object the diffusion process on a given network is independent and that, given temporal dependencies on its adoption (expressed through actions $a_{*,\psi} \in \mathcal{A}$), it is possible to identify the origin points of the diffusion. The identified *leaders* will show different topological characteristic and will influence their surroundings in different ways: our aim is to classify diffusion *leaders* characterizing some of their common traits.

To sum up, we call our leader extraction procedure EXTRACTLEADERS and report the pseudocode in Algorithm 2. For all objects $\psi \in \Psi$, we extract the directed induced subgraph $\mathcal{G}_\psi$ by filtering all nodes that performed an action $a_{*,\psi} \in \mathcal{A}$ and all the edges between them (performed by INDUCEDSUBGRAPH where δ is the temporal constraint discussed before). Then, for each connected component $\mathcal{C}_\psi \in \mathcal{G}_\psi$ we choose as our leaders all the nodes without incoming edges (performed by INDEGREE). We add them to the leader set $\mathcal{L}_\psi$ and we store in $\mathcal{T}_\psi$ their Minimum Diffusion

Algorithm 2 The pseudo-code of ExtractLeaders.

```
Require: $\mathcal{G} = (V, E); \mathcal{A}; \Psi, \delta;$
Ensure: $\mathcal{L}, \mathcal{T}$
$\mathcal{T} \leftarrow \{\}$
$\mathcal{L} \leftarrow \{\}$
for all $\psi \in \Psi$ do
    $\mathcal{G}_\psi \leftarrow$ InducedSubgraph$(\mathcal{G}, \psi, \delta)$
    $\mathcal{T}_\psi \leftarrow \{\}$
    $\mathcal{L}_\psi \leftarrow \{\}$
    for all $\mathcal{C}_\psi \in \mathcal{G}_\psi$ do
        for all $l \in \mathcal{C}_\psi$ do
            if InDegree($\mathcal{C}_\psi, l$)==0 then
                $\mathcal{L}_\psi \leftarrow \mathcal{L}_\psi \cup l$
                $\mathcal{T}_{l,\psi} \leftarrow MST(C_\psi, l)$
                $\mathcal{T}_\psi \leftarrow \mathcal{T}_\psi \cup \mathcal{T}_{l,\psi}$
            end if
        end for
    end for
    $\mathcal{L} \leftarrow \mathcal{L} \cup \mathcal{L}_\psi$
    $\mathcal{T} \leftarrow \mathcal{T} \cup \mathcal{T}_\psi$
end for
return $\mathcal{L}, \mathcal{T}$
```

Trees (calculating the minimum spanning tree MST with root in l using only nodes in C_ψ). At the end, we return the union of $\mathcal{L}_\psi$ and $\mathcal{T}_\psi$.

We now evaluate the complexity of Algorithm 2. First, we cycle over all the actions ($\mathcal{O}(|\Psi|)$). Then, we cycle over all the connected components of $\mathcal{G}_\psi$ and, for each one, we cycle over the nodes belonging to them: together the two cycles reach the complexity of $\mathcal{O}(|V_\psi|)$. Within the inner loop a minimum spanning tree ($\mathcal{O}(\log |V|)$, with Kruskal's algorithm) is computed for every leader. As a consequence, the final complexity of ExtractLeaders is $\mathcal{O}(|\Psi| \times |V| \log |V|)$. For large networks, it is fair to assume that $|\Psi| << |V|$, so the complexity would be $\Theta(|V| \log |V|)$. Moreover, since each action is independent from the others, with $|\Psi|$ processors the exact complexity would be $\mathcal{O}(|V| \log |V|)$.

8.2 Measures

As stated above, we are interested in capturing the three dimensions of social influence. We need three network measures able to capture these three dimensions. We call these dimensions Width, Depth and Strength. Each dimension matches with an intuitive concept:

- The Width is the ratio of neighbors influenced by an action performed by a node;
- The Depth is how many degrees of separation are in between a node and the most distant of the nodes influenced by it;
- The Strength is how strongly the influenced nodes are engaged.

In Section 8.2.1 we provide the formal definitions of the measures connected to these intuitions. In Section 8.2.2 we then use our toy example depicted in Figure 8.1 to provide an example for the calculation of these measures.

8.2.1 Definitions

Given a leader, with the Width measure we want to capture the direct impact of her actions on her closest friends. The Width is the degree of importance that a leader has over all its neighbors.

Definition 8.4 (Width) *Let G be a social graph, $\psi \in \Psi$ an object and $l \in \mathcal{L}_\psi \subset V$ a leader: the function $width : \mathcal{L}_\psi \to [0,1]$ is defined as:*

$$width(l,\psi) = \frac{|\{u|u \in \Gamma(l) \wedge \exists a_{u,\psi} \in \mathcal{A}\}|}{|\Gamma(l)|} \tag{8.1}$$

The value returned is the ratio of all the neighbors that, after the action of the leader, have adopted the same object.

The second measure, the Depth, is an complementary point of view over a leader's influence. Instead of looking at the closest friends, we want to know what is the length of the longest induced diffusion chain. The Depth is an evaluation of how much a leader can influence other influential leaders, which can influence other leaders and so on.

Definition 8.5 (Depth) *Let $T_{l,\psi}$ be a minimum diffusion tree for a leader $l \in \mathcal{L}_\psi$ and a given object $\psi \in \Psi$: the function $depth : T_{l,\psi} \rightarrow \mathbb{N}$ computes the length of the maximal path from l to a node $u \in T_{l,\psi}$. Likewise, the function $depth_{avg} : T_{l,\psi} \rightarrow \mathbb{R}$ computes the average length of paths from l to any leaf of the tree.*

The last proposed measure, the Strength, tries to capture quantitatively the total weight of the adoption of an object after the leader's action. A leader is influential if the nodes influenced by her are very engaged in adopting what she adopted. Given that the direct influence diminishes as new adopters become more distant, in the network sense, from the original innovator. Therefore, we have decided to introduce a distance damping factor.

Definition 8.6 (Strength) *Let $T_{l,\psi}$ be a minimum diffusion tree for a leader $l \in \mathcal{L}_\psi$ and an object $\psi \in \Psi$; $0 < \beta < 1$ a damping factor: the function $strength : T_{l,\psi} \times (0,1) \rightarrow \mathbb{R}$ is defined as:*

$$strength(T_{l,\psi}, \beta) = \sum_{i \in [0, depth(l)]} \beta^i L(T_{l,\psi}, i) \tag{8.2}$$

where $L : T_{l,\psi} \times \mathbb{N} \rightarrow \mathbb{R}$ is defined as:

$$L(T_{l,\psi}, i) = \sum_{\{u | u \in T_{l,\psi} \wedge distance(l,u) = i\}} \frac{w_{u,\psi}}{w_u} \tag{8.3}$$

and represents the sum, over all the nodes u at distance i from l, of the ratio between the weight of action ψ and the total weight of all the actions taken.

8.2.2 Examples

We now look at the example in Figure 8.1 and we calculate the Width, Depth and Strength values for the red node leader and the action x.

For the Width measure, from Figure 8.1*(left)* we see that $\Gamma(1) = \{2, 4, 7, 8\}$, i.e. 4 nodes. Given that $\Gamma_x(1) = \{u | u \in \Gamma(1) \wedge \exists a_{u,x}\} = \{2, 4\}$, we have $width(1, x) = \frac{|\Gamma_x(1)|}{|\Gamma(1)|} = 0.5$.

We now calculate the Depth measure. The leaves in Figure 8.1*(rigth)* are nodes 3, 4 and 6. Node 4 is a direct neighbor of 1, while node 3 is two edges away (through node 2). The maximal chain is $1 \rightarrow 2 \rightarrow 5 \rightarrow 6$, therefore we have: $depth(T_{1,x}) = 3$. We can also calculate $depth_{avg}(T_{1,x})$, that is the average path length in the tree from node 1 to all the leaves: $\frac{1+2+3}{3} = 2$.

Finally, for the Strength measure, we need to use the number of times each node performed action x. We also have to set our damping faction β, that we keep at 0.5. At the first degree we have nodes 2 and 4, that performed action x 2 and 4 times respectively; they also performed action y 1 and 2 times respectively: their contribution is then $\beta^0 \times (\frac{2}{2+1} + \frac{4}{4+2})$. Nodes 2 and 5 are at the second degree of separation as they never performed action y, therefore they add: $\beta^1 \times (1+1)$. Finally, at the third degree of separation, node 6 adds $\beta^2 \times \frac{6}{6+6}$. Wrapping up, $strength(T_{1,x}, 0.5) = 2.458\overline{3}$.

8.3 Experiments

In this section we present our data extracted from the well known music social media platform Last.Fm. We use the data to characterize the Width, Depth and Strength measures, by searching for associations with network topology measures. Finally, we analyze how different users spread their influence on different musical genres in the network.

8.3.1 Data

Last.Fm is one of the most popular online social network platforms, where people can share their own music tastes and discover new artists and genres basing on what they like. Once a user subscribes to an account, she can either start listening Last.Fm personalized Radio or send data about her own offline listenings (using an app called *scrobbler*, that sends metadata from the media player to the web app). For every single song, an user can express her preferences (e.g. *like* or *dislike*, that helps the recommendation system of the radio in proposing new music they may like) and add tags (e.g. genre of the song). Lastly, an user can add friends (undirected connections, the friendship request must be confirmed) and search her neighbors w.r.t. musical tastes. An user can see, in her homepage, her

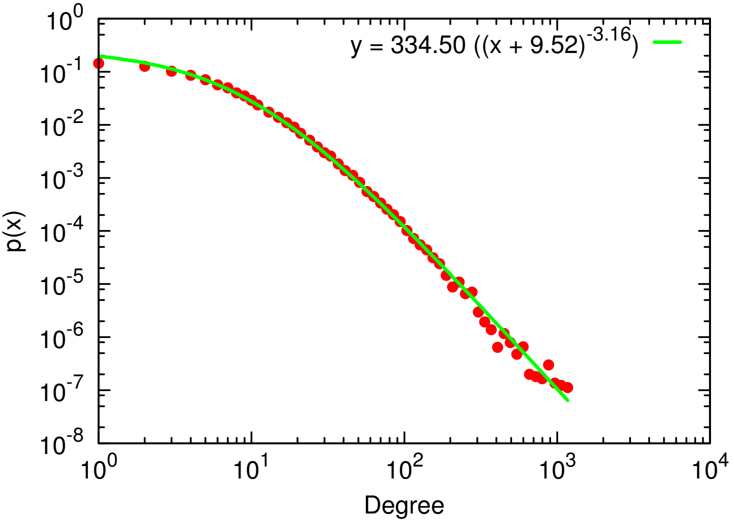


Figure 8.2: Log-binned distribution of the nodes' degree, with power law fitting.

friends' activities.

Using Last.Fm APIs[2], we downloaded a sample of the UK user graph, starting from a set of nodes and implementing a breadth-first approach. We decided to explore the graph up to the fifth degree of separation from our seeds. For each user, we retrieved: (a) her connections, and (b) for each week in the time window from Jan-10 to Dec-11, the number of single listenings of a given artist (e.g. in the week between April 11,2010 and April 18,2010 the user 1234 has listened 66 songs from the artist Metallica).

For each artist we downloaded the list of tags with an associated counter, representing the number of users that assigned that tag to that artist (e.g. Metallica has 4 tags: "metal" with counter 50670, "hard rock" with counter 23405, "punk" with counter 10500 and "adrenaline" with counter 670). This information has been cleaned as follows: we split all tags, associating the counter to each single word (in the last example: (metal, 50670), (punk, 10500), (hard, 23405), (rock, 23405), (adrenaline,

[2] `http://www.last.fm/api/`

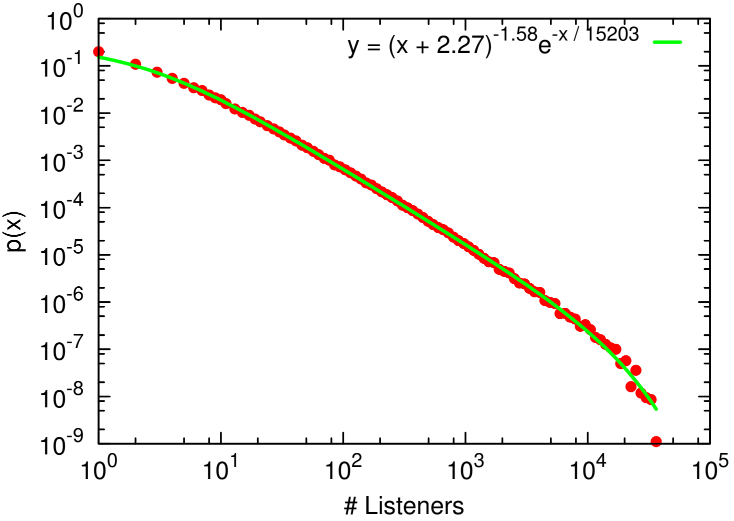


Figure 8.3: Log-binned distribution of number of listeners per artist, with truncated power law fitting.

670)), then we filtered the words referring to a musical genre ((metal, 50670), (punk, 10500), (rock, 23405)). Finally, we assigned a musical genre to an artist iff the survived tag with the greater counter had the relative rate ≥ 0.5 (in the example: $r_{metal}(Metallica) = \frac{50670}{50670+10500+23405} \simeq 0.6$, so Metallica are definitely metal).

After the crawl and cleaning stages, we built our social graph $\mathcal{G}$. In $\mathcal{G}$ every node is a user and each edge is generated by looking at the user's friends in the social media platform. The total amount of nodes is $75,969$, with $389,639$ edges connecting them. In Figure 8.2 we depicted the log-binned degree distribution of $\mathcal{G}$, along with the best fit: as expected, the social networks present a very broad degree distribution across three orders of magnitude. We are then in presence of a classical social network [90].

Each action in the data is one user listening to an artist w times in week t. In Figure 8.3 we depicted the log-binned distribution of the number of listeners per artist, along with the best fit. Again, the distribution has a

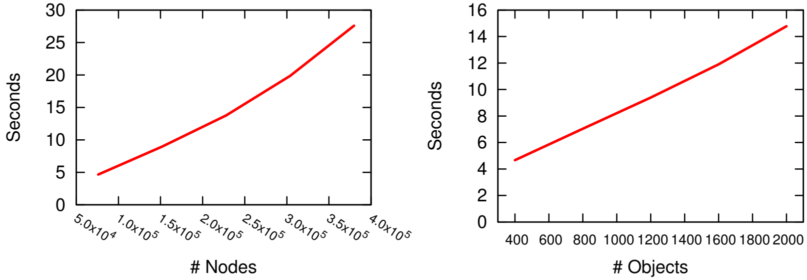


Figure 8.4: The runtimes of the ExtractLeaders procedure with growing number of nodes (*left*) and objects (*right*).

	Width	Strength	Degree	Clustering	Neigh Deg	Bet Centr	Clo Centr
AVG Depth	-0.032126	**-0.238019**	-0.080574	0.052778	-0.088847	-0.026344	**-0.139203**
Width	-	0.019259	**-0.316566**	**0.133336**	0.056522	-0.079626	**-0.599064**
Strength	-	-	0.026328	-0.020074	0.031226	0.003966	0.049765
Degree	-	-	-	**-0.163370**	-0.027931	**0.771641**	**0.564162**
Clustering	-	-	-	-	-0.050080	-0.063129	**-0.326146**
Neigh Deg	-	-	-	-	-	-0.005987	**0.399447**
Bet Centr	-	-	-	-	-	-	**0.226947**

Table 8.1: Pearson correlation coefficient ρ between Width, Depth, Strength and other network statistics for our leaders.

fat tail (with an exponential cutoff), therefore the vast majority of artists do not have enough listeners. We saw that each artist belongs to a music genre (coded in its tag) and we want to use this information in Section 8.3.3. For this reason, we decided to focus on 9 main music genres, namely: dance, electronic, folk, jazz, metal, pop, punk, rap and rock. We want to focus only on known artists, so we select only artists belonging to one of these genres and with at least 100 listeners.

Since we are interested in leaders, we need to focus only on new artists that were previously not existent. If an artist was in activity before our observation time window, there is no way to know if a user has listened to it before, therefore nullifying our leader detection strategy. For this reason, we focus only on artist whose first listening is recorded six months after the beginning of our observation period. With these two combined filters, each genre is represented from 9 artists (jazz) to 111 artists (electronic). With this filter, the cardinality of our action set $\mathcal{A}$ is equal to $168,216$ actions, while the object set Ψ contains a total of 402 artists.

On this dataset settings, the time taken to extract leaders and tribes was 4.670 seconds, with 10.286 additional seconds to calculate Width, Depth and Strength (total runtime 14.956 seconds), on a Dual Core Intel i7 64 bits @ 2.8 GHz, 8 GB of RAM and a kernel Linux 3.2.0-23-generic, with our Python implementation[3]. In our experimental settings, we set our damping factor $\beta = 0.5$ for the calculation of the Strength measure. We also set $\delta = 3$, meaning that if a user listened to a particular artist three weeks or more after its neighbor then we do not consider her to have been influenced by that neighbor. In Figure 8.4 we give a proof of the scalability of our approach: by adding nodes (left) and by adding objects (right). The runtimes of the ExtractLeaders procedure increase linearly.

8.3.2 Characterization of the Measures

For each leader, besides Width, Depth and Strength, we calculated also the Degree (number of edges connected to the node), the Clustering coefficient (ratio of triangles over the possible triads centered on the node), the Neighbor Degree (average degree of the neighbors of the node), the Betweenness (share of the shortest paths that pass through the node) and Closeness Centrality (inverse average distance between the node and all the other nodes of the network).

In Table 8.1 we report the Pearson correlation coefficient ρ between the computed network measures. We highlighted in bold the correlations whose p-value was significant or whose absolute value was strong enough to draw some conclusions. For the significance of p-values, the traditional choice is to set the threshold at $p < 0.01$. However, we decided to be more restrictive, setting our threshold at $p < 0.0005$. We also consider a correlation value ρ significant if $|\rho| > 0.1$. We now highlight the most interesting associations from Table 8.1.

The Depth measure is associated with low Closeness Centrality. This means that a deep influence spread is associated to nodes at the margin of the network. It is expected that nodes with high Closeness Centrality have also low Depth: being central, they cannot generate long chains of influence. The eccentricity of all the nodes of the network ranges from 6

[3] To assure experiment repeatability, we made our cleaned dataset and our code available at the page `http://goo.gl/h53hS`

to 10, meaning that some leaders cannot have a Depth larger than 5 and some leaders can have a Depth equal to 9. To make a fair comparison, we recalculate the Depth value capping it at 5, meaning that any Depth value larger than 5 is manually reduced to 5. Then, we recalculate the correlation ρ between the Depth capped to 5 and the Closeness Centrality obtaining as result $\rho = -0.1366$, with $p < 0.0005$. We can conclude that central nodes are not associated with deep spread of their influence in a social network.

For the Width measure, the anti-correlation with the Degree is not meaningful, as the Degree is in the denominator of Definition 8.4. On the other hand, we observe a positive association with Clustering, meaning that nodes could spread wider their influence in a tightly connected community; and a negative association with Closeness Centrality, meaning that central nodes could not spread a wide influence. Both associations could be explained with the negative correlation with Degree. Therefore, for both measures we run a partial correlation, controlling for the Degree. Given three variables X, Y and Z, the partial correlation is the correlation between the residuals R_X and R_Y resulting from the linear regression of X with Z and of Y with Z, respectively:

$$\rho_{XY \cdot Z} = \frac{\rho_{XY} - \rho_{XZ}\rho_{ZY}}{\sqrt{1-\rho_{XZ}^2}\sqrt{1-\rho_{ZY}^2}}.$$

In our case, X is the Width, Z is the Degree and Y is either the Clustering or the Closeness Centrality value. In practice, we calculate the correlation between Width and Clustering (or Closeness Centrality) by keeping the Degree constant. Results are in Table 8.2: even if significant according to the p-value, the relationship between Width and Clustering is very weak and deserves further investigation. On the other hand, it is confirmed that central nodes are also associated with low Width, regardless their degree.

From Table 8.1, it appears that the Strength measure is not strongly correlated with traditional network statistics. As a consequence, hubs that are associated with low Depth and low Width, do not have necessarily high Strength, making their usefulness in spreading influence in a net-

	Clustering	Clo Centr
Partial ρ	0.087216	-0.536861
p-value	1.57×10^{-14}	0

Table 8.2: Partial correlation and p-value of Clustering and Closeness Centrality with Width, controlling for Degree values.

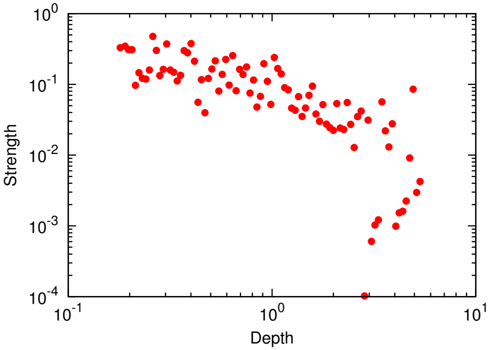


Figure 8.5: Log-binned distribution of the relationship between Depth and Strength.

work questionable. On the other hand, Strength appears to be negatively associated with Depth, as shown in Figure 8.5, suggesting a trade-off between how deeply a node can influence a network and how strong this influence is on the involved nodes.

The anti-correlation between the Strength and the Depth may be due to the damping factor β: from Definition 8.6 we see that β decreases nodes' contributions at each degree of separation (i.e. at increasing Depths). As a consequence, nodes that are farther from the leader contribute less to its Strength, i.e. the highest the Depth the smallest are the contributions to the Strength. We recalculated the Strength values by setting $\beta = 1$, therefore ignoring any damping factor and nullifying this effect. We obtained as result $\rho = -0.4168$ and a significant p-value, therefore concluding that the damping factor β is not causing the anti-correlation between Depth and Strength.

To sum up, we summarize the associations as follows:

- Central nodes are not necessarily important for spreading influence in a social network (low Width and Depth), a result already studied in [12, 21];

- Longer influence chains (higher Depths) are associated with a lower degree of engagement (lower Strengths), a phenomenon possibly related to the role played by "weak ties" [54];
- Influencing a node's neighbors is probably easier if the node is in a tightly connected community, but more evidences have to be brought to reject the role played by the node's degree.

8.3.3 Case Study

In this Section we present a case study based on Last.Fm data. Our aim is to use the Leader extraction technique defined in Section 8.1 and the proposed Width, Depth and Strength measures defined in Section 8.2.1 to characterize the spreading of musical genres among the users of the service. We recall that, as described in Section 8.3.1, the object set Ψ is composed by 402 artists, each one having a tag corresponding to her main music genre.

We organize this case study in three parts: first, in Section 8.3.3 we describe each tag according to the corresponding leaders, by looking at their Width, Depth and Strength; second, in Section 8.3.3 we look at what kind of topological features the leaders corresponding to each tag have; finally in Section 8.3.3 we look for graph patterns that can give us a complementary point of view over the influence patterns that we observe in our network.

Tag Clustering

For each couple leader l and object ψ, we calculate Depth, Width and Strength values; we compute the size of the Leader's Minimum Diffusion Tree ($|T_{l,\psi}|$); and we group together the objects with the same tag. Figure 8.6 shows the distribution of the size of the minimum diffusion trees. As we can see, we obtain approximately the same slope per each tag, with the exception of the "Pop" tag, for which there are many large diffusion trees. Most of the leaders influence only one other user in the social network.

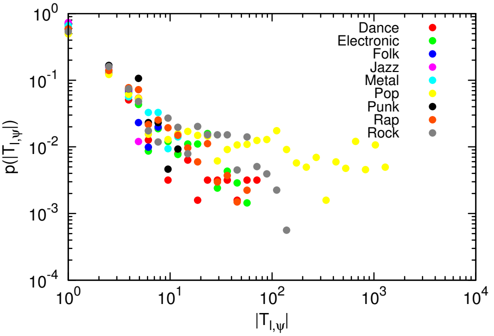


Figure 8.6: Log-binned distribution of the size of the Leader's Minimum Diffusion Tree ($T_{l,\psi}$) per tag.

Cluster	size	dance	electronic	folk	jazz	metal	pop	punk	rap	rock
0	1822	1.2529	1.1377	**1.5488**	1.3753	1.5085	0.7654	1.3114	1.1301	1.1063
1	136	1.2822	1.555	1.289	**2.353**	0.7859	0.7303	0.6458	1.3503	0.7065
2	664	0.5997	0.8702	0.9812	0.4884	0.9517	0.9701	**1.5016**	1.2075	1.194
3	482	1.2674	1.1611	1.0951	1.1276	0.9187	0.8047	**2.4815**	1.248	0.8905
4	973	1.1418	1.2031	1.1592	**1.4106**	0.8034	0.9197	0.6671	0.9724	0.9782
5	512	**1.2925**	0.9667	0.9570	1.0918	1.1062	0.9775	0.3356	1.0661	1.0199
6	682	0.8903	0.797	0.618	0.6446	**1.1303**	1.0892	1.0733	1.0822	1.0186
7	124	0.7579	**1.4554**	0.3507	0.6402	0	1.0958	0	1.0287	0.6279
8	524	0.9308	1.0106	1.1255	0.9131	**1.1564**	1.0707	0.4386	0.9593	0.8743
9	937	0.4027	0.4603	0.1935	0.2355	0.4589	**1.5641**	0.1358	0.3712	1.0616
10	232	0.7227	0.5777	0.2736	0.9989	0.3892	**1.4467**	0.3838	0.463	1.0031
11	612	0.7432	0.9416	0.7186	0.4036	0.7077	**1.2753**	0.0775	0.6859	0.8388

Table 8.3: The RCA scores of the presence of each tag in each cluster.

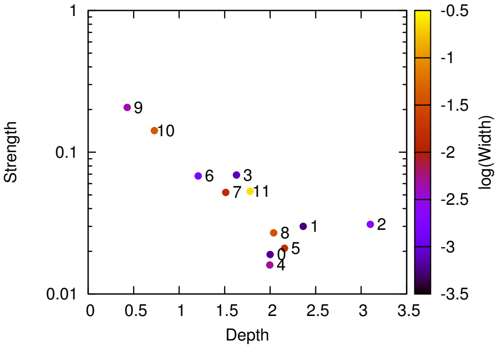


Figure 8.7: The centroids of our clusters.

To characterize the typical values of Width, Depth and Strength for each tag we cannot use the average or the median. This is because Strength and Width values are skewed, and because it is the combination of the three measures that really characterizes the leaders. We decided to cluster each leader using as features its Width, Depth and Strength values. We used the Self-Organizing Map (SOM) method [73] instead of a more classic approach (e.g. k-means) for the following reasons: (i) SOM does not require to set the number of clusters k; (ii) k-means outperforms SOM only if the number of resulting clusters is very small (less than 7) [76], but our study of the best k to be used in k-means with the Sum of Squared Errors (SSE) methodology resulted in a optimal number of clusters falling in a range between 9 and 13 (depending on a seed chosen for initial position of the centroids), so we expect a larger number of clusters; and (iii) SOM performs better if the data points are contained in a warped space [69], which is our case.

The centroids of the SOM are depicted in Figure 8.7: Depth on the x-axis, Strength on the y-axis and the Width as the color (please note the logarithmic scale both for Strength and Width). In Figure 8.7 we have another indirect proof of the trade-off between Depth and Strength. We can identify the clusters characterized by the highest and lowest Strength (9 and 4 respectively); by the highest and lowest Depth (2 and 9 respectively); and by the highest and lowest Width (11 and 1 respectively).

There are also clusters with relatively high combinations of two measures: cluster 10 with high Strength and Width or cluster 5 with high Depth and Width.

In Table 8.3, we report a presence score for each tag in each cluster. As we see from the "Size" column, there are larger and smaller clusters. Also, some tags attract more listeners (and more leaders) than others. For these reasons, to report just the share of leaders with a given tag in a given cluster is not meaningful. We decided to correct this ratio with the expected number of leaders with the given tag in the cluster. This is a measure known as Revealed Comparative Advantage and it is calculated as follows:

$$RCA(i,j) = \frac{freq_{i,j}}{freq_{i,*}} \Big/ \frac{freq_{*,j}}{freq_{*,*}},$$

where i is a tag, j is a cluster, $freq_{i,j}$ is the number of leaders who spread an artist tagged with tag i that are present in cluster j. In Table 8.3, for each cluster we highlighted what is the tag with the highest unexpected presence.

From Table 8.3 we obtain a description of what values of Width, Depth and Strength are generally associated with each tag. For space constraints, we report only a handful of them for the clusters with extreme values. Jazz dominates clusters 1 (with the lowest Width) and 4 (with the lowest Strength): this fact suggests that jazz is a genre for which it is not easy to influence people.

Cluster 9, with the lowest Depth but the highest Strength, is dominated by pop (that dominates also clusters 10 and 11, both with high Strength but low Depth). As a result, we can conclude that if a leader can influence some nodes to listen to a pop artist then these nodes will be very engaged with the new artist. On the other hand, it is unlikely that they will influence their friends too.

Finally, cluster 2 with the highest density has a large majority of punk leaders. From this evidence, we can conclude that punk is a genre that can achieve long chains of influence, exactly the opposite of the pop genre.

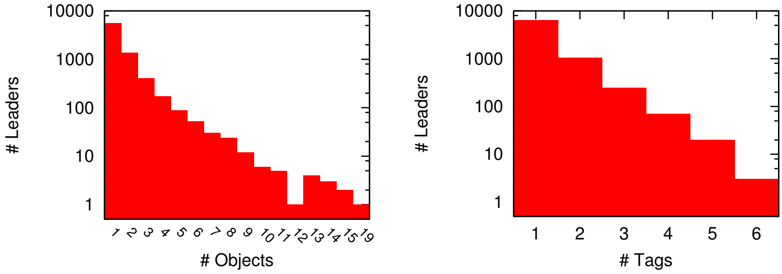


Figure 8.8: Distribution of number of objects (left) and of tags (right) per leader.

Tag Topology

In this Section we explore the connection between the Width, Depth and Strength values that we highlighted in the previous Section and the topological characteristics of the leaders for each tag. There is one caveat: a leader is not bounded to be leader just for one object ψ, but she is free to influence her neighbors with as many ψ as possible. For this reason, one leader can be counted in more than one tag. To help understand the magnitude of the issue, we depicted in Figure 8.8 the number of leaders influencing their neighbors for a given amount of actions (left) and for a given amount of tags (right). The y axis is logarithmic. Combining the information of Figure 8.8 with Figure 8.6, we conclude that the typical leader influences one neighbor for one artist. However, there is a certain amount of leaders expressing their leadership for at least 8 objects and 4 tags.

In Figure 8.9 we depict the log-binned distributions, for the leaders of each tag, of four of the topological measures studied in Section 8.3.2: Degree, Closeness Centrality, Clustering and Neighbor Degree. We omit Betweenness Centrality for its very high correlation with Degree. Overall, the distributions of the topological features appear to be very overlapping: there is no significant distinction between the tags.

The most noticeable information is carried by the Degree distributions (Figure 8.9, top left). Each one of these distributions appears very different from the overall degree distribution (Figure 8.2). There are fewer leaders

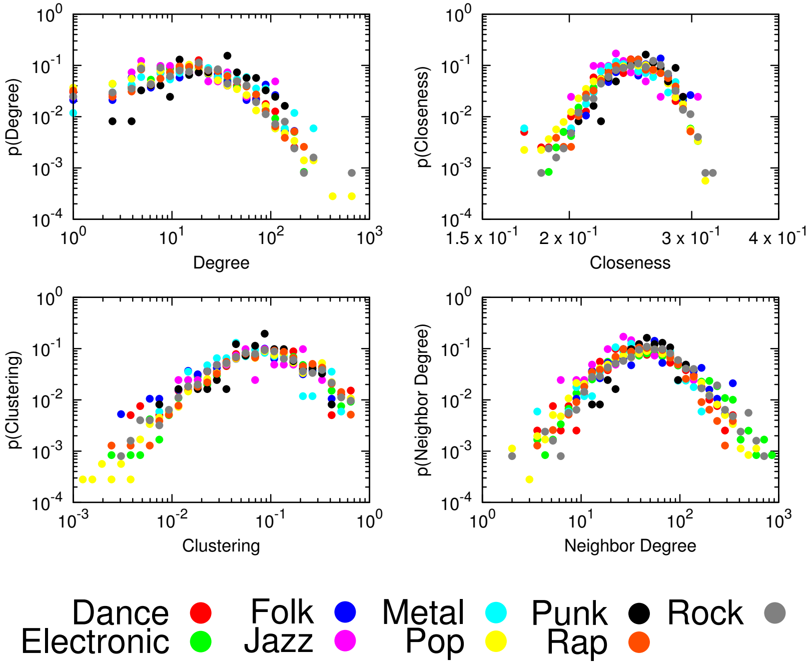


Figure 8.9: Distribution of leaders' Degree (top left), Closeness Centrality (top right), Clustering (bottom left) and Neighbor Degree (bottom right) per tag.

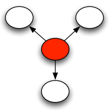
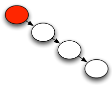
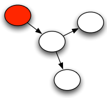
Pattern	dance	electr.	folk	jazz	metal	pop	punk	rap	rock
	3.62% (35.42%)	3.04% (22.50%)	3.94% (30.30%)	7.25% (62.50%)	4.14% (23.08%)	3.69% (32.01%)	6.56% (27.59%)	4.01% (27.97%)	4.22% (30.43%)
	2.55% (25.00%)	3.92% (29.00%)	3.15% (24.24%)	4.35% (37.50%)	4.83% (26.92%)	3.61% (31.29%)	10.66% (44.83%)	5.60% (38.98%)	4.12% (29.71%)
	3.40% (33.33%)	3.79% (28.00%)	3.94% (30.30%)	4.35% (37.50%)	6.90% (38.46%)	4.73% (41.01%)	12.30% (51.72%)	4.99% (34.75%)	4.52% (32.61%)

Table 8.4: Presence of different diffusion patterns per tag.

with low Degree than expected, therefore it appears that a high Degree increases the probability of being a leader. On the other hand, we know that central hubs have on average lower Depth and Width. As a consequence, it appears that the best leaders are the ones with an average degree, and from Figure 8.9 (top left) we see that each tag has an abundance of leaders with a Degree between 10 and 100.

Tag Patterns

Using our leaders' Minimum Diffusion Trees, we can extract some patterns that help us obtaining a complementary point of view over the diffusion of influence of different music genres. Each diffusion tree $T_{l,\psi}$ can be viewed as an entry in a graph database, allowing us to check in how many diffusion trees particular graph diffusion patterns appear. We use the VF2 algorithm[4] [32] to extract this information. To give an example, suppose we are interested in counting how frequent is the following star pattern: a leader influences three of its neighbors in the diffusion trees of pop artists. In our data, we have $5,043$ diffusion trees for pop artists, of which 581 have at least four nodes. Since the VF2 algorithm found the star pattern in 186 of these graphs, we say that it appears in 3.69% of the trees, or in 32.01% of the diffusion trees that have a sufficient number of nodes to contain it.

In Table 8.4 we report, for several tags, the results of such kind of analysis on three different patterns of four nodes: i) the star-like pattern described above, in which the leader influences three neighbors; ii) a chain where each node influences one neighbor; iii) a split where the leader influences a node, which itself influences two other neighbors. Two values are associated to each pattern and tag pair: the relative overall frequency, and the relative frequency considering only the trees with at least four nodes (in parentheses). We added the adjusted relative frequency because, as we saw in Figure 8.6, most leaders influence only one or two other users.

Please note that there is no necessary relation between the patterns

[4] Python implementation in the Networkx package `http://networkx.github.io/`

and Width, Depth and Strength measures: a low Depth does not imply the absence of the chain pattern, nor does a high Width imply a high presence of the star pattern. However, the combination of the two measures may provide some insights. For instance, we saw in Table 8.3 that jazz leaders are concentrated in the lowest Width cluster. However, many jazz leaders who affect at least three nodes tend to influence their immediate neighbors, much more than in any other genre (7.25% of all leaders, 62.5% of leaders who influence at least three other nodes). Therefore, on the one hand jazz leaders influence on average a small number of friends, however on the other hand they are likely to have at least three neighbors willing to be influenced by them.

The chain pattern is more commonly found in pop leaders than in folk ones, even though the clusters of their leaders described in Table 8.3 would suggest the opposite. It seems that pop leaders are not likely to influence nodes any further than the third degree of separation, while folk leaders tend to generate longer influence chains. Also in this case, punk leaders are commonly found in correspondence with chain patterns, just as Table 8.3 suggested.

Although pop leaders show a much greater Strength value than metal ones (by confronting in Table 8.3 their presence in high Strength clusters like 9 or 10 and low Strength clusters like 8 and 0), the split pattern tends to be more frequent in the metal genre (6.90% against 4.73% of the trees). This phenomenon suggests us that metal leaders tend to strongly influence a selected node, inducing it to spread the music to its neighbors. Pop leaders, on the other hand, affect more neighbors with higher Width and Strength, presumably flooding their ego networks with the songs they like.

8.4 Conclusion

In this Section, we presented a study of the propagation of behaviors in a social network. Instead of just studying cascade effects and the maximization of influence by a given starting seed, we decided to analyze three different dimensions: how many neighbors a leader can influence, how long the chain of triggered influences is and how engaged the influenced nodes are. We characterized each of this concept with a different measure: Width, Depth and Strength. We then created a scalable procedure to extract influence tree of action leaders from a social network and calculate each of these measures. We applied our algorithm to a real world network, extracted from the social music platform Last.Fm. Our results show that: (i) central hubs are usually incapable of having a strong effect in influencing the behavior of the entire network; (ii) there is a trade-off between how long the influence chains are and how engaged each element of the chain is; (iii) to achieve maximum engagement it is better to target leaders in tightly connected communities, although for this last point we do not have conclusive evidence. We also included a case study in which we show how artists in different musical genres are spread through the network.

Many future developments are possible. Here we studied the Width, Depth and Strength only of leaders who did not have any endogenous stimulus, but these measures can be calculated also for the influenced nodes, given that they influenced somebody else. Moreover, the limited influence that central hubs have on the overall network behavior may be studied in conjunction with the problem of network controllability [82]. Alternative leader detection techniques, such as the ones presented in [53], can be confronted with our proposed algorithm. Finally, a deeper analysis of the properties of the Width, Depth and Strength measures can be performed, using additional techniques and exploiting data from other social media services like Twitter and Facebook.

Chapter 9

Conclusion and Future Works

In this thesis we tackled the problem of developing techniques for the analysis and mining of Big Data in the application scenarios of markets. The outcomes have been achieved using methods coming from different disciplines: economics, statistics, and computer science. We organized the technical presentation of the work in two parts.

First, we discussed the implications of building a complex network over a retail market. Here the main difficulty is that the links between entities (mainly products and customers) must be built artificially by the analyst. However, we are dealing with the relatively well understood field of marketing, so the advantage is that, once the modeling phase has been done, the outcomes of the analysis can be easily understood and applied. Second, we considered some particular kind of complex networks (i.e. social and collaboration networks), where the attention is focused on people and on their relations with others. People (that are the nodes of the network) can be actors or objects of the market, and the analysis and mining over their relations can be used by intermediaries to better understand demand and supply.

The future research directions of this thesis can be mainly divided in two tracks. The first track regards the economic complexity. Even if in this thesis we considered several aspects of its applications (the hierarchy of needs, a classic marketing application and a geo-marketing application), there is a lot of space for other kind of analysis, like exploring the temporal dimension, by studying the evolution of the sophistication index over the time. Again, other kind of markets may be good fields of application for the economic complexity theory (e.g. the market of car pooling, with the bi-partite graph customer X point-of-interests; the stock exchange markets, with the bi-partite graph portfolio X stocks; or some service-based market). Moreover, the sophistication index can be used also for studies more sociological than marketing based. For example, we can imagine that the aggregated sophistication index of a community can be a good proxy of their well-being, and therefore we can build new social indicators based on this kind of measures.

The second track regards the diffusion models over the social networks. For example, the model proposed in Chapter 8 can be used to identify the leaders in a market. Once we identified these leaders, we can study their topological and personal characteristics. These findings can be used to identify new nodes, that can be considered the starting points of the simulation models typical of influence spreading studies (like SIR/SIS models). Again, the percolation model can be applied at several other market dynamics, like the search of innovators in a market of good/services and the study of their behavior. Finally, another natural field of application for influence spreading in a network-based market, is the churn and retention analysis. At the present, the better way to perform churn analysis is to observe the behavior of a customer and forecast with some change in this behavior when he/she is leaving the market. We can imagine a new framework that adds at this typical input of the model the information about what his neighborhood is doing, capturing beforehand the exogenous signals coming to the customer, letting the analyst more efficient in predicting the abandon.

All the future scenarios, anyway, are strongly related to the availability of data. For this reason it is important to study also the aspects regarding the incentives for people to share his own data (a recent and interesting proposal is represented by the personal data store model), and of course novel and efficient techniques for preserving the privacy of data maker (especially when the attributes are very sensitive). These two last points, that need to be addressed by companies, data scientists, lawyers and economists, represent the basis for the next inflow of data in this field, and for this reason they are considered a must for future research in application of complex networks analysis on big data for economics and markets.

References

[1] Rakesh Agrawal, Tomasz Imielinski, and Arun N. Swami. Mining association rules between sets of items in large databases. In *SIGMOD International Conference*, pages 207–216, Washington, D.C., 1993. 2, 6, 8, 9, 19, 25, 27, 69

[2] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, pages 487–499, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc. 2

[3] Rakesh Agrawal and Ramakrishnan Srikant. Mining sequential patterns. In *Proceedings of the Eleventh International Conference on Data Engineering*, ICDE '95, pages 3–14, Washington, DC, USA, 1995. IEEE Computer Society. 2

[4] Yong-Yeol Ahn, James P Bagrow, and Sune Lehmann. Link communities reveal multiscale complexity in networks. *Nature*, 466(7307):761–764, 2010. 9, 68, 74

[5] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1 edition, February 1993. 94

[6] M. AlmeidaNeto, P. Guimarães, Pr G. Jr, and Rd. A consistent metric for nestedness analysis in ecological systems: reconciling concept and measurement. *Oikos*, (March):1–13, 2008. 28

[7] Aris Anagnostopoulos, Ravi Kumar, and Mohammad Mahdian. Influence and correlation in social networks. *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining KDD 08*, 10(1):7, 2008. 12

[8] W-H Au, Keith CC Chan, and Xin Yao. A novel evolutionary data mining algorithm with applications to churn prediction. *Evolutionary Computation, IEEE Transactions on*, 7(6):532–545, 2003. 2

[9] Lars Backstrom, Dan Huttenlocher, Jon Kleinberg, and Xiangyang Lan. Group formation in large social networks: membership, growth, and evolution. *Science*, pages(9):44–54, 2006. 12

[10] Geoffrey H. Ball and David J. Hall. A clustering technique for summarizing multivariate data. *Behavioral Science*, 12(2):153–155, 1967. 3

[11] Jordi Bascompte, Pedro Jordano, Carlos J. Melián, and Jens M. Olesen. The nested assembly of plantanimal mutualistic networks. *PNAS*, 100(16):9383–9387, August 2003. 7, 28

[12] Michele Berlingerio, Michele Coscia, and Fosca Giannotti. Mining the temporal dimension of the information propagation. In *IDA*, pages 237–248, 2009. 123

[13] Michele Berlingerio, Michele Coscia, Fosca Giannotti, Anna Monreale, and Dino Pedreschi. Foundations of multidimensional network analysis. In *ASONAM*, pages 485–489, 2011. 88, 89

[14] Michael J. A. Berry and Gordon S. Linoff. *Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management*. *Wiley Computer Publishing, 2 edition, April 2004. 2, 8

[15] Daniel Boley. Principal direction divisive partitioning. *Data Mining and Knowledge Discovery*, 2:325–344, 1997. 3

[16] Daniel J Brass. A social network perspective on human resources management. *Research in Personnel and Human Resources Management*, 13(1):39–79, 1995. 9

[17] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA, 1984. 3

[18] Piotr Bródka, Przemysław Kazienko, Katarzyna Musiał, and Krzysztof Skibicki. Analysis of neighbourhoods in multi-layered dynamic social networks. *Int. Journal of Computational Intelligence Systems*, 5(3):582–596, 2012. 88

[19] Ronald S Burt. Social contagion and innovation: Cohesion versus structural equivalence. *American Journal of Sociology*, 92(6):1287–1335, 1987. 11

[20] Malú Castellanos, Umeshwar Dayal, Meichun Hsu, Riddhiman Ghosh, Mohamed Dekhil, Yue Lu, Lei Zhang, and Mark Schreiman. Lci: a social channel analysis platform for live customer intelligence. In *SIGMOD Conference*, pages 1049–1058, 2011. 7

[21] M. Cha, H. Haddadi, F. Benevenuto, and K.P. Gummadi. Measuring user influence in twitter: The million follower fallacy. In *AAAI Conference on Weblogs and Social Media (ICWSM)*, 2010. 123

[22] Meeyoung Cha, Alan Mislove, and Krishna P Gummadi. A measurement-driven analysis of information propagation in the flickr social network. *Proceedings of the 18th international conference on World wide web WWW 09*, 2647(1):721, 2009. 12

[23] Surajit Chaudhuri and Vivek R. Narasayya. New frontiers in business intelligence. *PVLDB*, 4(12):1502–1503, 2011. 19

[24] Sanjay Chawla. Feature selection, association rules network and theory building. *Journal of Machine Learning Research - Proceedings Track*, 10:14–21, 2010. 6, 8, 9

[25] Eunjoon Cho, Seth A. Myers, and Jure Leskovec. Friendship and mobility: user movement in location-based social networks. In *KDD*, pages 1082–1090, 2011. 8

[26] Kyung-Hee Choi and Steven E Gregorich. Social network influences on male and female condom use among women attending family planning clinics in the united states. *Sex Transm Dis*, 36(12):757–62, 2009. 107

[27] Nicholas A Christakis and James H Fowler. The spread of obesity in a large social network over 32 years. *The New England Journal of Medicine*, 357(4):370–379, 2007. 12

[28] Nicholas A Christakis and James H Fowler. The collective dynamics of smoking in a large social network. *The New England Journal of Medicine*, 358(21):2249–2258, 2008. 12

[29] W. Christaller. *Die zentralen Orte in Süddeutschland: Eine ökonomisch-geographische Untersuchung über die Gesetzmässigkeit der Verbreitung und Entwicklung der Siedlungen mit städtischen Funktionen*. University Microfilms, 1933. 51

[30] Edith Cohen, Mayur Datar, Shinji Fujiwara, Aristides Gionis, Piotr Indyk, Rajeev Motwani, Jeffrey D. Ullman, and Cheng Yang. Finding interesting associations without support pruning. In *ICDE*, pages 489–500, 2000. 6

[31] Vittoria Colizza, Alain Barrat, Marc Barthelemy, Alain-Jacques Valleron, and Alessandro Vespignani. Modeling the worldwide spread of pandemic influenza: Baseline case and containment interventions. *PLoS Medicine*, 4(1):e13, 2007. 11, 107

[32] L. P. Cordella, P. Foggia, C. Sansone, and M. Vento. A (sub)graph isomorphism algorithm for matching large graphs. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(10):1367–1372, 2004. 129

[33] Michele Coscia, Fosca Giannotti, and Dino Pedreschi. A classification for community discovery methods in complex networks. *Statistical Analysis and Data Mining*, 4(5):512–546, 2011. 9, 67, 71, 83

[34] Michele Coscia, Salvatore Rinzivillo, Fosca Giannotti, and Dino Pedreschi. Optimal spatial resolution for the analysis of human mobility. *ASONAM*, 2012. 8

[35] Michele Coscia, Giulio Rossetti, Fosca Giannotti, and Dino Pedreschi. Demon: a local-first discovery method for overlapping communities. In *KDD*, pages 615–623, 2012. 9

[36] Michele Coscia, Giulio Rossetti, Diego Pennacchioli, Damiano Ceccarelli, and Fosca Giannotti. ”you know because i know”: a multidimensional network approach to human resources problem. In *ASONAM*, pages 434–441, 2013. xvii, 85

[37] P. Diamond and H. Vartiainen. *Behavioral Economics and Its Applications*. Princeton University Press, 2008. 8

[38] David A. Easley and Jon M. Kleinberg. *Networks, Crowds, and Markets - Reasoning About a Highly Connected World*. Cambridge University Press, 2010. 8

[39] Martin Ester, Rong Ge, Wen Jin, and Zengjian Hu. A microeconomic data mining problem: customer-oriented catalog segmentation. In *KDD*, pages 557–562, 2004. 8

[40] Martin Ester, Hans peter Kriegel, Jrg S, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. pages 226–231. AAAI Press, 1996. 3

[41] Eduardo Fernandez-Huerga. The economic behavior of human beings: The institutional/post-keynesian model. In *Journal of Economic Issues*, volume 42, page 709, 2008. 8, 49

[42] Emilio Ferrara. Community structure discovery in facebook. *International Journal of Social Network Mining*, 1(1):67–90, 2012. 9

[43] Gerald R. Ferris, Wayne A. Hochwarter, Ceasar Douglas, Fred R. Blass, Robert W. Kolodinsky, and Darren C. Treadway. Social influence processes in organizations and human resources systems. 21:65 – 127, 2002. 10

[44] Alvis Cheuk M. Fong, Baoyao Zhou, Siu Cheung Hui, Jie Tang, and Guan Hong. Generation of personalized ontology based on consumer emotion and behavior analysis. *T. Affective Computing*, 3(2):152–164, 2012. 8

[45] Santo Fortunato and Claudio Castellano. Community structure in graphs. In *Computational Complexity*, pages 490–512. Springer, 2012. 9, 67

[46] J H Fowler and N A Christakis. Dynamic spread of happiness in a large social network: longitudinal analysis over 20 years in the framingham heart study. *Bmj Clinical Research Ed.*, 337(2):a2338–a2338, 2008. 12

[47] Prasad Gabbur, Sharath Pankanti, Quanfu Fan, and Hoang Trinh. A pattern discovery approach to retail fraud detection. In *KDD*, pages 307–315, 2011. 7

[48] Liqiang Geng and Howard J. Hamilton. Interestingness measures for data mining: A survey. *ACM Comput. Surv.*, 38(3):9+, 2006. 7

[49] Gourab Ghoshal and Albert Barabási. Ranking stability and super-stable nodes in complex networks. *Nature Communications*, 2:394+, 2011. 100

[50] Jacob Goldenberg, Barak Libai, and Eitan Muller. Using complex systems analysis to advance marketing theory development: Modeling heterogeneity effects on new product growth through stochastic cellular automata. *Academy of Marketing Sci. Rev.*, 2001(9). 12

[51] Marta C. Gonzalez, Cesar A. Hidalgo, and Albert-Laszlo Barabasi. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, June 2008. 65

[52] Michele Gorgoglione, Umberto Panniello, and Alexander Tuzhilin. The effect of context-aware recommendations on customer purchasing behavior and trust. In *RecSys*, pages 85–92, 2011. 8

[53] Amit Goyal, Francesco Bonchi, and Laks V. S. Lakshmanan. Discovering leaders from community actions. In *CIKM*, pages 499–508, 2008. 131

[54] Mark Granovetter. Threshold models of collective behavior. *American Journal of Sociology*, 83(6):1420–1443, 1978. 12, 123

[55] Ramaswamy Hariharan, Ji Meng Loh, James Shanahan, and Kenji Yamada. Spatial probabilistic modeling of calls to businesses. In *GIS*, pages 466–469, 2010. 8

[56] T.H. Haveliwala. Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. *Knowledge and Data Engineering, IEEE Transactions on*, 15(4):784–796, 2003. 10, 97

[57] C. A. Hidalgo, B. Klinger, A. L. Barabási, and R. Hausmann. The product space conditions the development of nations. *Science*, 317(5837):482–487, July 2007. 7

[58] César A. Hidalgo and Ricardo Hausmann. The building blocks of economic complexity. *Proceedings of the National Academy of Sciences*, 106(26):10570–10575, June 2009. 7, 33, 65

[59] Shawndra Hill, Foster Provost, and Chris Volinsky. Network-based marketing: Identifying likely adopters via consumer networks. *Statistical Science*, 22(2):256–275, 2006. 8

[60] Martin Hollis and Edward J. Nell. *Rational Economic Man*. Cambridge University Press, 2007. 8, 49

[61] Liangjie Hong, Amr Ahmed, Siva Gurumurthy, Alexander J. Smola, and Kostas Tsioutsiouliklis. Discovering geographical topics in the twitter stream. In *WWW*, pages 769–778, 2012. 8

[62] Shin-Yuan Hung, David C. Yen, and Hsiu-Yu Wang. Applying data mining to telecom churn management. *Expert Systems with Applications*, 31(3):515 – 524, 2006. 2

[63] Hyunseok Hwang, Taesoo Jung, and Euiho Suh. An ltv model and customer segmentation based on customer value: a case study on the wireless telecommunication industry. *Expert systems with applications*, 26(2):181–188, 2004. 2

[64] Warren L. Davis IV, Peter Schwarz, and Evimaria Terzi. Finding representative association rules from large rule collections. In *SDM*, pages 521–532, 2009. 19

[65] Di Jin, Dayou Liu, Bo Yang, and Jie Liu. Fast complex network clustering algorithm using agents. In *Dependable, Autonomic and Secure Computing, 2009. DASC'09. Eighth IEEE International Conference on*, pages 615–619. IEEE, 2009. 9

[66] G. V. Kass. An Exploratory Technique for Investigating Large Quantities of Categorical Data. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 29(2):119–127, 1980. 3

[67] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM, 2003. 12

[68] W O Kermack and A G McKendrick. A contribution to the mathematical theory of epidemics. *Proceedings of the Royal Society of London Series A*, 115(772):700–721, 1927. 11

[69] M. Y. Kiang and A. Kumar. A comparative analysis of an extended som network and k-means analysis. *Int. J. Know.-Based Intell. Eng. Syst.*, 8(1):9–15, 2004. 125

[70] Su-Yeon Kim, Tae-Soo Jung, Eui-Ho Suh, and Hyun-Seok Hwang. Customer segmentation and strategy development based on customer lifetime value: A case study. *Expert Systems with Applications*, 31(1):101–107, 2006. 2

[71] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, September 1999. 10

[72] Ipek Deveci Kocakoç and Sabri Erdem. Business intelligence applications in retail business: Olap, data mining & reporting services. *JIKM*, 9(2):171–181, 2010. 7, 19

[73] Teuvo Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78:1464–1480, 1990. 125

[74] Tamara G. Kolda, Brett W. Bader, and Joseph P. Kenny. Higher-Order web link analysis using multilinear algebra. In *ICDM*, pages 242–249, Washington, DC, USA, 2005. IEEE Computer Society. 10, 97

[75] Ravi Kumar, Yury Lifshits, and Andrew Tomkins. Evolution of two-sided markets. In *WSDM*, pages 311–320, 2010. 8

[76] U.A. Kumar and Y. Dhamija. A comparative analysis of som neural network with k-means clustering algorithm. *Proceedings of IEEE International Conference on Management of Innovation and Technology*, pages 55–59, 2004. 125

[77] Andrea Lancichinetti, Santo Fortunato, and János Kertész. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11(3):033015, 2009. 9

[78] R. Lempel and S. Moran. The stochastic approach for link-structure analysis (SALSA) and the TKC effect. *Computer Networks (Amsterdam, Netherlands: 1999)*, 33(1–6):387–401, 2000. 10

[79] Haigang Li. *Applications of data warehousing and data mining in the retail industry*. 2005. 7

[80] Xiaolei Li, Jiawei Han, Xiaoxin Yin, and Dong Xin. Mining evolving customer-product relationships in multi-dimensional space. In *ICDE*, pages 580–581, 2005. 8

[81] Xutao Li, Michael K. Ng, and Yunming Ye. Har: Hub, authority and relevance scores in multi-relational data for query search. In *SDM*, pages 141–152, 2012. 10

[82] Yang-Yu Liu, Jean-Jacques Slotine, and Albert-László Barabási. Controllability of complex networks. *Nature*, 473(7346):167–173, May 2011. 48, 131

[83] Yunhao Liu, Yiyang Zhao, Lei Chen, Jian Pei, and Jinsong Han. Mining frequent trajectory patterns for activity monitoring using radio frequency tag arrays. *IEEE Trans. Parallel Distrib. Syst.*, 23(11):2138–2149, 2012. 8

[84] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. Le Cam and J. Neyman, editors, *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967. 3

[85] Heikki Mannila, Hannu Toivonen, and Inkeri Verkamo. Efficient algorithms for discovering association rules. pages 181–192. AAAI Press, 1994. 2

[86] M L Markus. Toward a 'critical mass' theory of interactive media: Universal access, interdependence and diffusion. *Communication Research*, 14(5):491–511, 1987. 12

[87] A H Maslow. A theory of human motivation. *Psychological Review*, 50(4):370–396, 1943. 20, 41

[88] E. Moretti. Estimating the social return to higher education: evidence from longitudinal and repeated cross-sectional data. *Journal of Econometrics*, 121(1-2):175–212, 2004. 87, 91

[89] Andrew J Newman, Daniel K.C Yu, and David P Oulton. New insights into retail space and format planning from customer-tracking data. *Journal of Retailing and Consumer Services*, 9(5):253 – 258, 2002. 8

[90] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45(2):167–256, 2003. 118

[91] Michaek Kwok-Po Ng, Xutao Li, and Yunming Ye. Multirank: co-ranking for objects and relations in multi-relational data. In *SIGKDD*, pages 1217–1225, New York, NY, USA, 2011. ACM. 10, 105

[92] Kim-Ngan Nguyen, Loïc Cerf, Marc Plantevit, and Jean-François Boulicaut. Multidimensional association rules in boolean tensors. In *SDM*, pages 570–581, 2011. 6

[93] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120. 10, 51

[94] Luca Pappalardo, Salvatore Rinzivillo, Zehui Qu, Dino Pedreschi, and Fosca Giannotti. Understanding the patterns of car travel. *Eur. Phys. J. Special Topics*, (215):61–73, 2013. 65

[95] Jong Soo Park, Ming-Syan Chen, and Philip S. Yu. An effective hash based algorithm for mining association rules. In Michael J. Carey and Donovan A. Schneider, editors, *SIGMOD Conference*, pages 175–186. ACM Press, 1995. 2

[96] Romualdo Pastor-Satorras and Alessandro Vespignani. Epidemic spreading in scale-free networks. *Physical review letters*, 86(14):3200–3203, 2001. 11

[97] W.M. Patefield. An efficient method of generating random rxc tables with given row and column totals. (algorithm as 159.). *Journal of the Royal Statistical Society. Series C. Applied Statistics*, 30:91–97, 1981. 31

[98] Diego Pennacchioli, Michele Coscia, Fosca Giannotti, and Dino Pedreschi. Calculating product and customer sophistication on a large transactional dataset. Technical report, 2013. xvii, 18, 52

[99] Diego Pennacchioli, Michele Coscia, and Dino Pedreschi. Overlap versus partition: Marketing classification and customer profiling in complex networks of products. In *ICDE Workshop*, 2014. xvii, 66

[100] Diego Pennacchioli, Michele Coscia, Salvatore Rinzivillo, Fosca Giannotti, and Dino Pedreschi. Explaining the product range effect in purchase data. In *Big Data*, 2013. xvii, 49

[101] Diego Pennacchioli, Michele Coscia, Salvatore Rinzivillo, Fosca Giannotti, and Dino Pedreschi. The retail market as a complex system. *Submitted for publication*, 2014. xvii, 18

[102] Diego Pennacchioli, Giulio Rossetti, Luca Pappalardo, Dino Pedreschi, Fosca Giannotti, and Michele Coscia. The three dimensions of social prominence. In *SocInfo*, pages 319–332, 2013. xvii, 107

[103] J.B. Pick. *Geographic Information Systems in Business*. Idea Group Pub., 2005. 8

[104] Martin F. Porter. An Algorithm for Suffix Stripping. *Program*, 14(3):130–137, 1980. 92

[105] J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993. 3

[106] J. Ross Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993. 63

[107] Troy Raeder and Nitesh V Chawla. Market basket analysis with networks. *Social network analysis and mining*, 1(2):97–113, 2011. 66

[108] Usha N. Raghavan, Réka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76(3):036106+, September 2007. 9

[109] Harikrishna G. N. Rai, Kishore Jonna, and P. Radha Krishna. Video analytics solution for tracking customer locations in retail shopping malls. In *KDD*, pages 773–776, 2011. 8

[110] Martin Rosvall and Carl T Bergstrom. Multilevel compression of random walks on networks reveals hierarchical organization in large integrated systems. *PloS One*, 6(4):e18209, 2011. 9, 68, 72

[111] Bryce Ryan and Neal C Gross. The diffusion of hybrid seed corn in two iowa communities. *Rural Sociology*, 8(1):15–24, 1943. 12

[112] Maximilian Schich, Sune Lehmann, and Juyong Park. Dissecting the canon: Visual subject co-popularity networks in art research. In *ECCS2008*, September 2008. 48

[113] Zuo-Jun M. Shen and Xuanming Su. Customer behavior modeling in revenue management and auctions: A review and new research opportunities. *Production and Operations Management*, 16(6):713–728, 2007. 48

[114] Bai-En Shie, Hui-Fang Hsiao, Philip S. Yu, and Vincent S. Tseng. Discovering valuable user behavior patterns in mobile commerce environments. In *PAKDD Workshops*, pages 77–88, 2011. 8

[115] Antonis Sidiropoulos and Yannis Manolopoulos. Generalized comparison of graph-based ranking algorithms for publications and authors. *J. Syst. Softw.*, 79(12), 2006. 10, 102

[116] Ramakrishnan Srikant and Rakesh Agrawal. Mining sequential patterns: Generalizations and performance improvements. In *Proceedings of the 5th International Conference on Extending Database Technology: Advances in Database Technology*, EDBT '96, pages 3–17, London, UK, UK, 1996. Springer-Verlag. 2

[117] RuthMaria Stock and WayneD. Hoyer. An attitude-behavior model of salespeoples customer orientation. *Journal of the Academy of Marketing Science*, 33:536–552, 2005. 8

[118] Yizhou Sun, Charu C. Aggarwal, and Jiawei Han. Relation strength-aware clustering of heterogeneous information networks with incomplete attributes. *PVLDB*, 5(5):394–405, 2012. 19

[119] Jie Tang, Jing Zhang, Ruoming Jin, Zi Yang, Keke Cai, Li Zhang, and Zhong Su. Topic level expertise search over heterogeneous networks. *Machine Learning*, 82(2):211–237, 2011. 10

[120] Adriano Veloso, Marcos André Gonçalves, and Wagner Meira Jr. Competence-conscious associative rank aggregation. *JIDM*, 2(3):337–352, 2011. 10

[121] Michael M Wagner, J Michael Robinson, Fu-Chiang Tsui, Jeremy U Espino, and William R Hogan. Design of a national retail data monitor for public health surveillance. *J Am Med Inform Assoc*, 10(5):409–418, 2003. 7

[122] Haizhou Wang and Mingzhou Song. Ckmeans.1d.dp: Optimal k-means Clustering in One Dimension by Dynamic Programming. *The R Journal*, 3(2):29–33, December 2011. 41

[123] Pu Wang, Marta C González, César A Hidalgo, and Albert-László Barabási. Understanding the spreading patterns of mobile phone viruses. *Science*, 324(5930):1071–1076, 2009. 11, 12

[124] Chih-Ping Wei, I Chiu, et al. Turning telecommunications call details to churn prediction: a data mining approach. *Expert systems with applications*, 23(2):103–112, 2002. 2

[125] Art Weinstein. *Market Segmentation: Using Demographics, Psychographics and Other Niche Marketing Techniques to Predict and Model Customer Behavior*. Probus Chicago, IL, 1994. 2

[126] Jierui Xie and Boleslaw K. Szymanski. Towards linear time overlapping community detection in social networks. February 2012. 9

[127] Jaewon Yang and Jure Leskovec. Defining and evaluating network communities based on ground-truth. In *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*, page 3. ACM, 2012. 9