Università degli Studi di Pisa

Dipartimento di Informatica
Dottorato di Ricerca in Informatica

Ph.D. Thesis

Multidimensional Network Analysis

Michele Coscia

Supervisor
Fosca Giannotti

Supervisor
Dino Pedreschi

May 9, 2012

Abstract

This thesis is focused on the study of multidimensional networks. A multidimensional network is a network in which among the nodes there may be multiple different qualitative and quantitative relations. Traditionally, complex network analysis has focused on networks with only one kind of relation. Even with this constraint, monodimensional networks posed many analytic challenges, being representations of ubiquitous complex systems in nature. However, it is a matter of common experience that the constraint of considering only one single relation at a time limits the set of real world phenomena that can be represented with complex networks. When multiple different relations act at the same time, traditional complex network analysis cannot provide suitable analytic tools. To provide the suitable tools for this scenario is exactly the aim of this thesis: the creation and study of a Multidimensional Network Analysis, to extend the toolbox of complex network analysis and grasp the complexity of real world phenomena. The urgency and need for a multidimensional network analysis is here presented, along with an empirical proof of the ubiquity of this multifaceted reality in different complex networks, and some related works that in the last two years were proposed in this novel setting, yet to be systematically defined. Then, we tackle the foundations of the multidimensional setting at different levels, both by looking at the basic extensions of the known model and by developing novel algorithms and frameworks for well-understood and useful problems, such as community discovery (our main case study), temporal analysis, link prediction and more. We conclude this thesis with two real world scenarios: a monodimensional study of international trade, that may be improved with our proposed multidimensional analysis; and the analysis of literature and bibliography in the field of classical archaeology, used to show how natural and useful the choice of a multidimensional network analysis strategy is in a problem traditionally tackled with different techniques.

To my family: my parents for their exceptional support of all my needs, and my sister being way smarter and professional than I will ever dream.

εαν μη ελπηται ανελπιστον ουκ εξευρησει, ανεξερευνητον εον και απορον.

Acknowledgments

The most important role in this thesis, for which I really need to be grateful beyond what I can express, is the one played by my supervisors. Fosca Giannotti and Dino Pedreschi are really a fundamental part of my professional career and probably all the opportunities I was able to obtain are due to their hard work.

I also need to thank the professional guiding figures of Ricardo Hausmann and Cesar A. Hidalgo, two incredibly enthusiastic and professional researchers who, even if they do not have any formal role in this thesis, have accompanied me through more than half of my period as graduate student and they are offering me the possibility of having a real impact on the world, far greater than the one I could dream before. Also Albert-Laszlo Barabasi gave me great opportunities and a great scientific environment to live in. A profound gratitude goes also to the reviewers of this thesis, Hannu Toivonen, Yong-Yeol Ahn, Paolo Ferragina and Maria Simi, whose comments greatly improved the quality of the work I'm presenting.

Among all my co-authors, the prominent figure who deserves the biggest acknowledgment is for sure Michele Berlingerio, having taught me everything I know about being a good researcher and an efficient computer scientist, losing in the process probably more time, energy and patience than he expected. A big thank also to Maximilian Schich, because I do believe that 90% of what I know about complex networks is probably due to him. I cannot stress enough also how important was to work together with Anna Monreale and Ruggero Pensa, two of the people who worked harder than one can believe, and also responsible for the preparation that lead me into starting the PhD. Also Salvo Rinzivillo was the most funny and enjoyable co-author ever, able to "make me christian", as he would say. And finally, of course, a big thank to Amedeo Cappelli.

There is an incredible amount of other people with whom I do not have a formal collaboration, nevertheless their impact of this thesis is far from negligible. They are too many, and a data mining approach is needed to cluster them into geographically separated groups. This consideration gives also the idea how ridiculous is for me to take the entire credit as sole author of this thesis, that is basically a collaborative melting pot of an incredible hive-mind: I should rather take the blame for all the errors and misunderstanding I put in it.

Pisa is the place where my career is born. Therefore I should start thanking Alessio Orlandi (for that coding summer, for Lucca Comics and for Zurich), Diego Pennacchioli (who got plenty of unrequested updates about this thesis), Roberto Trasarti (I hope he remembers the fun we had in our complex network class in Lucca), Filippo Volpini, Giulio Rossetti and Riccardo Guidotti (my beloved trio of undergraduates), Lorenzo Gabrielli (the most precise person I know), Mirco Nanni (a fellow cinephile), Chiara Renso (she always brings loads of stuff to eat and to drink from her missions) and Chiara Falchi (I probably owe her way more than a month of dinners). I will probably be killed by all the people I did not mentioned, they are that many, but it's better to stop here.

I spent an important period of my life at the Harvard Kennedy School, and I hope to continue to stay there for a long time. I then thank Catalina Prieto and Jennifer Gala, being the solution to all my problems involving the US. Stephen Kosack keeps demonstrating how a great guy is, and I am looking forward to see what our thousands of projects will became. A big thank also to Juan Jimenez, Muhammed Yildirim, Isabel Meirelles and Alex Simoes (an MIT intruder!) for the work with the Product Space. Finally a thanks also to Juan Pablo Chauvin and Jasmina Beganovic, for their appreciation of my t-shirts and the quotes on my whiteboard.

Finally, for the group in Northeastern University, the first thought goes to Sune Lehmann, for the great talks we had. And then I think about Yong-Yeol Ahn, with the hope and promise that we will be able to actually make real the collaboration we were planning. I also need to apologize to Nicholas Blumm, Dashun Wang and Chaoming Song for all the times I stole part of their desk or their chairs. And again, for all the people not mentioned here, the thesis is long enough with scientific blabbering, but if it would contain all the friendship you gifted to me, it would explode exponentially.

Damiano Ceccarelli for sure deserves a special thanks, being my only non-scientific co-author. But, more importantly, being the voice of the reason for basically everything else. Silvia Tomanin is probably my main motivator, constantly increasing the threshold of what should I do to beat her, but she will keep anyway to end up as a winner. I won't give up, by the way. I did not forget Gabriele Pastore, Francesca Aversa and all the guys from the forum, even if in these years I probably gave them less attention than I should. A big special thank also to Eleonora Grasso: our relationship was the cornerstone of three years of my life and one of the few things that kept me sane in the darkest hours of a PhD student (and there are many of those). I wish the very best for your future.

The last final "Thank you", the special one, to my family: my parents and my sister. You are simply perfect in understanding, supporting and correcting me in every occasion. I wish the world would be an easier place, where work does not bring you far from home, because being that far from where my heart is, I just feel lost.

And then there is you, Clara. The biggest, beautiful and exciting bet I have in my future. I have put everything on it.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

A complex network is a model used to represent complex interacting phenomena such as social interactions among human beings, biological reactions in organisms and technological systems. An interaction takes places when there is some sort of information or physical exchange between two actors, for example in a social network when two individuals establish a friendship or enmity link between each other. To analyze the properties and understand the behavior of these phenomena through this model in different settings is a scientific field gaining a lot of attention in the last decade. Countless different problems have been tackled and an impressive number of good solutions, algorithms and descriptions of reality, has been proposed. A very brief, and incomplete, list includes the following main topics:

- Community discovery, i.e. the decomposition of a complex network in its modular structure [78, 94];
- Link prediction, i.e. the prediction of the new relations that we will observe given the current state of the network (or the discovery of possible missing connections due to incomplete data) [74, 207];
- Flow analysis, i.e. the analysis of the structural properties of networks unveiled by different random walk strategies over the edges of the graph;
- Cascade events, i.e. the investigation over the dynamics of epidemic events changing the state of nodes through their connections [111];
- Graph motifs mining, i.e. the discovery of regularities in the connection patterns of the nodes in the network [268].

By exploiting the tools developed in the investigation of these topics, and usually combining them with each other or other analytic tools, complex network analysis has been used to tackle many specific problems. For example, link prediction algorithms can be used to predict whether a user will trust the information provided by another user in a recommendation system [170]; or flow analysis is used for web page ranking in the popular Google search engine [208].

How can we explain this amount of interest by the scientific community? First of all, complex networks are by definition complex systems. A complex system is a system composed of different parts that expresses at the global level properties that are not present in any single part taken alone. Their importance is derived by different factors. First, they are ubiquitous: complex systems are present in many different scientific fields such as biology (the brain), ecology (the Earth climate), engineering (telecommunication structures) and many more. Second, to understand what originates their global properties is non trivial and can lead to important scientific results, such as deeper understanding of how the human brain works or a prediction of the evolution of the Earth's ecosystem.

This makes complex network analysis a suitable test field for many different approaches and theories. In fact, crucial advancements in this field have been carried on by different professional figures: computer scientists, mathematicians, physicists, but also sociologists, economists and humanities scholars. Complex network analysis is a melting pot of different disciplines, where different backgrounds can find a common vocabulary and primitives, making this novel field a truly new branch of science for the next years. Thus, we have a further reason for explaining the success of complex networks: their ubiquity in modeling such different phenomena.

The clash of many different areas of expertise has lead complex network analysis to be applied to many and different problems. Novel problems and settings have been explored in recent years with this model. Clearly, to be useful a model has to represent the features of real world phenomena in a simple way, but without losing too many details in the process. Therefore, from the original simple graph, many extensions have been proposed: weighted, dynamic, asymmetric relations are now fundamental building bricks of any study aiming to unveil novel insights about the interacting phenomena in the real world. However, these extensions do not address a critical feature, present in many interacting phenomena. In fact, weighted or directed relations do not help us when we are dealing with phenomena characterized by multiple different kinds of interactions.

We are not the only researchers that raised this issue. We will see that some intuitions about the intrinsic multifaceted nature of the real world are already present in literature. But it is not necessary to perform experiments or to deeply study obscure data to understand that multiple different relations interact with each other everyday everywhere. Let us consider the case of a social network. At the present day, a person can establish a social relation with hundreds of different people. Are all these people "friends"? Is it possible to organize all these relationships in the same class? Of course not: we have relatives, sentimental relationships, work mates and several different reasons, and degrees, to call the people we know "friends" or "acquaintances".

To be just a little more formal, it is well known that complex systems show their complexity in their multifaceted dynamics. There are several different competing forces acting either independently or in a complex interaction, either in equilibrium or in disequilibrium. As for complex networks, the interplay among different relations cannot be expressed with the traditional single relational models. In particular, the simple graph, a simplified representation used in the latter years, is not enough for this increase in complexity.

In this shift of setting and representation, also the traditional complex network analysis needs to evolve and embrace the new complexity it is supposed to explain. If reality is multifaceted, or as we name it in this thesis "multidimensional", then also network analysis should be multidimensional. Here we introduce the term "dimension" to indicate a particular edge type in a complex network. It is not an equivalent of the term "relations". While each different relation is a dimension of a network, a dimension may also be a quality of the same relation, such as the different discrete points in time when the relation was present. We will address this distinction more formally in the thesis.

Just as non-linear and non-equilibrium systems needs a new paradigm for statistics, called superstatistics, multidimensional networks need new models (multigraphs, data tensors, and so on) and tools (multidimensional community discovery, multilink prediction, shortest path in multigraphs with cost modifiers, just to name some of them). This is exactly the aim of this thesis: the creation and study of a Multidimensional Network Analysis, to extend the known metaphors of complex network analysis and grasp the complexity of real world phenomena.

In this thesis we want to accomplish several objectives. First, we want to advocate the urgency and need for a multidimensional network analysis. We present an empirical proof of the ubiquity of this multifaceted reality in different complex networks. We are able to create multidimensional representations of many different interacting phenomena. We want also to let emerge the fact that these multidimensional representations are indeed more accurate than a simple monodimensional model, and/or can lead to better insights about the phenomenon being represented. The need for a multidimensional network analysis is also witnessed by many other researchers, and we provide a collection of their early works in this novel analytic setting. We also point out that these applications are indeed useful and advanced, but a common analytic ground, needed to fully understand and develop novel insights, is yet to be defined.

The preliminary steps in the definition and creation of this common ground are exactly the second main objective of this thesis. We want to tackle this problem at two different levels. We start by looking at the basic extensions of the known model: what is the new meaning of the degree in the multidimensional network analysis? What does happen to the scale free structure in a multirelational environment? What is the new relation between the degree and the number of neighbors for a node? How does multidimensionality influence the clustering coefficient or the centrality measures?

We then move our attention to the development of novel algorithms and frameworks for well-understood and useful traditional problems in complex network analysis. For example, we are interested in multidimensional community discovery, that we take as the main case study of this thesis, and therefore tackled with special attention. Traditionally, in community discovery the problem definition is to find a graph partition, clustering together densely connected sets of nodes. If we translate this problem definition in multidimensional terms, we want to find sets of nodes multidimensionally densely connected. But what does "multidimensionally dense" mean here? Does it mean that all the different relations need to be expressed for each couple of nodes? Or that is it necessary that at least one relation is expressed at a time for each couple, and it is only required that different relations connect different couples? This ambiguity will be tackled down in this thesis.

Another example is link prediction. Link prediction has a straightforward problem definition: to rank not observed edges, i.e. couples of nodes, according to how likely they are to appear in the future (or how likely they are not present due to missing data). If we have multiple relations in our network, a new dimension appears. We are not supposed to identify just a couple of nodes that have in between them an unexpected missing link, but we need also to decide in which particular relation, or set of relations. Is it sufficient to simply apply a traditional link predictor to each relation in an independent fashion? Or is it true that actually the different dimensions are influencing each other, and then a completely new framework has to be defined?

As a last example, we want to consider how to include known analysis frameworks into multidimensional network analysis. In fact, we are interested in how a dynamic framework can be included into a multidimensional formulation. If we consider time as source of dimensions, i.e. a relation established in 2012 is a dimension and the same relation in 2011 is another dimension, then with the primitives of multidimensional networks we can perform temporal analysis. This distinction unveils a characteristic of multidimensionality: it is possible to define two different classes of dimensions, the *explicit* and the *implicit* dimensions. We will explain the difference later on in the thesis.

We conclude this thesis with a third objective: an example of how useful multidimensional network analysis can be when applied to analytic real world scenarios. We chose two of them. In the first scenario, we present a network analysis approach to international economics, namely the creation and the analysis of the Product Space, i.e. a network map of products connected if they are frequently co-exported by the same countries. From this analysis, an impressive amount of useful knowledge can be extracted, leading to predictions of new products exported by the countries and even their future economic growth. The aim of this first scenario is to indicate where are the parts in which multidimensional network analysis is able to provide analytic improvements over the monodimensional analysis performed. Our second scenario is the analysis of literature and bibliography in the field of classical archaeology. In this scenario we show how natural and useful the choice of a multidimensional network analysis strategy is in a problem traditionally tackled with different techniques.

This thesis is organized as follows. Each of the aforementioned objectives constitutes a main part of the thesis. The first part, Setting the Stage, is devoted to the presentation of the urgency and need for a multidimensional network analysis. We firstly present some aspects of traditional complex network analysis in Chapter 2. Particular attention is devoted to the case of community discovery in Section 2.3, with an extensive study of the state of the art in the field. We will then briefly define in a formal way our model for multidimensional networks in Chapter 3. In Chapter 4 we explore the literature regarding multidimensional network analysis. In Chapter 5 we conclude our exploration about the need of a multidimensional network analysis by presenting many real

world examples of multidimensional networks, that will be analyzed in the following part of the thesis.

The second part, Multidimensional Network Analysis, is the core of this thesis. Here we tackle our second and main objective: the exploration of the various building bricks of multidimensionality in complex networks. We start from the bottom, by creating a simple extension mechanism to translate the most basics network measures into multidimensional basics in Chapter 6. We then take a step further in Chapter 7 by defining a collection of novel measures that acquire a meaning only in the multidimensional setting, and are trivially solved in the monodimensional case. Finally, we conclude our main section by proposing also some more advanced analysis in Chapter 8. In Section 8.1 we propose novel evaluation measures and a framework for the discovery of multidimensional communities. The other advanced analysis we consider, with a lower resolution, are generative models for multidimensional networks (Section 8.2), multidimensional link prediction (Section 8.3) and the problem of finding the shortest path in a multigraph with cost modifiers (Section 8.4).

The third part, that concludes this thesis, deals with the real world analytical examples we chose: the Product Space creation (Chapter 9) and the analysis of the co-classification network in classical archaeology, in Chapter 10.

Chapter 11 concludes the thesis, by presenting the future research directions opened by this study.

The three parts of this thesis are based on peer reviewed papers published in international conferences. From the first part, the state of the art of complex network and the main definition of the multidimensional netowrk model are inherited from [37]. In the same paper, we introduced also the basic extension to the complex network model (Chapter 6) and the Dimension Connectivity measures (Section 7.3). Dimension Relevance measures (Section 7.1) and multidimensional network null models (Section 8.2) are introduced and studied in [42]. Dimension Correlation and the mapping of temporal analysis with multidimensional networks (Section 7.2) are published in [41, 40, 43]. The community discovery problem in multidimensional complex networks (Section 2.3 for the review and Section 8.1 for the actual algorithm) has been tackled in [78, 36, 39]. The Product Space analysis (Chapter 9) has been published as a book with Harvard University and MIT [128]. Finally, the analysis of publications in Classical Archeology (Chapter 10) has been presented to scholars both from art history and computer science [233].

Part I

Setting the Stage

Chapter 2

Network Analysis

In this chapter we present the basic notions of complex network theory. We will start by explaining how a network is represented with a graph, what variants can be defined for this basic representation and what are the basic statistical properties of graphs. We then present in each section one of the main sub branches of complex network analysis in computer science. We start with the main case study of this thesis, namely the community discovery in complex network, with an extensive review of the field. We provide a novel classification of community discovery algorithms, with the aim of presenting where in this branch multidimensional networks can play an important role (and where multidimensionality is already taken into account). The other subsections are shorter and do not provide an exhaustive classification and literature review, outside the scope of this thesis. These sub branches are: network models, link analysis and information propagation. We provide for completeness also a brief overview of problems not directly tackled in the thesis, such as graph mining and privacy concerns in social networks. Where not otherwise specified, we use as basic references the review works presented in [195] and [66], which provide a more complete collection of literature references.

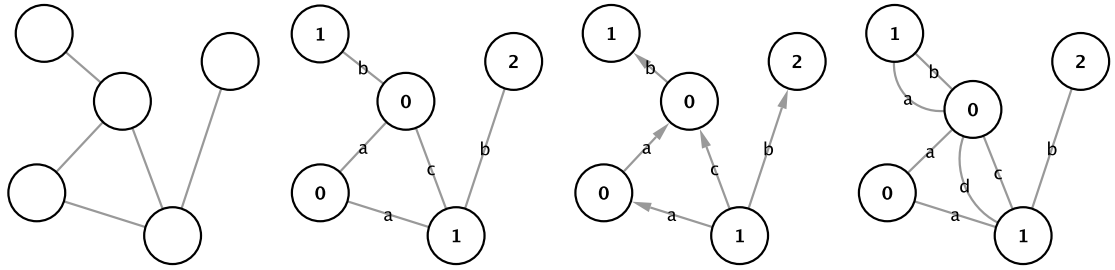
2.1 The Graph Representation

A graph is a mathematical structure used to model pairwise relations between entities from a certain collection. A network is a set of entities with connections among them. The entities are modeled as nodes. Nodes are also called vertices: in this thesis we will use the terms "node" and "vertex" interchangeably as synonyms, while the term "entity" is used to indicate what a node in the graph represents in the real world. The interactions between nodes are represented by the edges.

A set of nodes joined by edges, as depicted in Figure 2.1(a), is only the simplest type of network; there are many ways in which networks may be more complex than this. We can add to our representation additional information. Here we present a list of them, taking the example of a classical social network:

- We can add (multiple) labels both to vertices and to edges. Thus, there may be more than one different type of vertex in a network, or more than one different type of edge. For example nodes in a social network can be men or women, or they may have different nationalities, while edges may represent friendship, but they could also represent enmity. An example of labeled graph is depicted in Figure 2.1(b).

- We can add a variety of properties, numerical or otherwise, associated with vertices and edges, thus specifying some attributes. In our social network setting, people can have different ages or incomes, and edges can be weighted by the geographical proximity or by how well two people know each other.



(a) A basic graph (b) A labeled graph (c) A labeled digraph (d) A labeled multigraph

Figure 2.1: Different degrees of complexity in the graph representation.

- Edges can be directed, i.e. they point in only one direction. Graphs composed of directed edges are themselves called directed graphs or sometimes digraphs, for short. A graph representing telephone calls or email messages between individuals would be directed, since each message goes in only one direction. Directed graphs can be either cyclic, meaning they contain closed loops of edges, or acyclic meaning they do not. The labeled graph in Figure 2.1(c) has been enriched with the direction on its edges.
- The graph can be bipartite, it means that they contain vertices of two distinct types, with edges running only between unlike types. Examples are the affiliation networks in which people are joined together by common membership of groups.
- Graphs may also evolve over time, with vertices or edges appearing or disappearing, or values defined on those vertices and edges changing.
- One can also have hyperedges, i.e. edges that join more than two vertices together. Graphs containing such edges are called hypergraphs. Hyperedges could be used to indicate family ties in a social network. For example n individuals connected to each other by virtue of belonging to the same immediate family could be represented by an n-edge joining them.
- A multigraph is a graph which is permitted to have multiple edges, (also called parallel edges), that is, edges that have the same end nodes. In Figure 2.1(d) we have represented a very simple labeled multigraph. We have depicted a multigraph with only two double edges, but between the same two nodes there can be an arbitrary number of edges.

All these variants in the graph model enrich the possible representation of real world interactions events. In particular it is worth noting that the multigraph model is able to represent multidimensional data. In a multidimensional networks two interacting entities can be connected through different channels. For example in a social network two individuals can connect each other via an instant messaging software, a cellphone call, the membership in a particular website and so on. In Chapter 4 we will see how it is possible to significantly improve the precision and the relevance of a complex network analysis by considering the multidimensional nature of human relationships.

2.2 Statistical Properties

Typical social network studies address issues of centrality (which individuals are best connected to others or have most influence) and connectivity (whether and how individuals are connected to one another through the network). Aim of this section is to present statistical properties, such as path lengths and degree distributions, that are proved to characterize the structure and behavior of networked systems. When possible, we will discuss the meaning and the values taken by a metric on the toy example depicted in Figure 2.2.

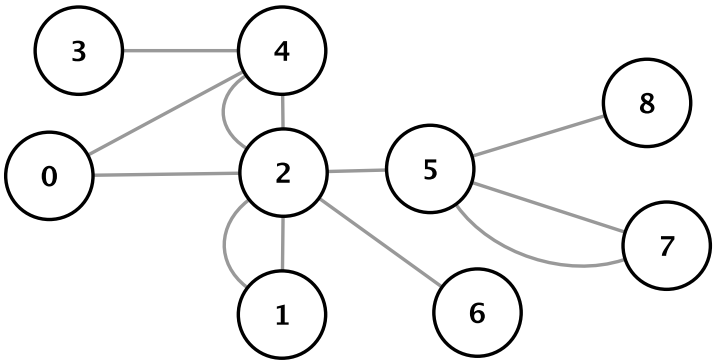


Figure 2.2: A network toy example.

The first important notion is the **degree**. In graph theory, the degree (or valency) of a vertex of a graph is the number of edges incident to the vertex, with loops counted twice. In Figure 2.2 the degree of vertex 0 is equal to 2. Note that the degree is not necessarily equal to the number of vertices adjacent to a vertex, since in a multigraph there may be more than one edge between any two vertices. This is the case of vertex 2 in our example in Figure 2.2. Its degree is equal to 7, while the number of neighbors directly reachable from it is 5. In a directed graph it is necessary to consider also the direction of the edge. Thus each vertex has both an in-degree and an out-degree, which are the numbers of in-coming and out-going edges respectively.

We define p_k to be the fraction of vertices in the network that have degree of at least k. Equivalently, p_k is the probability that a vertex chosen uniformly at random has degree k or higher. A plot of p_k for any given network can be formed by making a histogram of the degrees of vertices. This histogram represents the **degree distribution** for the network. In a random graph, see Section 2.4.1, each edge is present or absent with equal probability, and hence the degree distribution is Poisson in the limit of large graph size. Real-world networks are mostly found to be very unlike the random graph in their degree distributions. Far from having a Poisson distribution, the degrees of the vertices in most networks are highly right-skewed, meaning that their distribution has a long right tail of values that are far above the mean. These networks are called scale free networks and their degree distributions follow a power law. Scale free networks are proved to be ubiquitous [15].

A network may present a **power law degree distribution**, i.e. to contain a very high amount of nodes with extremely low degree (1 or 2) and few hubs with a very high degree. There are several explanation for this phenomenon, one of which is the rich-get-richer effect: who has already an high degree have an higher probability of obtaining new edges [30]. This means that in the network there are few nodes with a very high degree and the vast majority of nodes has a very low degree. One statistical parameter that is able to describe how strong is this effect, or in other words how is the ratio between the high degree vertices and the other low degree vertices, is the exponent of the cumulative degree distribution's slope. In other words the power law degree distribution can be approximate with $p_k \sim k^{-\alpha}$. This means that the probability that a randomly chosen vertex has degree greater or equal to k follows this law. It has been experimentally proved that in most of real word networks α takes values between 2 and 3 [195].

The **component** to which a vertex belongs is the set of vertices that can be reached from it by paths running along edges of the graph. In our toy example in Figure 2.2 we have, for sake of simplicity, only one component. In a directed graph a vertex has both an in-component and an out-component, which are the sets of vertices from which the vertex can be reached and which can be reached from it. In network theory, a giant component is a connected subgraph that contains a majority of the entire graph's nodes [58]. It has been proved that many real world social networks present a giant component, that collects from 70% to 100% of the nodes of the network. Usually, the Giant Component appears when the average degree is greater than 1 [189].

A **geodesic path** is the shortest path through the network from one vertex to another. Note that there may be, and often there is, more than one geodesic path between two vertices. In graph theory, the shortest path problem is the problem of finding a path between two vertices (or nodes) such that the sum of the weights of its constituent edges is minimized (or maximized in case the weight of the edge does not represent the cost of going from one node to the other, but the strength of the relation). One can also consider a special case of this problem, in which all edges are unweighted, or their weights are all equal to one. In this case the shortest path is the minimum number of edges to be crossed in order to go from one vertex to another. For example, in Figure 2.2 we do not have weights assigned to our edges. So the shortest path between 0 and 6 pass through node 2, so its length is equal to 2 (2 edges are crossed).

It has been discovered that most pairs of vertices in most networks seem to be connected by a short path through the network. This is the so called **small-world effect**. In practice, the values of the average length of all the geodesic paths in a network are in many cases quite small, much smaller than the number n of vertices, for instance. It typically increase as $\log n$ [262], or even shrink.

The small-world effect has obvious implications for the dynamics of processes taking place on networks. For example, if one considers the spread of information, or indeed anything else, across a network, the small-world effect implies that the spread will be fast on most real world networks. If it takes only six steps for a rumor to spread from any person to any other, for instance, then the rumor will spread much faster than if it takes a hundred steps, or a million. This affects the number of "hops" a packet must make to get from one computer to another on the physical Internet network, the number of legs of a journey for an air or train traveler, the time it takes for a disease to spread throughout a population, and so forth. Many works present in literature take advantage of this knowledge (along with the previously presented power law degree distribution) defining efficient algorithms working with these assumptions. For example, we can use the higher degree nodes in order to optimize the p2p-search task [4].

The **diameter** of a network is the length (in number of edges) of the longest geodesic path between any two vertices. A few authors have also used this term referring to the average geodesic distance in a graph, although strictly the two quantities are quite distinct. As one can see, the definition of diameter is based on the definition of geodesic path. Thus the value of this metric can change in different models of graph, for example it can be weighted or not. Usually the diameter shrinks in a growing network. This means that if we have a social network and we observe the new users and edges arrival, the distance between the most distant entities usually became smaller and smaller [173]. In the toy example depicted in Figure 2.2, the diameter is equal to 4 (starting from the most isolated vertex 3 to the other side, represented by vertex 7 or vertex 8).

The **betweenness centrality** of a vertex i is the number of geodesic paths between other vertices that run through i. Some studies have shown that betweenness appears to follow a power law for many networks and propose a classification of networks into two kinds based on the exponent of this power law [107]. Betweenness centrality can also be viewed as a measure of network resilience: it tells us how many geodesic paths will get longer when a vertex is removed from the network. In our example in Figure 2.2 we do not report the entire process needed for computing the betweenness centrality due to the lack of space, but we can give the idea that the vertices 2 and 5 are the most central in the network, because a great part of the shortest paths in the network must pass through them.

Closely related to the betweenness centrality is another centrality index, called **closeness centrality**. The closeness centrality is the average distance of a vertex from every other vertex in the network. This definition has some known issues when the network has more than one component. As diameter, both betweenness and closeness centrality are defined on the notion of shortest path, thus changing their values depending on the chosen graph model. In literature many other centrality measures are known.

Another important studied phenomenon in real world networks is the **transitivity**, recorded by the so called clustering coefficient. In many networks it is found that if vertex A is connected to vertex B and vertex B to vertex C, then there is a very high probability that vertex A will also be connected to vertex C. In the language of social networks, the friend of your friend is likely

also to be your friend. In terms of network topology, transitivity means the presence of an high number of triangles in the network, i.e. sets of three vertices each of which is connected to each of the others.

The transitivity property play a crucial role in another studied aspect of complex networks: the **community structure**, i.e. groups of vertices that have an high density of edges within them and a lower density of edges between groups. In social networks it is straightforward to verify that people do organize themselves into (overlapping) groups along lines of interest, occupation, age, and so forth. This division can be detected in the communities of a network that represents their interactions [190]. Other examples can be citation networks, in which authors would divide into groups representing particular areas of research interest [266]; or in the World Wide Web the community structure might reflect the subject matter of pages.

The detection of this particular structure inside complex networks is one of the most interesting and explored fields of research. We present in Chapter 2.3 some of the most important community detection algorithms, along with their strong points and the open problems. As we will see, also in this research track we are far from getting a definitive answer to the problem of identifying communities in a network. The definition itself of community in a network is controversial, and this stimulated further research.

2.3 Community Discovery

One critical feature of complex networks, which has been widely studied in the literature since its early stages of analysis, is the possibility of identifying groups and communities within the structure of many phenomena represented by this model. Community detection is important for many reasons, such as node classification which entails homogeneous groups, group leaders or crucial group connectors. A "Community" is usually considered to be a set of entities where each entity is closer to the other entities within the community than to the entities outside it. Communities are groups of entities that probably share common properties and/or play similar roles within the interacting phenomenon that is being represented. Communities may correspond to groups of pages of the World Wide Web dealing with related topics [92], to functional modules such as cycles and pathways in metabolic networks [123, 209], to groups of related individuals in social networks [106] and so on.

Community discovery is very similar to the clustering problem, i.e. it is a traditional data mining task. In data mining, clustering is an unsupervised learning task, which aims to assign large sets of data into homogeneous groups (clusters). In fact, community discovery can be viewed as a data mining analysis on graphs: an unsupervised classification of its nodes. In addition, community discovery is the most studied data mining application on social networks. Other applications, such as graph mining [268], are in an early phase of their development. Instead community discovery has achieved a more advanced development with contributions from different fields such as physics.

Nevertheless, this is only part of the community discovery problem. In classical data mining clustering, we have data that is not in a relational form. Thus, in this general form, the fact that the entities are nodes connected to each other through edges has not been explored much. Therefore, the concept of spatial proximity needs to be mapped between entities (i.e. vertices) in graph representation.

The traditional and most accepted definition of proximity in a network is based on the topology of its edges. In this case the definition of community is formulated according to the differences in the densities of links in different parts of the network. Many networks have been found to be non-homogeneous, consisting not of an undifferentiated mass of vertices, but of distinct groups. Within these groups there are many edges between vertices, but between groups there are fewer edges. The aim of a community detection algorithm is, in this case, to divide the vertices of a network into some number k of groups, while maximizing the number of edges inside these groups and minimizing the number of edges established between vertices in different groups. These groups are the desired communities of the network.

This definition is no longer suitable due to the increasing complexity of network representations

and of the novel analytical settings, such as the information propagation or multidimensional network analysis. For example, in a temporal evolving setting, two entities can be considered close to each other if they share a common action profile even if they are not directly connected. Thus each novel approach to community discovery has had to face this problem and has developed its own definition of community for its own solution. The underlying definition of community is the criterion that we use to classify community discovery algorithms.

In addition to the variety of different definitions of community, communities have a number of interesting features. These features can be a hierarchical or overlapping configuration of the groups inside the network. Or else the graph can include directed edges, thus giving importance to this direction when considering the relations between entities. The communities can be dynamic, i.e. evolving over time, or multidimensional, i.e. there could be multiple relations and sets of individuals that behave as isolated entities in each relation of the network, thus forming a dense community when considering all the possible relations at the same time. Or they can interact inside all relations, and still the result is a densely connected community, but with a different configuration. We tackle this problem, the ambiguity of the concept of "multidimensional density", in Section 8.1.

As a result this extreme richness of definitions and features has lead to the publication of an impressive number of excellent solutions to the community discovery problem. It is therefore not surprising that there are a number of review papers describing all these methods, such as [94]. However, existing reviews tend to analyze the different techniques from a very technical perspective. They do not consider organizing the algorithms according to their definition of community, which are many and different as acknowledged also by other papers, such as [200], in which authors say "[all the methods] require us to know what we are looking for in advance before we can decide what to measure", in which "know what we are looking for" clearly means define what a community is. To use a metaphor, existing reviews talk about bricks and mortar but not about the architectural style. Further, no one considered the problem of community discovery in a multidimensional perspective.

We have thus chosen to cluster the community discovery algorithms by considering their definition of what is a community, which depends on what kinds of groups they aim to extract from the network. For each algorithm we record the characteristics of the output of the method, thus highlighting which sets of features the reviewed algorithm is suitable or not suitable for. We also consider some general frameworks that provide both a community discovery approach and a general technique. These are applicable to other graph partitioning algorithms by adding new features to these other methods.

We now explain the classification of algorithms based on community definitions. Firstly, we report in Table 2.1 the general notation used in this section. Sometimes we need an additional notation to better explain what an algorithm exactly does. We introduce this additional notation when needed and the scope of the additional notation is limited to the paragraph of one particular algorithm. Then, before presenting the classification, we make explicit what are the problem features we consider more important for community discovery, including, of course, multidimensionality.

2.3.1 Problem Features

There are many features to be considered in the complex task of detecting communities in graph structures. In this section we present some of the features an analyst may be interested in for discovery network communities. We use them to evaluate the reviewed algorithms in Table 2.2 and also to motivate our classification.

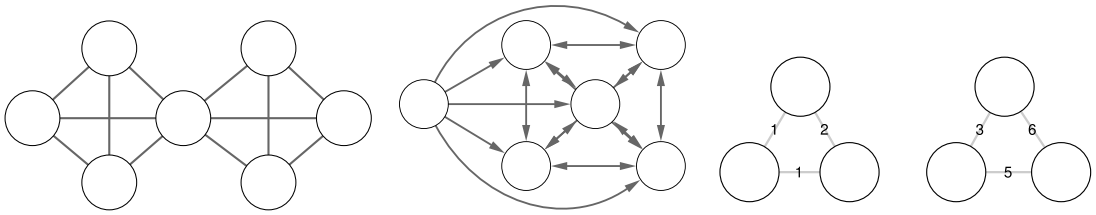
Table 2.2 records the main properties of a community discovery algorithm. These properties can be grouped into two classes. The first class considers the features of the problem representation, the second the characteristics of the approach.

Within the first class of features we group together all the possible variants in the representation of the original real world phenomenon. The most important features we consider are:

- **Overlapping**. In some real world networks, communities can share one or more common

Symbol	Description
n	Number of vertices of the network
m	Number of edges of the network
k	Number of communities of the network
$\bar{K}$	Avg degree of the network
K	Max degree in the network
T	Number of action in the network
A	Max number of actions for a node
D	Number of dimensions (if any)
c	Number of vertex types (if any)
t	Number of time step (if any)

Table 2.1: Resume of the main notation used in this section.



(a) Overlapping Communities (b) Directed Community (c) Weighted Communities

Figure 2.3: Different community features.

nodes. For example, in social networks actors may be part of different communities: work, family, friends and so on. All these communities will share a common member, and usually more since a work colleague can also be a friend outside the working environment. Figure 2.3(a) shows an example of possible overlapping community partitions: the central node is shared by the two communities. Table 2.2 indicates if an algorithm considers this feature in the "Overlap" column.

- **Directed**. Some phenomena in the real world must be represented with edges and links that are not reciprocal. This, for example, is the case of the web graph: a hyperlink from one page to another is directed and the other page may not have another hyperlink pointing in the other direction. Figure 2.3(b) shows an example in which the direction of the edges should be considered. The leftmost node is connected to the community, but only in one direction. If reciprocity is an important feature, the leftmost node should be considered outside the depicted community. See "Dir" column in Table 2.2.

- **Weighted**. A group of connected vertices can be considered as a community only if the weights of their connections are strong enough, i.e. over a given threshold. In the case of Figure 2.3(c), the left group might not be strong enough to form a community. See "Weight" column in Table 2.2.

- **Dynamic**. Edges that can appear and disappear. Thus, communities might also evolve over time. See "Dyn" column in Table 2.2.

The second class of features collects various desired properties that an approach might have. These features can specify constraints for input data, improve the expressive power of the results or facilitate the community discovery task.

- **Parameter free**. A desired feature of an algorithm, especially in data mining research, is the absence of parameters. In other words, an algorithm should be able to make explicit the knowledge that is hidden inside the data without needing any further information from the

analyst regarding the data or the problem (for instance the number of communities). See "NoPar" column in Table 2.2.

- **Multidimensional input**. This is the most important feature in the economy of this thesis. As we already know, multidimensionality in networks is an emerging topic [244, 170, 37]. When dealing with multiple dimensions, the notion of community changes. The concept of multidimensionality is used (with various names: multi-relational, multiplex, and so on) by some approaches as a feature of the input considered by the approach, as we also discussed in Chapter 4. This is the reason why multidimensionality is considered a feature of the input. However, in our opinion, multidimensionality feature should not be placed here, since what we want to extract are truly multidimensional communities. So far, no approach in the community discovery literature is able to do that, and this is the reason why the multidimensionality feature is "misplaced". We explore the idea of returning multidimensional communities in Section 8.1. See "MDim" column in Table 2.2.

- **Incremental**. Another desired feature of an algorithm is its ability to provide an output without an exhaustive search of the entire input. An incremental approach to the community discovery is to classify a node in one community by looking only at its neighborhood, or the set of nodes two hops away. Alternatively newcomers are put in one of the previously defined communities without starting the community detection process from the beginning. See "Incr" column in Table 2.2.

- **Multipartite input**. Many community discovery approaches work even if the network has the particular form of a multipartite graph. The multipartite graph, however, is not entirely a feature of the input that we might want to consider for the output. Many algorithms often use a (usually) bipartite projection of a classical graph in order to apply efficient computations. As in the case of multidimensionality, this is the reason for including the multipartite input as a feature of the approach and not of the output. See "Multip" column in Table 2.2.

There is one more "meta feature" that we consider. This is the possibility of applying the considered approach to another community discovery technique by adding new features to the "guest method". This meta feature will be highlighted with an asterisk next to the algorithm's name.

Table 2.2 also has a "Complexity" column that gives the time complexity of the methods presented. The two "BES" columns give the Biggest Experiment Size, in terms of nodes ("BESn") and edges ("BESm"), that are included in the original paper reviewed. Note that the Complexity and BES columns often offer an evaluation of the actual values, since the original work did not provide an explicit and clear analysis of the complexity or their experimental setting. A question mark indicates where evaluating the complexity would not be straightforward, or where no experimental details are provided.

2.3.2 The Definition-based classification

We now review community detection approaches. We group together the algorithms in eight classes sharing the same definition of what a community is, i.e. the same conditions satisfied by a group of entities that allow them to be clustered together in a community. This classification should help to get a higher level view of the universe of graph clustering algorithms, by uncovering a practical and reasoned point of view for those analysts seeking to obtain precise results in their analytical problems. The proposed categories are the following:

- **Feature Distance**. Here we collect all the community discovery approaches that start from the assumption that a community is composed of entities which ubiquitously share a very precise set of features, with similar values (i.e. defining a distance measure on their features, the entities are all close to each other). A common feature can be an edge or any attribute linked to the entity (in our problem definition: the action). Usually, these approaches propose

	Name	Overlap	Dir	Weight	Dyn	NoPar	MDim	Incr	Multip	Complexity	BESn	BESm	Year	Ref
Feature Distance	Evolutionary*				✓			✓		$\mathcal{O}(n^2)$	5k	?	2006	[67]
	MSN-BD			✓					✓	$\mathcal{O}(n^2ck)$	6k	3M	2006	[179]
	SocDim	✓		✓			✓			$\mathcal{O}(n^2 \log n)*$	80k	6M	2009	[245]
	PMM			✓			✓			$\mathcal{O}(mn^2)$	15k	27M	2009	[249]
	MRGC		✓		✓		✓		✓	$\mathcal{O}(nD)$	40k	?	2007	[26]
	Infinite Relational					✓	✓			$\mathcal{O}(n^{2c}D)$	160	?	2006	[144]
	Find-Tribes				✓				✓	$\mathcal{O}(mnK^2)$	26k	100k	2007	[101]
	AutoPart		✓			✓		✓		$\mathcal{O}(mk^2)$	75k	500k	2004	[64]
	Timefall				✓	✓			✓	$\mathcal{O}(mk)$	7.5M	53M	2008	[90]
	Context-specific Cluster Tree					✓			✓	$\mathcal{O}(mk)$	37k	367k	2008	[211]
IntDensity	Modularity	✓	✓	✓			✓	✓	✓	$\mathcal{O}(mk \log n)$	118M	1B	2004	[75]
	MetaFac				✓		✓			$\mathcal{O}(mnD)$	?	2M	2009	[178]
	Variational Bayes		✓			✓				$\mathcal{O}(mk)$	115	613	2008	[133]
	$LA \rightarrow IS^{2*}$	✓	✓							$\mathcal{O}(mk+n)$	16k	?	2005	[32]
	Local Density		✓			✓		✓		$\mathcal{O}(nK \log n)$	108k	330k	2005	[230]
Bridge	Edge Betweenness		✓	✓						$\mathcal{O}(m^2n)$	271	1k	2002	[106]
	CONGO*	✓		✓						$\mathcal{O}(n \log n)$	30k	116k	2008	[116]
	L-Shell	✓						✓		$\mathcal{O}(n^3)$	77	254	2005	[22]
	Internal-External Degree	✓								$\mathcal{O}(n^2 \log n)$	775k	4.7M	2009	[163]
Diffusion	Label Propagation			✓		✓		✓		$\mathcal{O}(m+n)$	374k	30M	2007	[220]
	Node Colouring			✓	✓				✓	$\mathcal{O}(ntk^2)$	2k	?	2007	[251]
	Kirchhoff	✓			✓					$\mathcal{O}(m+n)$	115	613	2004	[267]
	Communication Dynamic	✓	✓		✓			✓		$\mathcal{O}(mnt)$	160k	530k	2008	[108]
	GuruMine		✓		✓					$\mathcal{O}(TAn^2)$	217k	212k	2008	[113]
	DegreeDiscountIC		✓							$\mathcal{O}(k \log n + m)$	37k	230k	2009	[70]
	MMSB	✓	✓							$\mathcal{O}(nk)$	871	2k	2007	[14]
Close	Walktrap			✓						$\mathcal{O}(mn^2)$	160k	1.8M	2006	[215]
	DOCS	✓								?	325k	1M	2009	[263]
	Infomap		✓	✓		✓				$\mathcal{O}(m \log^2 n)$	6k	6M	2008	[227]
Structure	K-Clique	✓								$\mathcal{O}(m \frac{\ln m}{\ln n})$	20k	127k	2005	[209]
	S-Plexes Enumeration	✓								$\mathcal{O}(kmn)$	?	?	2009	[155]
	Bi-Clique	✓							✓	$\mathcal{O}(m^2)$	200k	500k	2008	[168]
	EAGLE	✓	✓	✓						$\mathcal{O}(3^{\frac{n}{3}})$	16k	31k	2009	[237]
Link	Link modularity	✓		✓					✓	$\mathcal{O}(2mk \log n)$	20k	127k	2009	[89]
	HLC*	✓		✓					✓	$\mathcal{O}(nK^2)$	885k	5.5M	2010	[12]
	Link Maximum Likelihood	✓								$\mathcal{O}(mk)$	4.8M	42M	2011	[25]
NoD	Hybrid*	✓	✓	✓		✓				$\mathcal{O}(nkK)$	325k	1.5M	2010	[86]
	Multi-relational Regression			✓			✓			?	?	?	2005	[61]
	Hierarchical Bayes									$\mathcal{O}(n^2)$	1k	4k	2008	[74]
	Expectation Maximization		✓							?	112	?	2007	[200]

Table 2.2: Resume of the community discovery methods.

this community definition in order to apply classical data mining clustering techniques, such as the Minimum Description Length principle [223, 121].

- **Internal Density**. In this group we consider the most important articles that define community discovery as a process driven by directly detecting the denser areas of the network.
- **Bridge Detection**. This section includes the community discovery approaches based on the concept that communities are dense parts of the graph among which there are very few edges that can break the network down into pieces if they are removed. These edges are "bridges" and the components of the network resulting from their removal are the desired communities.
- **Diffusion**. Here we include all the approaches to the community discovery task that rely on the idea that communities are groups of nodes that can be influenced by the diffusion of a certain properties or information inside the network. In addition, the community definition can be narrowed down to the groups that are only influenced by the very same set of diffusion sources.
- **Closeness**. A community can also be defined as a group of entities that can reach each of its own community companions with very few hops on the edges of the graph, while the entities outside the community are significantly farther apart.
- **Structure**. Another approach to community discovery is to define the community exactly as a very precise and almost immutable structure of edges. Often these structures are defined as a combination of smaller network motifs. The algorithms following this approach define some kinds of structures and then try to find them efficiently inside the graph.
- **Link Clustering**. This class can be viewed as a projection of the community discovery problem. Instead of clustering the nodes of a network, these approaches state that it is the relation that belongs to a community, not the node. Therefore they cluster the edges of the network and thus the nodes belong to the set of communities of their edges.
- **No Definition**. There are a number of community discovery frameworks which do not have a basic definition of the characteristic of the community they want to explore. Instead they define various operations and algorithms to combine the results of various community discovery approaches and then use the target method community definition for their results. Alternatively, they let the analyst define his / her own notion of community and search for it in the graph.

In each section we clarify which features in a particular community discovery category of the ones presented in the previous section are derived naturally, and which features are naturally difficult to achieve. We are not formally building an axiomatic approach, such as the one built in [150] for spatial clustering. Instead, we are using the features presented and an experimental setting to make the rationale and the properties of each category in this classification more explicit. The experiments made to support this point are presented after the classification in this section.

Where possible, we also provide a simple graphical example of the definition considered. This example provides a graphical intuition of the main properties of the given classification, in terms of the strong and weak points in particular community features.

The aim of this section is to focus on the most recent approaches and on the more general definitions of community. We are not focusing on historical approaches. Some examples of classical clustering algorithms that have not been extensively reviewed are the Kernighan-Lin algorithm [146] or the classical spectral bisection approach [217]. Thus, for a historical point of view of the community discovery problem, please refer to other review papers.

There is a sort of overlap for some community definitions. For example a definition of internal density may also include communities with sparse external links, i.e. bridges. We see in the Internal Density category that in this definition a key concept is modularity [75]. Modularity is a quality function which considers both the internal density of a community and the absence

of edges between communities. Thus methods based on modularity could be clustered in both categories. However, the underlying definition of modularity focuses on the internal density, which is the reason for the proposed classification. To give another example, a diffusion approach may detect the same communities whose members can reach each other with just a few hops. However this is not always the case: the diffusion approach may also find communities with an arbitrary distance between its members.

Many approaches in the literature do not explicitly define the communities they want to detect or, worse, they generically claim that their aim is to find dense modules of the network. This is not a problem for us, since the underlying community definition can be inferred from a high-level understanding of the approach described in the original paper. One cannot expect researchers to be able to categorize their method before an established categorization has been accepted.

In order to gain stronger evidence of the differences between the proposed categories, consider Figures 2.4, 2.6, 2.8, 2.10, 2.13 and 2.14. These figures depict the simplest typical communities that have been identified from the definitions of Feature Distance, Internal Density, Bride Detection, Diffusion, Closeness and Structure Definition, respectively. As can be seen, there are a number of differences between these examples. The Bridge Detection example (Figure 2.8) is a random graph, thus with no community structure defined for the algorithms in the Internal Density category. The Diffusion example (Figure 2.10) is also a random graph, however although the diffusion process identifies two communities, no clear bridges can be detected.

The overlap is due to the fact that many algorithms work with some general "background" meta definition of community. Further, many algorithms may present common strategies in the exploration of the search space or in evaluating the quality of their partition in order to refine it. Consider for example [161] and [222]. In these two papers there is a thorough theoretical study concerning modularity and its most general form. In [161], for example, the authors were able to derive modularity as a random walk exploration strategy, thus highlighting its overlap with the algorithms clustered here in the "Closeness" category.

Evaluating the overlap and the relationships between the most important community discovery approaches is not simple, and is outside the scope of this section. Here we focus on the connection between an algorithm and its particular definition of community. Thus we can create our useful high-level classification to connect the needs of particular analyses (i.e. the community definitions) to the tools available in the literature. To study how to derive one algorithm in terms of another, thus creating a graph of algorithms and not a classification, is an interesting open issue we leave for future research.

2.3.3 Feature Distance

In this category we review the community discovery methods that define a community according to this meta definition:

Meta Definition 1 (Feature Community) *A feature community in a complex network is a set of entities that share a precise set of features (including the edge as a feature). Defining a distance measure based on the values of the features, the entities inside a community are very close to each other, more than the entities outside the community.*

This meta definition operates according to the following meta procedure:

Meta Procedure 1 *Given a set of entities and their attributes (which may be relations, actions or properties), represent them as a vector of values according to these attributes and thus operate a matrix/spatial clustering on the resulting structure.*

Using this definition the task of finding communities is very similar to the classical clustering problem in data mining. In data mining, clustering is an unsupervised learning task. The aim of a clustering algorithm is to assign a large set of data into groups (clusters) so that the data in the same clusters are more similar to each other than any other data in any other cluster. Similarity

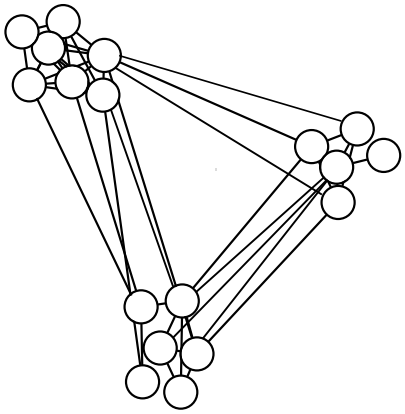


Figure 2.4: An example of a graph that can be partitioned with a notion of "distance" between its nodes.

is defined through a distance measure, usually based on the number of common features of the entities, or on similar values of these attributes.

An example of the clustering technique is K-means [143]. One natural clustering approach to the community discovery is some evolutions of co-clustering [84, 68] and/or some spectral approaches to the clustering problem [180]. In [183] there is a survey on co-clustering algorithms, while in [150] there is an interesting axiomatic framework for spatial clustering. Given the rich literature and methods to cluster matrices, community discovery approaches in this category may find clusters with virtually any feature we presented. Table 2.2 illustrates this by looking at the vast feature set for all methods present in this category. Given the fact that each node and edge is represented by a set of attributes, it is very easy to obtain multidimensional and multi-partite results by simply clustering it in a complex multidimensional space.

In order to understand the downsides of this category, consider Figure 2.4, which depicts a network whose nodes are positioned according to a distance measure. This measure could consider the direct edge connection, however it is not mandatory. The nodes are then grouped into the same community if they are close in this space (which may be highly dimensional depending on the number of features considered). Figure 2.4 shows that, depending on the number of node/edge attributes, the underlying graph structure may lose importance. This may lead to counter-intuitive results if the analyst tries to display the clusters by only looking at the graph structure, thus resulting in a lot of inter-community edges. We will discuss this point further in our experimental section.

Here we focus on some clustering techniques with some very interesting features: the Evolutionary clustering [67]; RSN-BD [179], a k-partite graph based approach; MRGC [26], that is a clustering technique working with tensors; two approaches that use modularity for the detection of latent dimensions for a multidimensional community discovery with a machine learning classifier that maximizes the number of common features ([245] and [249]); a Bayesian approach to clustering based on the predictability of the features for nodes belonging to the same group [144]; and an analysis of the shared attribute connections in a bipartite graph entity-attribute [101].

An interesting clustering principle is the Minimum Description Length principle [223, 121]. In MDL the main concept is that any regularity in the data (i.e. common features) can be used to compress it, i.e. to describe it using fewer symbols than the number of symbols needed to describe the data literally (see also [79] and [120]). The more regularities there are, the more the data can be compressed. This is a very interesting approach since, in some implementations, it enables the community discovery to be performed without setting any parameters. After considering the classical clustering approaches, in this section we also present three main algorithms that implement

a MDL community discovery approach: Autopart [64] (that is, to the best of our knowledge, the first popular community discovery that formulates the ground theory for the MDL community detection), the Context-specific cluster tree [211], and Timefall [90].

Evolutionary* [67]

In [67] the authors tackle the classical clustering problem by adding a temporal dimension. This novel situation includes several constraints:

- **Consistency**. Any insights derived from a study of previous clusters are more likely to apply to future clusters.
- **Noise Removal**. Historically consistent clustering provides greater robustness against noise by taking previous data points into effect.
- **Smoothing**. The true clusters shift over time.
- **Cluster Correspondence**. It is generally possible to place today's clusters in relation to yesterday's clusters, so the user will still be situated within the historical context.

To consider these constraints, two clustering division measures are defined: snapshot quality and history cost. The snapshot quality of C_t, a proposed cluster division, measures how well C_t represents the data at time-step t. The history cost of the clustering is a measure of the distance between C_t and C_{t-1}, the clustering used during the previous time-step.

This setting is similar to incremental clustering, but with some differences, [91]. There are two main differences. First, the focus is on optimizing a new quality measure which incorporates a deviation from history. Secondly, it works on-line (i.e. it must cluster the data during time-step t before seeing any data for time-step $t+1$), while other frameworks work on data streams [9].

This framework can be added to any clustering algorithm. The time complexity will be $\mathcal{O}(n^2)$, particularly on the agglomerative hierarchical clustering, used for the examples in the original paper, although some authors claim that a quasi-linear implementation [152] is possible. However, the framework is presented here because it is possible to apply its principles to all the other community discovery algorithms presented in this survey.

There are two framework applications worth noting. The first is FacetNet [177], in which a framework to evaluate the evolution of the communities is developed. The second one is [148], in which the concepts of nano-communities and k-clique-by-clique are introduced. These concepts are useful for assessing the snapshots and historical quality of the communities identified in various snapshots with any given method.

RSN-BD [179]

RSN-BD (Relation Summary Network with Bregman Divergence) is a community discovery approach focused on examples of real-world data that involve multiple types of objects that are related to each other. A natural representation of this setting is a k-partite graph of heterogeneous types of nodes. This method is suitable for general k-partite graphs and not only special cases such as [102]. The latter has the restriction that the numbers of clusters for different types of nodes must be equal, and the clusters for different types of objects must have one-to-one associations.

The key idea is that in a sparse k-partite graph, two nodes are similar when they are connected to similar nodes even though they are not connected to the same nodes. To spot this similarity, authors produce a derived structure (i.e. a projection) to make these two nodes closely connected. In order to do this, the authors of [179] add a small number of hidden nodes. This derived structure is called a Relation Summary Network and must be as close as possible to the original graph. They can evaluate the distance between the two structures by linking every original node with one hidden node and every hidden node couple if both hidden nodes are linked by the same original node. The distance function then sums up all the Euclidean distances between the weights of the edges in the original graph and in the transformed graph (any Bregman divergence distance function can be

used). A Bregman divergence defines a class of distance measures for which neither the triangle inequality, nor symmetry, is respected, and these measures are defined for matrices, functions and distributions [28]. The total complexity of the algorithm, as discussed by the authors, is $\mathcal{O}(n^2ck)$.

MRGC [26]

In this model, each relation between a given set of entity classes is represented as a multidimensional tensor (or data cube) over an appropriate domain, with the dimensions associated with the various entity classes. In addition, each cell in the tensor encodes the relation between a particular set of entities and can either take real values, i.e., the relation has a single attribute, or itself is a vector of attributes.

The general idea is that each node and each relation is a collection of attributes. All these attributes are a dimension of the relational space. MRGC (Multi-way Relation Graphs Clustering), basically tries to find a solution on one dimension at a time. It finds the optimal clustering with respect to each dimension by keeping every other intermediate result on the other dimensions fixed (thus its time complexity is given by the number of relations times the number of dimensions, i.e. $\mathcal{O}(mD)$). It then evaluates the solutions and keeps recalculating over all dimensions until it converges. Although defined for relation graphs, this model can be also used for identify community structures in social networks.

MRGC operates in a multi-way clustering setting where the objective is to map the set of entities in a (smaller) set of clusters by using a set of clustering functions (i.e. it is a general framework in which previous co-clustering approaches, such as [72], can be viewed as special cases). The crucial mechanism in this problem is how to evaluate the quality of the multi-way clustering in order to get to the convergence. In this case, the authors propose to measure it in terms of the approximation error or the expected Bregman distortion [27] between the original tensor and the approximate tensor built after applying the clustering function.

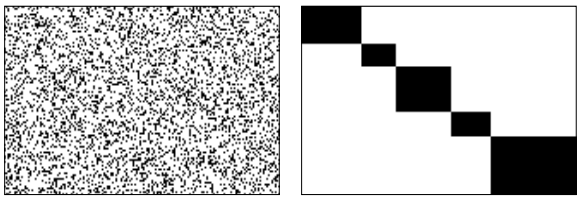
SocDim [245]

One basic (Markov) assumption in community discovery is frequently that the label of a node is only dependent on the labels of all its neighbors. SocDim tries to go beyond this assumption by building a classifier which not only considers the connectivity of a node, but assigns additional information to its connection i.e. a description of a likely affiliation between social actors. This information is called latent social dimensions and the resulting framework is based on relational learning.

In order to do this, two steps are performed by SocDim. Firstly, it extracts latent social dimensions based on network connectivity. It uses modularity in order to find in the structure of the network the dimensions in which the nodes are placed (following the homophily theory which states that actors sharing certain properties tend to form groups [185]). This can usually be done in $\mathcal{O}(n^2 \log n)$. This step may be replaced if there is already knowledge of the social dimensions. Secondly, it constructs a discriminative classifier (one-vs-rest linear [248] or structural [254] SVM): the extracted social dimensions are considered as normal features (including other possible sources) in the classical supervised learning task. It is then possible to use the predicted labels of the classifier to reconstruct the community organization of the entities. This is a multidimensional community discovery because the classifier will determine which dimensions are relevant to a class label. This work is the basis of a further evolution [246] that has an edge-centric view of communities (similar to the methods classified in the Link Clustering category)

PMM [249]

This work was originally presented in [247] and then evolved in [249]. It presents a variation of the modularity approach on a multidimensional setting. The goal of the PMM (Principal modularity Maximization) algorithm is: given a lot of different dimensions, find a concise representation of them (the authors call this step "Structural Feature Extraction", computing modularity with the



(a) The original matrix (b) Reordered matrix

Figure 2.5: An example of the MDL principle for matrices: the matrix on the left is exactly the same matrix as the one on the right, but reordered in order to describe it simply.

Lanczos method. The latter is an algorithm to find eigenvalues and eigenvectors of a square matrix [109], of complexity $\mathcal{O}(mn^2)$) and then detect the correlations between these representations (in the "Cross-Dimension Integration", using a generalized canonical correlation analysis [147]).

After this step, the authors obtain lower-dimensional embedding, which captures the principal pattern across all the dimensions of the network. They can then perform k-means [143] on this embedding to find out the discrete community assignment.

Infinite Relational [144]

Suppose there are one or more relations (i.e. edges) involving one or more types (i.e. nodes). The goal of the Infinite Relational Model is to partition each type into clusters (i.e. communities), where a good set of partitions allows relationships between entities to be predicted by their cluster assignments. The authors' goal is to organize the entities into clusters that relate to each other in predictable ways, by simultaneously clustering the entities and the relations.

Formally, suppose that the observed data are m relations involving n types. Let R^i be the ith relation, T^j be the jth type, and z^j be a vector of cluster assignments for T^j. The task is to infer the cluster assignments, and the ultimate interest lies in the posterior distribution $P(z_1, ..., z_n \mid R_1, ..., R_m)$.

To enable the IRM to discover the number of clusters in type T, the authors use a prior [214] that assigns some probability mass to all possible partitions of the type. Inferences can be made using Markov chain Monte Carlo methods to sample from the posterior on cluster assignments. This method has a very high time complexity ($\mathcal{O}(n^{2c}D)$).

Find-Tribes [101]

Find-Tribes was not explicitly developed for community discovery purposes. However, the technique can still be used to identify some kind of community. It uses a particular definition of a community, according to which the entities in a group tend to behave in the same way.

As input, the authors require a bipartite graph $G = (R \cup A, E)$ of entities R and attributes A. The entities should connect to several attributes. The aim of the algorithm is to return those groups sharing "unusual" combinations of attributes. This restriction can be easily generalized in order to also obtain the "usual" groups as outputs.

The strategy for the desired task revolves around the development of a good definition of "unusual". For an entity group to be considered anomalous, the shared attributes themselves need not be unusual, but their particular configuration should be. A projected non-bipartite graph $H'(R, F)$ is built, then for each edge a score c_{ij} (the number of attributes in the shared sequence, the number of time steps of overlap, a probabilistic Markov chain of attributes and so on) is computed, measuring how significant or unusual its sequence of shared attributes is. In the end a threshold d is chosen and all edges f_{ij} removed for which $c_{ij} < d$ are removed. The connected components of H' are the desired tribes and the overall complexity is $\mathcal{O}(mnK^2)$.

AutoPart [64]

Autopart is the basic formulation of the MDL approach to the community discovery problem. There is a binary matrix that represents associations between the n nodes of the graph (and their attributes). An example of a possible adjacency matrix is shown in Figure 2.5(a).

The main idea is to reorder the adjacency matrix so that similar nodes, i.e. nodes that are connected to the same set of nodes, are grouped with each other. The adjacency matrix should then consist of homogeneous rectangular/square blocks of a high (low) density, representing the fact that certain node groups have more (less) connections with other groups (right hand side of Figure 2.5(b)), which can be encoded with a great compression of the data. The aim of the algorithm is to identify the best grouping that minimizes the cost (compression) function [210].

A trade-off point must therefore be identified that indicates the best number of groups k. The authors solved this problem using a two-step iterative process: first, they find a good node grouping G for a given number of node groups k that minimize entropy; and second, they search for the number of node groups k by splitting the previously identified groups and verifying if there is a possible gain in the total encoding cost function, at a total time complexity of $\mathcal{O}(mk^2)$.

Context-specific Cluster Tree [211]

In this variant of the MDL approach, a binary $n_s \times n_d$ matrix represents a bipartite graph with n_s source nodes and n_d destination nodes. The aim is to automatically construct a recursive community structure of a large bipartite graph at multiple levels, namely, a Context-specific Cluster Tree (CCT). The resulting CCT can identify relevant context-specific clusters. The main idea is to subdivide the adjacency matrix into tiles, or "contexts", with a possible reordering of rows and columns, and to compress them, either as-is (if they are homogeneous enough) or by further subdividing.

The entire graph is considered as a whole community. If the best representation of the considered (sub)graph is the random graph, by testing its possible compression with a total encoding cost function, then the community cannot be split into two sub-communities. In fact, by definition the random graph has no community structure at all. Otherwise, the graph is split and the algorithm is reapplied recursively. Each edge is visited once for each subdivision (thus the complexity is $\mathcal{O}(mk)$). The result is a tree of communities in which the bottom levels are a context specialization of the generic communities at the top of the tree.

This idea of recursive clustering is also applied to streaming setting [10, 240], although with a number of parameters. This is a hierarchical evolution of the existing flat method described in [68].

Timefall [90]

Timefall is an MDL approach that can be described as a parameter-free network evolution tracking. Given n time-stamped events each related to several of m items, it simultaneously finds (a) the communities, that is, item-groups (e.g., research topics and/or research communities) and (b) a description of how the communities evolve over time (e.g., appear, disappear, split, merge), and (c) a selection of the appropriate cut-points in time when existing community structures change abruptly.

The adjacency matrix representing the graph is split according to the row timestamps. Columns are then clustered with a Cross Association algorithm [68], which is the basis of the MDL community discovery algorithms. The MDL principle is used again to connect the column clusters of the matrices across the split rows: if two column clusters can be encoded together with a low encoding cost then they are connected, ignoring time points with little or no changes. The time complexity is equal to $\mathcal{O}(mk)$.

2.3.4 Internal Density

For this group of approaches, the underlying meta definition is:

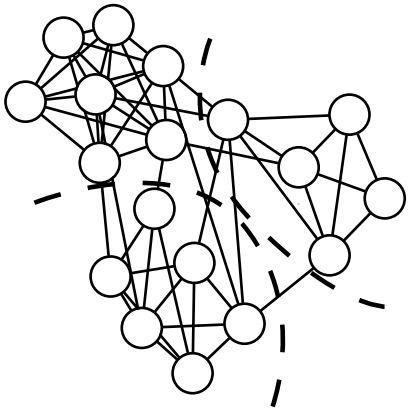


Figure 2.6: An example of a graph which can be partitioned with a notion of internal density between its nodes.

Meta Definition 2 (Dense Community) *A dense community in a complex network is a set of entities that are densely connected. In order to be densely connected, a group of vertices must have a number of edges significantly higher than the expected number of edges in a uniform random graph with the same number of vertices and edges (which has no community structure).*

Note that in this definition the community is implicitly considered as denser than its environment, an assumption that in many cases does not hold (such as for the overlap of different communities). The following meta procedure is generally shared by the algorithms in this category:

Meta Procedure 2 *Given a graph, try to expand or collapse the node partitions in order to optimize a given density function, stopping when no increment is possible.*

Figure 2.6 shows a network in which the identified communities are significantly denser than a random graph with the same degree distribution.

A key concept for satisfying this meta definition is modularity [199]. Briefly, consider dividing the graph into c non-overlapping communities. Let c_i denote the community membership of vertex v_i, k_i represents the degree of vertex i. Modularity is like a statistical test in which the null model is a uniform random graph model. In this model one entity connects to others with uniform probability. For two nodes with degree k_i and k_j respectively, the expected number of edges between the two in a uniform random graph model is $\frac{k_i k_j}{2m}$, where m is the number of edges in the graph. Modularity measures how far the interaction deviates from a uniform random graph with the same degree distribution. It is defined as:

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j),$$

where $\delta(c_i, c_j) = 1$ if $c_i = c_j$ (i.e. the two nodes are in the same community), and 0 otherwise, and A_{ij} is the number of edges between nodes i and j. A larger modularity indicates a denser within-group interaction. Note that Q could be negative if the vertices are split into bad clusters. $Q > 0$ indicates that the clustering captures some degree of community structure. Essentially, the aim is to find a community structure such that Q is maximized.

Modularity is involved in the community discovery problem on two levels. Firstly, it can quantify how good a given network partition is. It gives a result of the quality of the partition even without any knowledge of the actual communities of the network. This is especially suitable for very large networks. On the other hand, modularity is not the perfect solution for evaluating

a proposed community partition. It suffers from well known problems, in particular the resolution problem. Modularity fails to identify communities smaller than a scale that depends on the total size of the network and on the degree of interconnectedness of the communities, even in cases where modules are unambiguously defined. Furthermore, with modularity only communities extracted according to the meta definition proposed in this section can be evaluated. Any other kind of definition of communities will result in a not so meaningful evaluation by applying modularity. For an extensive review of the known problems of modularity see [94, 112].

The second level of the modularity usage in the graph partitioning task is represented by community discovery algorithms that are based on modularity maximization. These algorithms suffer from the aforementioned problems of the usage of modularity as quality measures. However, modularity maximization is a very prolific field of research, and there are many algorithms relying on heuristics and strategies for finding the best network partition.

We will present the main example of a modularity-based approach, providing references for minor modularity maximization algorithms. A good review of the eigenvector modularity based work is in [197].

Modularity is not the only cost function that is able to quantify whether a set of entities is more related than expected and thus can be considered as a community. The other reviewed methods that rely on different techniques, but share the same meta definition of community proposed in this section, are: MetaFac [178], a hypergraph factorization technique; a physical-chemical algorithm using a Bayesian approach [133]; a local density-based approach called $LA \rightarrow IS^2$ [32]; and another proposed function used to measure the internal local density of a cluster [230].

Optimizing a density function is suitable for many graph representations such as directed graphs and weighted graphs. However in addition to modularity problems, there are other weak points. For example, more complex structures are not tractable in this approach such as multidimensional networks. If multiple different qualitative relations are present in a network, how should a consistent value of "multirelational density" be computed? There are some works that scratch the surface of the ambiguity of density in multidimensional networks [36], however given the current situation none of these approaches can be used in pure multidimensional settings. Since multidimensional network analysis is the main focus of this thesis, we propose in Section 8.1 a solution to the multidimensional density ambiguity problem.

Modularity [75]

To find a partition that provides the maximum value of modularity is an NP-complete problem. Many greedy heuristics have therefore been proposed. After a pioneering work proposing modularity [196], Newman presented an efficient strategy for modularity maximization, namely repeatedly merging the two communities whose amalgamation produces the largest increase in Q. This produces a dendrogram representing the hierarchical decomposition of the network into communities at all levels, which must be cut in the modularity peak in order to obtain the communities, as depicted in Figure 2.7.

Figure 2.7 also shows another problem of modularity maximization heuristics. It has been discovered that modularity does not have a single peak given all the possible partitions, but there are several local optima. Moreover, real networks have many near-global-optima at various places [112] (the rightmost peak in Figure 2.7) and we cannot know where the algorithm locates its solution.

The optimization proposed by Clauset et al. [75] is to store a matrix containing only the values of the communities, i.e. the modularity changes when joining the communities i and j. The algorithm can now be defined as follows. Calculate the initial values of $\Delta Q_{i,j}$ and keep track of the largest element of each row of the matrix ΔQ. Select the largest $\Delta Q_{i,j}$ among these largest elements, join the corresponding communities, update the matrix ΔQ and the collection of the largest elements and increment Q by $\Delta Q_{i,j}$. Repeat this last step until the dendrogram is complete. In [169] the modularity maximization approach is adapted to the case of a directed network. We therefore have a matrix representation of the graph, but the matrix is not symmetric. The algorithm is based on [198].

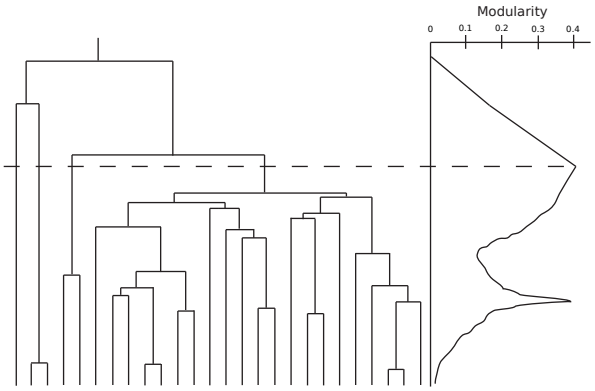


Figure 2.7: A dendrogram result for the modularity maximization algorithm, with a plot of resulting modularity values given the partition.

More recent works point to also applying the modularity approach to overlapping communities [203]. A local evaluation of modularity has also been proposed, by dividing the graph into known, boundary and unexplored sets. Two more implementations of modularity-based algorithms can be found in [8].

Another optimization of modularity-based approaches is presented in [85]. This is basically a divisive algorithm that optimizes the modularity Q using a heuristic search. This search is based on a measure (λ) that depends on the node degree, and its normalization involves all the links in the network after summation. The node selected, in an original External Optimization algorithm [23] is always the node with the worst λ_i-value. There is a τ-EO version [48] that is less sensitive to different initializations and allows escape from local maxima. A number of other optimization strategies have been proposed (size reduction [18], simulated annealing [122]).

Finally, we present the last greedy approach working with the classical definition of modularity [47]. The previous largest graph used for modularity testing was 5.5 million nodes [259], with this improvement it is possible to scale up to 100 million nodes. The algorithm is divided into two phases that are repeated iteratively. For each node i the authors consider the neighbors J of i and evaluate the gain in modularity that would take place by removing i from its community and by placing it in the community of J. The node i is then placed in the community for which this gain is maximum until no individual move can improve the modularity. The second phase consists in building a new network whose nodes are now the communities found during the first phase. It is then possible to reapply the first phase to the resulting weighted network and to iterate. This method has been tested on the UK-Union WebGraph [49], on co-citation networks [260], and on mobile phone networks.

A particularly interesting modularity framework is Multislice modularity [192]. The authors extend the null model of modularity (the random graph) to the novel multiplex setting. They use several generalizations, namely an additional parameter that controls coupling between dimensions, basing their operation on the equivalence between modularity-like quality functions and Laplacian dynamics of populations of random walkers [161]. Basically they extend Lambiotte et al.'s work by allowing multidimensional paths for the random walker ([110]), considering the different connection types with different weights ([31]), and a different spread of these weights among the dimensions ([253]).

In order to represent both snapshots and dimensions of the network, the authors use slicing. Each slice s of a network is represented by adjacency A_{ijs} between nodes i and j. The authors also specify inter-slice couplings C_{jrs} that connect node j in slice r to itself in slice s. They notate the strengths of each node individually in each slice, so that $k_{js} = \sum_i A_{ijs}$ and $c_{js} = \sum_r C_{jsr}$, and define the multislice strength $\kappa_{js} = k_{js} + c_{js}$. The authors then specify an associated multislice

null model. The resulting multislice extended definition of modularity is the following:

$$Q = \frac{1}{2\mu} \sum_{ijsr} \left\{ \left(A_{ijs} - \gamma_s \frac{k_{is} k_{js}}{2m_s} \delta_{sr} \right) + \delta_{ij} C_{jsr} \right\} \delta(c_{is}, c_{jr}).$$

In this extension γ_s is the resolution parameter, that may or may not be different for each slice. If $\gamma_s = 1$ for any s, then this formula degenerates on the usual interpretation of modularity as a count of the total weight of intra-slice edges minus the weight expected at random. Otherwise inter-slice coupling C_{jsr} is considered. C_{jsr} takes values from 0 to ∞. If $C_{jsr} = 0$ we degenerate again in the usual modularity definition. Otherwise the quality-optimizing partitions force the community assignment of a node to remain the same across all slices in which that node appears. In addition the multislice quality is reduced to that of an adjacency matrix summed over the contributions from the individual slices with a null model that respects the degree distributions of the individual contributions. The generality of this framework also enables different weights to be included across the C_{jsr} couplings. After defining the new quality function, the algorithm needed to extract communities can be one of many modularity-based algorithms.

In Table 2.2 we merged all modularity approaches on the single "Modularity" row. One caveat is that, depending on the implementation, not all the features may be returned (for example only Multislice implementation is able to consider multidimensionality).

MetaFac [178]

In this paper the concept of metagraph is introduced. The metagraph is a relational hypergraph to represent multi-relational and multi-dimensional social data. In practice, there are entities which connect to different kinds of objects in different ways (e.g.in a social media through tagging, commenting or publishing a photo, video or text). The aim is to discover a latent community structure in the metagraph, for example the common context of user actions in social media networks. In other words the authors are interested in clusters of people who interact with each other in a coherent manner. In this model, a set of entities of the same type is called a facet. An interaction between two or more facets is called a relation.

The idea of the authors is to use an M-way hyperedge to represent the interactions of M facets: each facet as a vertex and each relation as a hyperedge on a hypergraph. A metagraph defines a particular structure of interactions between facets (groups of entities of the same type), not between facet elements (the entities themselves). In order to do so, the metagraph is defined as a set of data tensors. A tensor is an array with N dimensions. This is a mathematical and computer science definition of tensors, for the notion of tensor in physics and engineering see [193]. For an extensive review of tensors, tensor decomposition and their applications and tools see [153] (in this work some examples are also provided of possible applications of tensor decompositions: signal processing [166], numerical linear algebra [165] and, closer to our area of interest, data mining [241, 242], graph analysis tasks [21, 1] and recommendation systems [71]).

Given the metagraph and its defined data tensors, the authors apply a tensor decomposition and factorization operation, which is a very hard task with a number of known issues. To the best of our knowledge, only recently have some memory and time efficient techniques been developed, such as [154]. In the metagraph approach the tensor decomposition can also be viewed as a dynamic analysis, when the sets of tensors are temporally annotated and the resulting core tensor refers to a specific time-step t. This is called metagraph factorization (for time evolving data). Finally, the MF problem can be stated in terms of optimization, i.e. minimizing a given cost function, thus obtaining facet communities (for a time complexity of $\mathcal{O}(mnD)$).

Variational Bayes [133]

In the Variational Bayes framework, a complex network is modeled as a physical system, and then the problem of assigning each node to a module (inferring the hidden membership vector) in the network is tackled by solving the disorder-averaged partition function of a spin-glass.

The authors define a joint probability by considering the number of edges present and absent within and among the K communities of a network. Traditional methods [127] need to specify K, this one is parameter free: the most probable number of modules (i.e. occupied spin states) is determined as $K = argmax_K p(K|A)$. Such methods also need to infer posterior distributions over the model parameters (i.e. coupling constants and chemical potentials) $p(\pi, \theta|A)$ and the latent module assignments (i.e. spin states) $p(\sigma|A)$. The computationally intensive solution is tackled using the variational Bayes approach [140].

This is a special case of the more general Stochastic Block Model, which is a family of solutions that reduces the community discovery problem to a statistical inference one. Historical approaches are [134, 261], while other algorithms with the same technique, but different community definitions, are presented in different categories.

$LA \rightarrow IS^2$* [32]

The authors of $LA \rightarrow IS^2$ adopt the following definition of a community: a group C of actors in a social network forms a community if its communication density function achieves a local maximum in the collection of groups that are close to C [34]. Basically, a group is a community if adding any new member to, or removing any current member from, the group decreases the average number of the communication exchanges.

This work is an evolution of [33]. It is built on two distinct phases: Link Aggregate (LA) and the real core of community detection (IS^2). The authors need a two-step approach because the IS^2 algorithm performs well at discovering communities given a good initial guess, for example when this guess is the output of another clustering algorithm, in this case called Link Aggregate (LA).

In LA, the nodes are ordered according to some criterion, for example decreasing Page Rank [208], and then processed sequentially according to this ordering. A node is added to a cluster if adding it improves the cluster density. If the node is not added to any cluster, it creates a new cluster. The complexity of this stage is $\mathcal{O}(mk + n)$.

IS^2 explicitly constructs a cluster that is a local maximum w.r.t. a density metric by starting at a seed candidate cluster and updating it by adding or deleting one node at a time as long as the metric strictly improves. The algorithm can be applied to the results of any other clustering technique, thus making this approach useful as a general framework to improve some incomplete, or approximate, results.

Local Density [230]

The Local Density algorithm is founded on the classical approach which characterizes this category, i.e. to define a density quality measure to be optimized and then recursively merge clusters if this move produces an increase in the quality function. Here this function is the internal degree of a cluster C, i.e. the number of edges connecting vertices in C to each other, $deg_{int}(\mathcal{C}) = |\{(u, v) \in E|u, v \in \mathcal{C}\}|$. Thus it is possible to define the local density of cluster as

$$\delta_l(\mathcal{C}) = \frac{2deg_{int}(\mathcal{C})}{|\mathcal{C}|(|\mathcal{C}| - 1)}.$$

Optimizing $\delta \in [0, 1]$ alone makes small cliques superior to larger but slightly sparser subgraphs, which is often impractical. For clusters to only have a few connections to the rest of the graph, one may optimize the relative density

$$\delta_r(\mathcal{C}) = \frac{deg_{int}(\mathcal{C})}{deg_{int}(\mathcal{C}) + deg_{ext}(\mathcal{C})},$$

where $deg_{ext}(\mathcal{C}) = |\{(u, v) \in E|u \in \mathcal{C}, v \in V \setminus C\}|$. The final quality measure used is $f(\mathcal{C}) = \delta_l(\mathcal{C})\delta_r(\mathcal{C})$. A good approximation of the optimal cluster for a given vertex can be obtained by a local search, guided with simulated annealing [149].

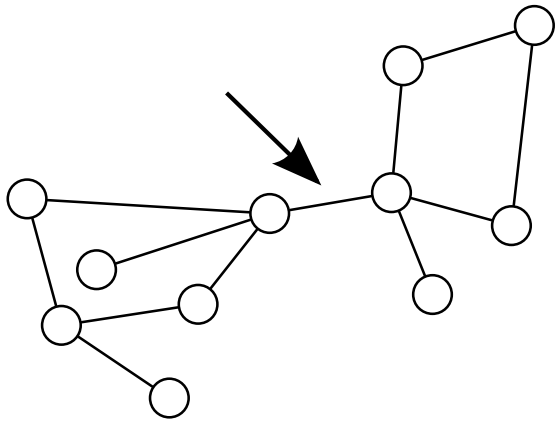


Figure 2.8: An example of a graph that can be partitioned by identifying a "bridge".

2.3.5 Bridge Detection

The meta definition of community for the algorithms in this category is:

Meta Definition 3 (Isolated Community) *An isolated community in a complex network is a component of the network obtained by removing all the sparse bridges from the structure that connect the dense parts of the network.*

Usually, approaches in this category implement the following meta procedure:

Meta Procedure 3 *Rank nodes and edges in the network according to a measure of their contribution in keeping the network connected and then remove these bridges or avoid expanding the community by including them.*

The bridge identified by the arrow in Figure 2.8 is a perfect example of an edge to be removed to decompose the network into disconnected components, which represent our communities. The main focus for these approaches is how to find these bridges (which can be both nodes or edges) inside the network. The most popular approach in this category is to use a centrality measure. No assumptions at all are made about the internal density of the identified clusters.

In a social network analysis, a centrality measure is a metric defined in order to obtain a quantitative evaluation of the structural power of an entity in a network [124]. An entity does not have power in the abstract, it has power because it can dominate others. There are a number of measures defined to capture the power of an entity in a network. These include: Degree centrality, actors who have more ties to other actors may have more favorable positions; Closeness centrality, the closer an entity is to other entities in the network, the more power it has; Betweenness centrality, the most important entity in the network is the entity present in the majority of the shortest paths between all other entities.

Here we focus on two methods based on an edge definition of the traditional node betweenness centrality: the very first edge betweenness community discovery algorithm [106], which has recently been the focus of further evolutions, i.e. a general approach that uses split betweenness in order to obtain an overlapping community discovery framework [116]. We then also consider two alternative methods [22, 163] which try to detect the bridges by expanding the community structure and computing a community fitness function.

As can be seen in Table 2.2, these algorithms are good at finding overlapping partitions (this is not true for the original edge betweenness algorithm, however basically the CONGA strategy enables it to detect overlapping clusters). The weak points of this approach appear when dealing with dynamic, multidimensional or incremental structures. We are not able to prove this point in

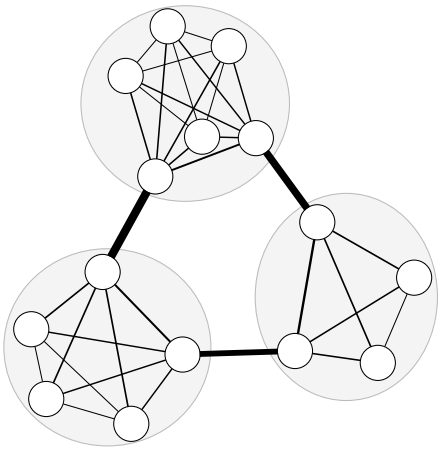


Figure 2.9: An intuitive example of the bridge detection approach. In this graph the edge width is proportional to the edge betweenness value. Wider edges are more likely to be a bridge between communities.

the experimental section so we will use an intuitive explanation. In order to compute the fitness function to detect bridges, it is necessary to start from the assumption that the algorithm has a complete view over all connections among the clusters, which may be hard in an incremental setting. Furthermore, for routing algorithms that are needed to compute the betweenness or closeness centrality, there are some constraints on the structure of the network which are not satisfied in a multidimensional setting. Consider a network with two dimensions and a rule that states that jumping from one dimension to another, lowers the cost of the path. We thus have negative cycles and a significant shortest path cannot be computed (since in Bellman-Ford's algorithm, disallowing edge repetition, it is possible to obtain a shortest path that will always cross all the negative cycles it can, thus destroying the concept of bridge [129]).

Edge Betweenness [106]

The main assumption of this work is that if a network contains communities or groups that are only loosely connected by a few inter-group edges, then all the shortest paths between different communities must go along one of these edges. In order to find these edges, which are mostly between other pairs of vertices, the authors generalize Freeman's betweenness centrality [100] to edges, and define the "edge betweenness" of an edge as the number of shortest paths between pairs of vertices that run along it. Figure 2.9 depicts an example, where the size of the edges is proportional to their edge betweenness. As can be seen, the higher edge betweenness values are taken by the edges between communities. By removing these edges, it is possible to separate one group from one another and thus reveal the underlying community structure of the graph.

This is one of the first community discovery algorithms developed after the renewed interest in social network analysis that started in the late 1990s. Previously, the traditional graph partitioning approaches constructed communities by adding the strongest edges to an initially empty vertex set (as in hierarchical clustering [265]). Here, the authors construct communities by progressively removing edges from the original graph.

While the classical implementation of the edge betweenness algorithm is $\mathcal{O}(mn)$, a speed-up for parallel systems that are linear [106] has recently been proposed. Thus without the parallel algorithm the worst case time complexity is $\mathcal{O}(m^2n)$. There are slight variations of this method using different centrality measures ([219, 258]).

CONGA [116]

CONGA (Cluster-Overlap Newman Girvan Algorithm) is based on the well-known edge betweenness community discovery algorithm [106]. It adds the ability to split vertices between communities, based on the new concept of "split betweenness".

The split betweenness [115] of a vertex v is the number of shortest paths that would pass between the two parts of v if it was split. There are many ways to split a vertex into two, the best split is the one that maximizes the split betweenness. Basically, with the following split operation, any disjoint community discovery algorithm can be applied and returns overlapping partitions ([117]):

1. Calculate edge betweenness of edges and split betweenness of vertices.
2. Remove edge with maximum edge betweenness or split vertex with maximum split betweenness, if greater.
3. Recalculate edge betweenness and split betweenness.
4. Repeat from step 2 until no edges remain.

Given a relaxed assumption on the edge betweenness computation, the total time complexity of CONGA is $\mathcal{O}(n \log n)$.

L-Shell [22]

In L-Shell algorithm, the idea is to expand a community as much as it can, stopping the expansion whenever the network structure does not allow any further expansion, i.e. the bridges are reached.

The key concept is the $l-shell$, a group of l vertices whose aim is to grow and occupy an entire community while two quantities are computed: the emerging degree and total emerging degree. The emerging degree of a vertex is defined as the number of edges that connect that vertex to vertices that the $l-shell$ has not already visited as it expanded from the previous $(l-1)$, $(l-2)$, ... $-shells$. The total emerging degree K_j of an $l-shell$ is thus the sum of the emerging degrees of all vertices on the leading edge of the $lshell$.

For a starting vertex j the algorithm starts an $l-shell$, $l=0$, at vertex j (add j to the list of community members) and computes the total emerging degree of the shell. Then it spreads the $l-shell$, $l=1$, it adds the neighbors of j to the list, and computes the new total emerging degree. Now it can compute the change in the emerging degree of the shell. If the total emerging degree is increased less than a given threshold α, then a community has been found. Otherwise it increases the size of the shell (posing $l=l+1$) until α is crossed or the entire connected component is added to the community list. As can be seen, for each node we have a quadratic problem, i.e. the time complexity is $\mathcal{O}(n^3)$. The assumption is that a community is a structure in which the total emerging degree cannot be significantly increased, i.e. the vertices at the border of the community have few edges outside it and these edges are the bridges among different communities.

Internal-External Degree [163]

An approach close to $l-shell$ starts from the similar basic assumption that communities are essentially local structures, involving the nodes belonging to the modules themselves plus at most an extended neighborhood of them. The fitness chosen here is the total internal degree of nodes on the sum of internal and external degrees to the power of a positive real-valued parameter (α). Given a fitness function, the fitness of a node A with respect to sub-graph $\mathcal{G}$, f_G, is defined as the variation of the fitness of sub-graph $\mathcal{G}$ with and without node A. The process of calculating the fitness of the nodes and them joining them together in a community stops when the nodes examined in the neighborhood of $\mathcal{G}$ all have negative fitness, i.e. their external edges are all bridges, after a total time complexity of $\mathcal{O}(n^2 \log n)$.

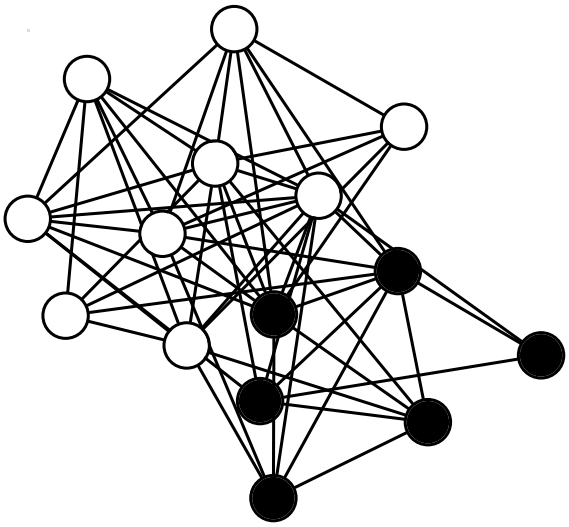


Figure 2.10: An example of graph partitioned with a diffusion process.

Large values of α yield very small communities, instead small values deliver large modules. For α=1 this method recalls [219] closely, which is another algorithm that falls into this category. Going from α=0.5 to α=2 reveals the hierarchical structure of the network.

2.3.6 Diffusion

A diffusion is a process in which vertices or edges of a graph are randomly designated as either "occupied" or "unoccupied", and the various properties of the resulting patterns of vertices are then queried [195] (see Figure 2.10, which also highlights the lack of clear bridges between communities or any density difference between the inside and the outside of clusters). A generalization of a diffusion process can be used for community discovery in complex networks, according to the following definition of community:

Meta Definition 4 (Diffusion Community) *A diffusion community in a complex network is a set of nodes that are grouped together by the propagation of the same property, action or information in the network.*

The definition of the meta procedure followed by algorithms in this category is thus:

Meta Procedure 4 *Perform a diffusion or percolation procedure on the network following a particular set of transmission rules and then group together any nodes that end up in the same state.*

According to this meta definition, a community can also be defined as a set of entities influenced by a fixed set of sources. This is important because algorithms which are not explicitly developed as approaches for graph partitioning are also considered as a community discovery method. Basically, this definition of the problem overlaps with another well-known data mining problem: influence spread and flow maximization [92], which is often used for viral marketing [171]. Preliminary ideas can be found in [96], even if only a novel centrality measure is defined, and the approach can be mapped in the Newman edge betweenness algorithm [106]. Another approach that mixes physics and information theory is [271].

Other interesting works in viral marketing are, given a community partition, the analysis of the group characteristics in order to predict their evolution [20]. In addition, it is possible to predict if a single vertex will be attached to a group, or even classify some features (and the evolution of these features) of a group. While it is not a community discovery work, [20] can be used as a framework

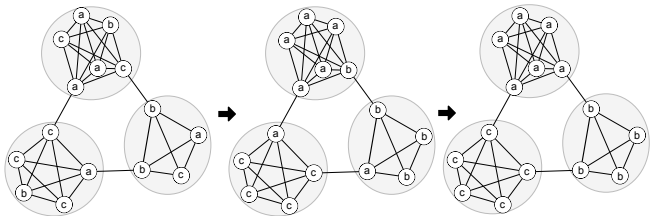


Figure 2.11: Possible steps of a label propagation-based community discoverer.

after a community detection algorithm in order to obtain a temporal evolving description of the identified groups.

To sum up, the classical community discovery diffusion-based algorithms presented here are: a label propagation technique [220], dynamic node coloring for temporal evolving communities [251], and edge resistor algorithms that consider the original graph as an electric circuit [267].

The influence propagation approaches reviewed here are: GuruMine [113], a framework whose aim is to analyze "tribes", DegreeDiscountIC [70], a classical spread maximization algorithm, and a mixed membership stochastic blockmodel algorithm [14], which uses Bayesian inferences in order to compute the final state of the influence vectors for each node in the network.

In this category, it is natural to deal with directed communities, since the diffusion process, when dealing with information spread, is naturally modeled following asymmetric relations. It is also intrinsically dynamic, thus many diffusion algorithms provide this feature in the community discovery solution. We found that no approach currently considers multidimensional networks, however we believe that considering different communication channels inside a network should be a key feature of this category.

Label Propagation [220]

Suppose that a node x has neighbors $x_1, x_2, ..., x_k$ and that each neighbor carries a label denoting the community that it belongs to. Then x determines its community based on the labels of its neighbors. A three-step example of this principle is shown in Figure 2.11.

The authors assume that each node in the network chooses to join the community to which the maximum number of its neighbors belong. As the labels propagate, densely connected groups of nodes quickly reach a consensus on a unique label. At the end of the propagation process, after a quasi-linear time complexity ($\mathcal{O}(m + n)$) nodes with the same labels are grouped together as one community.

Clearly, a node with an equal maximum number of neighbors in two or more communities can belong to both communities, thus identifying overlapping communities. It is easy to define an overlapping version of this algorithm [118].

Node coloring [251]

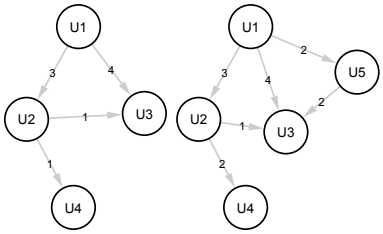
Consider an affiliation network in which some individuals form groups by attending the same event. In this approach, which represents an evolution of [35], the base input representation is an evolving bipartite graph of individuals connected to events.

Various rules have been defined to connect groups over time and form communities of groups:

1. In each time step, every group is a representative of a distinct community;
2. An individual is a member of exactly one community at any one time (but can change community affiliation over time);
3. An individual tends not to change his / her community affiliation very frequently;
4. If an individual keeps changing affiliation from one community to another, then it is not a true member of any of those communities;

User	Action	Time
U1	β	12
U5	β	14
U1	α	15
U2	β	15
U3	β	16
U4	β	17
U2	α	18
U4	α	19
U3	α	19

(a) Action Table



(b) Action α (c) Action β

Figure 2.12: The GuruMine data structures: the action table and the influence graphs.

5. An individual is frequently present in the group representing the community with which he / she is affiliated.

The authors define the community interpretation of a graph G as a function $f : V \to \mathbb{N}$. Each individual belongs to exactly one community in each time-step, and each group represents exactly one community. Thus, although the affiliation can change over time, this is a disjoint community detection algorithm, not an overlapping one. To measure the quality of a community interpretation, the authors use costs (whenever an individual changes color, or it connects to groups with different colors, and so on) to penalize violations of Rules 3 and 5. The optimization problem is then to find the valid community interpretation by minimizing the total cost resulting from the individual edges, group edges and color usage. The authors present an exhaustive global optimum algorithm with exponential time complexity (the algorithm with dynamic programming tries all possible colorings of the graph) and then some heuristics, ending up with a final complexity of $\mathcal{O}(ntk^2)$. In [250] the authors present another set of heuristics and optimizations.

Kirchhoff [267]

In this paper, the basic idea is to imagine each edge as a resistor with the same resistance. It is then possible to connect a virtual "battery" between chosen vertices so that they have fixed voltages. Having made these assumptions the graph can be viewed as an electric circuit with a current flowing through each edge (resistor). By solving Kirchhoff's equations, the authors obtain the voltage value of each node. The authors claim that, from a node's voltage value they are able to judge whether it belongs to one community or another. This approach is very efficient, since the complexity is $\mathcal{O}(m + n)$.

A further expansion [17] applies a walk-based approach in order to unveil the hidden hierarchical structure of the network and identify good choices for the seed poles. The authors then apply a very similar implementation of this method using a Kirchhoff matrix.

GuruMine [113]

The aim of GuruMine is to investigate how influence (for performing certain actions) propagates from users to their network friends, potentially recursively, thus identifying a group of users that behave homogeneously (i.e. a tribe, or a community). For instance, Table 2.12(a) shows a possible action table with two actions, α and β, and five users. Figures 2.12(b) and 2.12(c) represent the influence graphs of these two actions. $U1$ can be considered as a tribe leader in both cases. However, for action α, $U1$ cannot be considered a leader if the threshold regarding the minimum number of influenced users is equal to 4.

Since the set of influenced users is the same, we have a "tribe leader", meaning the user leads a fixed set of users (tribe) w.r.t. a set of actions, which can be considered a community. The general goal is similar to recent works such as [6, 175, 145]. However, here the input includes not just a graph (which is not edge-weighted) but also an action table which plays a central role in the definition of leaders. This action table contains a triple $(u; t; a)$ indicating that user u performed

action a at time t, from which a directed propagation graph is derived. If the composition of the influenced graph is the same, we have a tribe.

Any algorithm for extracting leaders must scan the action log table and traverse the graph (which means that the complexity also depends on this table and is $\mathcal{O}(TAn^2)$). The implementation works with only one scan, with the action log stored in chronological order. With this scan the influence matrix $IM_\pi(U;A)$ can be computed. For tribe leaders the influence cube $Users \times Actions \times Users$ is needed, with cells containing Boolean entries if user v was influenced by user u w.r.t. action a. A tribe is essentially an item-set, i.e. a community with common behavior. This phase is implemented by ExAMiner [51]. This work is part of a larger framework that also has a query interface [114].

DegreeDiscountIC [70]

This work is in the context of the classical data mining influence spread problem. The problem definition consists in deciding who to include in the initial set of targeted users so that, if necessary, they influence the largest number of people in the network. This knowledge can be used for community discovery: each seed node is the head of a community that acts uniformly, and the set of these influenced nodes is the community members. This work is an implementation of the idea in [145] and the improvement of the algorithm proposed in [175].

Influence is propagated in the classical network representation of social interactions according to a stochastic cascade model. Let S be the subset of vertices selected to initiate the influence propagation. In the cascade model (IC), let A_i be the set of vertices that are activated in the i-th round, and $A_0 = S$. For each edge with one inactive endpoint, there is a probability of activation proportional to the active neighbors, and this is repeated until the cascade cannot expand any further. Then all edges not used for propagation are removed, and the set of influenced vertices is simply the set of vertices reachable from S in G'. This cascade can be evolved in a weighted model (WC), by considering the number of inactive neighbors of an active node and the activated neighbors of an inactive node. A discount on the degree of these vertices is considered if both connected nodes are part of the seed set. With this and more finely tuned heuristics on degrees, the authors manage to develop a well performing algorithm with a reasonable level of complexity (equal to $\mathcal{O}(k \log n + m)$).

MMSB [14]

In the mixed membership stochastic blockmodel approach (MMSB), the authors implement the following mechanism: each node belongs to any possible community with a certain probability. These probabilities are then influenced by the probabilities of all other nodes. In practice, the influence of affiliations spreads over the network until convergence, by averaging the vector of probabilities of each node with the vector of the general influences. In other words, this process is equivalent to label propagation, and instead of a simple number indicating the membership there is a vector of probabilities.

The indicator vectors are in the form of $\overrightarrow{z}_{p \to q}$, which denotes the group membership of node p when it is approached by node q (note that this is not symmetric). Then, for each node i a mixed membership vector $\overrightarrow{\pi_i}$ is drawn, and the value of the interaction between this vector and the original one of the node is sampled. The authors also introduce a sparsity parameter to calibrate the importance of non-interaction.

As for other mixed membership models, this is intractable to compute. A number of approximate inference algorithms for mixed membership models have recently appeared such as mean-field variational methods [252], expectation propagation [187] and Monte Carlo Markov chain sampling [88]. In these papers, the authors apply mean-field variational methods to approximate the posterior of interest, which has a complexity of $\mathcal{O}(nk)$. An extension of this work which considers also the degree of the vertices as a normalization factor is [191]. A work very related to this one, working with a very similar notion of propagating probabilities as influence or information, is [83].

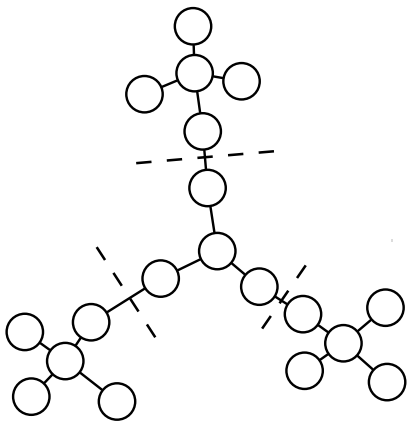


Figure 2.13: An example of a graph which can be partitioned by considering the relative distance, in terms of number of edges, among its vertices.

2.3.7 Closeness

A very intuitive notion of community in a complex network is based on the concept of how close its members are connected together. A community is a set of individuals who can communicate with each other very easily because they can reach any other member in a relatively lower number of hops than the network's average. Figure 2.13 shows a simple example of this configuration. The underlying definition of community in this case is:

Meta Definition 5 (Small World Community) *A small world community in a complex network is a set of nodes that can reach any member of its group usually by crossing a very low number of edges, significantly lower than the average shortest path in the network.*

We use the term "small world" [262] since it conveys the idea of very closely connected nodes. A very efficient approach used with this problem definition relies on random walks. A random walk is a process in which at each time step a walker is on a vertex and moves to a vertex chosen randomly and uniformly from its neighbors. The same procedure is followed for the new selected vertex. This is a Markov process. However, various strategies have been formulated in order to obtain very sophisticated random walk based application. For example, the popular link analysis PageRank algorithm [208] is based on random walks. This ends up in the following meta procedure:

Meta Procedure 5 *Given a network, perform several random walks and then cluster together nodes which appear frequently in the same walk.*

Algorithms in this category inherit the weakness in multidimensional networks from Bridge Detection algorithms, since also in this case paths are important in this community discovery category.

To the best of our knowledge there are three main community discoverers that use random walks to find communities whose members are very close to each other: Walktrap [215], based on the assumption that when performing random walks the virtual surfer is trapped in the high density regions of the graph (i.e. the communities); DOCS [263], a more complex framework that also uses modularity as a fitness function; and Infomap [227], which applies an information-theoretic approach. An older approach in this category is the Markov Cluster Algorithm [256], which is still commonly used especially in bioinformatics. It simulates a controlled flow through random walks in a network using matrix multiplication and inflation.

Walktrap [215]

The Walktrap approach is based on the following intuition: random walks are able to unveil the real distance among nodes by frequently exploring nodes in the same community. The key problem is the definition of the distance function between any two vertices, computed from the information given by random walks in the graph. High values of this measure mean that the two vertices i and j "see" the network in a very similar way, thus they belong to the same community. Therefore, this distance must be large if the two vertices are in different communities, and small otherwise. In the original paper this distance is defined as:

$$r_{ij} = \sqrt{\sum_{k=1}^{n} \frac{(P_{ik}^t - P_{jk}^t)^2}{d(k)}}$$

where P_{ik}^t is the probability to go from i to j in t steps and $d(k)$ is the degree of vertex k.

A critical parameter is the length t of the random walks: it must be sufficiently long to gather enough information regarding the topology of the graph. However it must not be too long because when the length of a random walk starting at vertex i tends towards infinity, the probability of being on a vertex j only depends on the degree of vertex j (and not on the starting vertex i).

Similar random walk approaches are [97, 270]. However they are less efficient compared to the average complexity of Walktrap, which is at the worst case $\mathcal{O}(mn^2)$.

DOCS [263]

This method is based on a spectral partition and random walk expansion, and is an extension of [264]. The general idea is to obtain an initial guess in a first step regarding the community structure, and then collapse or expand these communities according to the hints given by the random walks among them.

The first step is to coarsen the original graph into a series of higher level graphs. This is guided by modularity maximization. In the lazy random walk stage, vertices are labeled as contributing or non contributing vertices depending on whether or not they can be moved to another cluster and provide an increase in modularity. They are also sorted in a descending order by their contributing values. The target communities can then be extracted.

Infomap [227]

The Infomap algorithm is one of the most accurate community discovery methods [162]. It is based on a combination of information-theoretic techniques and random walks. The authors explore the graph structure with a number of random walks of a given length and with a given probability of jumping to a random node. This approach is equivalent to the random surfer of the PageRank algorithm [208].

Intuitively, the random walkers are trapped in a community and exit from it very rarely. Each walk is described as a sequence of steps inside a community followed by a jump. By using unique names for communities and reusing a short code for nodes inside the community, this description can be highly compressed, in the same way as re-using street names (nodes) inside different cities (communities). The renaming is done by assigning a Huffman coding to the nodes of the network. The best network partition will result in the shortest description for all the walks.

2.3.8 Structure Definition

A number of works tackle community discovery with a very strong assumption: to be called a community, a group of vertices must follow a very strict structural property. In other words, they use the following meta definition of community:

Meta Definition 6 (Structure Community) *A structure community in a complex network is a set of nodes with a precise number of edges between them, distributed in a very precise topology*

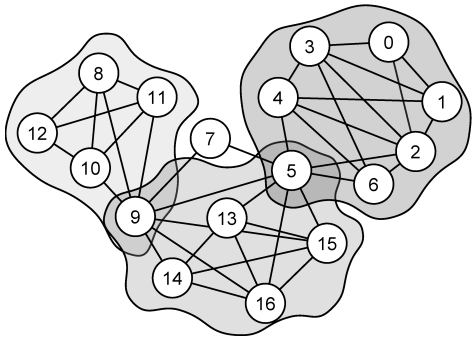


Figure 2.14: The overlapping community structure detected by a clique-percolation approach.

defined by a number of rules. Sets of nodes that do not satisfy these structural rules are not communities.

The aim of the community discovery algorithm is to find all the maximal structures in the network that satisfy the desired constraints. The corresponding meta procedure implemented in this category is simple (i.e. find in a efficient way all the maximal structures defined) and hence there is no need to discuss it further.

This task is similar to a very well-known data mining problem in network analysis: graph mining. Some examples of graph mining algorithms are [268, 38, 204, 157]. However, traditional graph mining algorithms only return all the single different structure patterns with their support. In community discovery there is only one important structure and the desired result is the list of all vertex groups that make up that structure in the network.

We will thus ignore pure graph mining algorithms and just focus on structural community discovery approaches. The methods reviewed here are: clique percolation [209] and its evolution for bipartite graphs [168], the s-plexes detection [155] and a maximal clique approach [237]. We will not focus on other minor evolutions, such as the k-dense approaches [228].

Since a defined structure may be, without any constraint, overlapping, weighted, directed or multidimensional, there is virtually no structural feature that cannot be embedded in a definition used by the algorithms in this category. Depending on the desired structure, analysts can also find communities that do not overlap with any of the previous categories, thus avoiding densities, or bridges or any other previous definition. The downside of this strategy arises when working in an incremental setting: given a simple modification on the structure, such as adding or deleting a single node or edge, the algorithm is likely to recompute everything from scratch. This is because properties of the substructure that are discovered may be violated by any single modification.

K-Cliques [209]

Palla et al. suggest that a community can be interpreted as a union of smaller complete (fully connected) sub-graphs that share nodes. The authors define a k-clique-community as the union of all k-cliques that can be reached from each other through a series of adjacent k-cliques. Two k-cliques are said to be adjacent if they share $k-1$ nodes. A 2-clique is simply an edge and a 2-clique-community is the union of those edges that can be reached from each other through a series of shared nodes. Consider Figure 2.14. In this case the clique percolation approach detects $\{0,1,2,3\}$ as a 4-clique. Then it considers $\{1,2,3,4\}$: it is again a 4-clique and it shares 3 vertices with the previous one. Thus the two cliques are joined in one community. The same is true for the 4-cliques $\{2,3,4,6\}$ and $\{2,4,5,6\}$, thus identifying the community $\{0,1,2,3,4,5,6\}$. In this process, two communities can have an overlap of some vertices (in the example, vertices 5 and 9).

The algorithm first extracts all complete sub-graphs of the network that are not part of a larger complete sub-graph. The aim of the first phase is to populate a clique-clique overlap matrix. In this data structure each row (and column) represents a clique and the matrix elements are equal

to the number of common nodes between the corresponding two cliques. The diagonal entries are equal to the size of the clique. The k-clique-communities can be found by erasing every off-diagonal entry smaller than $k-1$. The complexity of this procedure, since the hardness of clique detection, is $\mathcal{O}(m^{\frac{\ln m}{10}})$.

S-Plexes Enumeration [155]

An s-plex is a relaxed concept of the c-isolated clique [136, 135]. Let $G = (V, E)$ be an undirected graph. A set $S \subseteq V$ of k vertices is called c-isolated if it has less than ck outgoing edges, where an outgoing edge is an edge between a vertex in S and a vertex in $V \backslash S$. A c-isolated clique is a concept that is considered too restrictive for a community. Instead, the authors use a relaxed version of a c-isolated clique called s-plex [24]: in an undirected graph $G = (V, E)$, a vertex subset $S \subseteq V$ of size k is called an s-plex if the minimum degree in $G[S]$ is at least $k - s$. Hence, cliques are exactly 1-plexes.

Since in an s-plex S of size k every vertex $v \in S$ is adjacent to at least $k - s$ vertices, the sub-graph induced by S in the complement graph (the graph with the same set of vertices and complementary edge set) $G[S]$ is a graph with a maximum degree of at most $s - 1$. The idea is to enumerate maximal s-plexes in G by deleting minimal sub-graphs with a maximal degree of $s - 1$ in the complement graph. A key concept for this solution is the pivot set P. The pivot set contains the pivot vertex v and those vertices that belong to the s-plex but are not adjacent to v. The pivot vertex is defined as the vertex with the lowest index of those vertices with less than c outgoing edges.

The algorithm is an evolution of [205] and removes vertices from the candidate set C with too few neighbors in C. It builds the complement graph, then for each possible pivot set P applies the deletion of minimal sub-graph in the complement graph. Finally, it removes enumerated s-plexes that either have pivot $u \neq v$ or are not maximal. The complexity is $\mathcal{O}(knm)$.

Bi-Clique [168]

This is a bipartite graph version that solves various issues regarding the k-clique approach [209], namely the impossibility to analyze sparse network regions, due to the fact that 2-clique communities are simply the connected components of the network. The first non-trivial k-clique has size k = 3 and nodes must have at least two links in order to qualify for participation in a 3-clique. In networks with heavy tailed degree distributions, a large fraction of the nodes have less than two edges.

Bi-clique is a natural approach for affiliation networks, where in a one-mode projection all (sparse) information regarding the bipartite linkages is reduced to a giant quasi-clique. All the information contained in edge weights is typically discarded in a subsequent thresholding operation. The Bi-Clique algorithm detects structures between 2-clique communities and 3-clique communities where the k-clique algorithm usually fails.

The algorithm begins by isolating the N maximal bi-cliques in the bipartite network using [255]. Using this list the authors create two symmetric clique overlap matrices for the two classes of nodes. Then, for both matrix diagonal, elements greater than or equal to a and b (the two parameters of the algorithm) respectively are set to one, while everything else is set to zero. The final overlapping matrix is obtained by the matrix intersection, using the AND operator. The final step is to determine the connected components of L; each component corresponds to a bi-clique community. The final complexity of the approach is $\mathcal{O}(m^2)$.

EAGLE [237]

EAGLE starts from the following assumption: in every dense-linked community there is at least one large clique. This clique could be considered the core of the community. EAGLE firstly finds out all the maximal cliques in the network with the Bron-Kerbosch algorithm [59] (complexity $\mathcal{O}(3^{\frac{n}{3}})$), discarding those whose vertices are part of other larger maximal cliques and those with

less than k vertices. EAGLE then calculates the similarity between each pair of communities. It then selects the pair of communities with the maximum similarity, incorporating them into a new community and calculating the similarity between the new community and other communities. The similarity measure is the modularity [75]. This calculation is repeated until only one community remains, thus completing the dendrogram.

The second stage is to cut the dendrogram. Any cut through the dendrogram produces a cover of the network. To determine the place of the cut, a measurement is required to judge the quality of a cover, computed with a given variant of modularity.

2.3.9 Link Clustering

Some recent approaches have been based on the idea that a community is not a partition of network nodes, but a partition of the links. In other words, it is the relationship between two entities that belongs to a particular environment and the entities belong to all the communities of their edges (or a subset of them).

The meta procedure in this class is:

Meta Procedure 6 *We are given a set of relations M between a set of entities N. We cluster together relations that are similar, i.e. established between the same set of entities, and we then connect each entity n to the communities its relations belong to.*

The underlying meta definition of community is:

Meta Definition 7 (Link Community) *A link community in a complex network is a set of nodes that share a number of relations clustered together since they belong to a particular relational environment.*

This approach implies an overlapping partition, since a node belongs to all the communities of its links, and only in rare occasions do all the links belong to a single community. We provide evidence for this point in the experimental section, by looking at the average number of communities a node belongs to, according to algorithms in this category. One feature that is ignored by this community definition is the direction of a relation, since an undirected link belongs to a single community. There is no way to attach a relationship from u to v to a community and a relationship from v to u to another community, since they both belong to the same relational environment.

The basic approach to the link clustering problem is to define a projection graph in which the nodes represent the links of the original graph and the definition of a proximity value in order to understand how close two edges of the network are. In both cases the critical point is to measure the relations between the edges. A classical clustering algorithm can then be applied.

The methods reviewed here reflect both approaches. The first [89] defines the projection graph with a random walk measure for the proximity of the projected edges, then uses modularity to compute the modules of the network. The second one [12] is a general framework in which it is possible to define any distance measure for the nodes (such as the Jaccard index) and then apply a classical hierarchical clustering technique based on this distance definition. Finally we present also a bayesian approach to this problem [25].

Link modularity [89]

In this work, by defining communities as a partition of the links rather than the set of nodes, the authors interpret the usual modularity Q in terms of a random walker moving on the nodes. They further define two walking strategies: a link-link and a link-node-link random walk. They project the adjacency matrix onto a bipartite incidence matrix. The elements $B_{i\alpha}$ of this $n \times m$ matrix are equal to 1 if link α is related to node i, and 0 otherwise.

The incidence matrix is then projected onto a line graph: a link is added between two nodes in this projected graph if these two nodes have at least one node of the other type in common in the original incidence bipartite graph. Modularity is then computed on this line graph. The total complexity of creating the line graph and computing modularity is $\mathcal{O}(2mk \log n)$.

Hierarchical Link Clustering HLC* [12]

In this approach, the authors start from the assumption that whereas nodes belong to multiple groups (individuals have families, co-workers and friends), links often exist for one dominant reason (two people are in the same family, work together or have common interests) and therefore they cluster them. They define a link similarity measure as the Jaccard coefficient. This measure is computed on the sets of neighbors of each edge sharing one node (i.e. only adjacent edges). The formula used is:

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

where e_{ik} is an edge between nodes i and k and $n_+(i)$ is the set of neighbors of node i. The approach can be used with an arbitrary similarity function for the edges. Furthermore, although weights and multipartite structures are not considered with this formula, the authors claim that it is possible to extend the approach in order to obtain such features.

The authors then build a dendrogram with a classical hierarchical clustering approach using the defined similarity measure, with a time complexity of $\mathcal{O}(n\bar{K}^2)$. In the dendrogram each leaf is a link from the original network and branches represent link communities. In the hierarchical structure identified, links occupy unique communities whereas nodes naturally occupy multiple communities, owing to their links. Thus the extracted network structure is both hierarchical and overlapping. The dendrogram is then cut by optimizing the partition density objective function [93].

Link Maximum Likelihood [25]

In this work the general idea of a link clustering is combined with multidimensional networks: the idea is that communities arise when there are different types of edges, i.e. dimensions, in a network. Basically the approach is to generate a model for the observed network with a given partition of edges into link communities and then testing these communities with a maximum likelihood approach. The generation and test is very similar to the technique implemented in the Expectation Maximization [200] presented in the following category, but in this case is applied on edges instead of applying it on nodes.

2.3.10 No Definition

There are a number of frameworks for community discovery that use a very trivial definition of community or have no definition at all. These methods often assume that there are some desirable features for communities that are not provided by many algorithms. They define preprocessing and/or postprocessing operations and then apply them to a number of other different known methods which do not extract communities with the desired features. In this way they improve the results.

Basically, the meta definition adopted is:

Meta Definition 8 (Community) *Communities in a complex network are sets which present a number of particular features regardless of why their nodes are grouped together.*

Of course, the meta procedures and features of these approaches depend on both the pre/postprocess and the "hosted" method. The works presenting a proper definition of a community are, for instance, the evolutionary clustering [67] or the CONGA algorithm [116], which have already been outlined in this section. Given that we have presented their desired common features for the sets in the form of an independent community definition, we have not included these methods in this category.

Instead we focus on four methods: the first is a hybrid framework combining Bayesian and non-Bayesian approaches [86], the second relies on a custom definition of community given by the analyst and then performs a multidimensional community discovery, by identifying the noisy relations inside the network [61], the third one is a bayesian hierarchical approach [74], finally the last one is based on an expectation maximization principle [200].

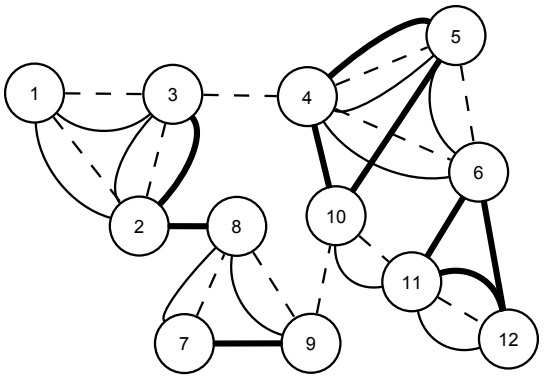


Figure 2.15: A multidimensional network. Solid, dashed and tick lines represent edges in three different dimensions.

Hybrid* [86]

For this framework, the authors start from the point that overlapping communities are a more precise description for the multiplicity of node links compared to non-overlapping approaches. If a node's links cannot be explained by a single membership, then the community discovery problem has to be solved in an overlapping formulation. On the other hand, if a node's links can be explained almost equally well by a number of single and mixed memberships, hard clustering may be simpler. The conclusion is that a combination of an overlapping community discoverer that takes an already hard defined community as input with a non overlapping method should perform better. Thus the HFCD framework is built. It is made up of three parts: the Bayesian core, the hint source procedure and the coalescing strategies.

The Bayesian core is the overlapping community discovery algorithm that collects the hints from the other non overlapping method and outputs the final community partition. In [86] the authors use a Latent Dirichlet Allocation on Graphs [130, 46] as their core method. The Bayesian core needs some hints in order to perform the community discovery procedure. These hints are provided by any other non overlapping community detection algorithm, namely modularity [75] and Cross Associations [68] (here reviewed in its evolution as a Context-specific Cluster Tree [211]).

The most important contribution of this approach is in creating a procedure that solves the problem of how to incorporate the hints into the core model. This is done by the coalescing strategies. The authors propose three different strategies: attributes (each community is an attribute of the node), seeds (the community partition is used as an initial configuration of the second community discovery phase), and prior (a mix of the previous two). In order to make the inference procedure both for attributes and for the initial configuration, the authors use the Gibbs sampling technique [119]. The additional complexity over the used methods is $\mathcal{O}(nk\bar{K})$.

Multi-relational Regression [61]

This algorithm aims to discover hidden multidimensional communities. The authors use the term "relation" for a dimension, i.e. a criterion to connect entities. They define relation networks, group them together and create a kind of social network, calling it a multi-relational social network or heterogeneous social network, another name for a multidimensional or multiplex network. The basic assumption is that each relation (explicit or implicit) plays a different role in different tasks.

For instance consider the multidimensional network in Figure 2.15. The authors suppose that an analyst might want to specify that nodes 8, 9, 10 and 11 belong to the same community. The three dimensions (represented by solid, dashed and thick edges) then have a different importance in reflecting the user information needed. The thick dimension can be considered as noise, and the most important dimension is obviously the dashed dimension. The community discovery process should take this situation into account in order to provide an output close to the information needs

of the user.

The authors thus represent each relation with a weighted matrix. Each element in the matrix reflects the relation strength between the two corresponding entities. This matrix is then mined depending on a user example (or information need): the user submits a query defining the desired community structure. From this structure, the algorithm reconstructs the possible hidden relation, combining the single relation graphs with linear techniques, and then performs the community discovery on the resulting hidden graph.

The hidden relation is tackled as a prediction problem: once the combination coefficients of the desired entities and the desired relations are computed, the hidden relation strength between any object pair can be predicted. This is a regression problem that can be solved with a number of techniques [45]. For a discussion of the issues in this solution based on unconstrained linear regression see [126]. The exact regression used is the Ridge Regression.

Hierarchical Bayes [74]

In this work authors start from the assumption that many real world networks present a hidden hierarchical organization able to explain some of the basic properties of the structure. By reconstructing this latent organization, they are able to group together nodes which are part of the same functional module of the network. It is evident that there is no traditional definition of community at all, and also the authors acknowledge that to reconstruct the hidden dendrogram is a task which goes beyond the simple clustering.

Basically, authors generate and sample a set of dendrograms, which are able to generate a random network with similar features to the observed network, with a Monte Carlo algorithm. The sampling is driven by the maximum likelihood, i.e. the dendrograms are extracted according to how well they can reproduce the observed features. By varying the p_r parameter, the probability to join two vertices in the dendrogram, authors can tune the dendrogram generation in order to fit different properties of the network. Finally, the set of dendrograms is merged into a single consensus dendrogram, which is the best overall representation of the observed network. Although their technique presents an exponential time complexity at the worst case, authors found that in average their complexity should not exceed $\mathcal{O}(n^2)$.

Expectation Maximization [200]

This work acknowledges the basic problem in the community discovery literature, i.e. it is needed firstly to define what a community is and only after it is possible to implement an algorithmic procedure able to create a partition of the network which reflect the best community division according to the starting definition. However, the problem is that sometimes it is hard to define a priori what a community is in a particular network, and failing to do so may end up in finding not significant results. The proposed method is instead able to adapt its definition of community to the most likely present in the data, which may be anyone of the presented classification in this paper.

Basically the authors consider the group membership of each node as an unknown feature. They then define for each vertex i the probability that a (directed) link from a particular vertex in group r connects to vertex i as η_{ri}. Finally, π_r is the probability of belonging to group r. Both η_{ri} and π_r are unknown and depend on each other. With an iterative, self-consistent approach that evaluates both simultaneously, two characteristic equations which define the expectation maximization algorithm are derived, and the problem can be then solved.

2.3.11 Empirical Test

In this section we briefly present an empirical evaluation of some of the presented algorithms. The aim is to strengthen the intuition regarding the desired features which each category is either able to present naturally or has difficulties with.

Algorithm	k	$\bar{n}$	Q	fl	C^{-1}	o
SocDim	12	45.583	N/A	6.583	0.451	2.096
Autopart	6	43.500	0.309	18.500	0.212	1
Modularity	8	32.625	0.724	0.375	0.744	1
Local Density	31	8.419	0.714	0.226	0.549	1
Edge Betweenness	11	23.727	0.738	0.455	0.656	1
CONGA	119	5.277	N/A	3.958	0.076	2.406
Label Propagation	13	20.077	0.735	0.385	0.616	1
Walktrap	12	21.750	0.738	0.250	0.652	1
Infomap	17	15.353	0.721	0.765	0.510	1
K-Clique	16	16.125	N/A	1.562	0.341	0.989
S-Plex	96	3.615	N/A	2.417	0.070	1.330
Link Modularity	37	26.216	N/A	3.730	0.395	3.716
HLC	256	3.734	N/A	2.539	0.063	3.663

Table 2.3: The evaluation measures for the communities extracted with different approaches.

To do this, as our benchmark we use the friendship dimension extracted from the Facebook network presented in Chapter 5. We have depicted this structure in Figure 5.1. As we already know, the friendship dimension contains 261 nodes and 1,722 edges. We chose this network because the human eye can easily spot natural denser areas: there are four main ones at the bottom and left hand side of the picture and three big areas in the upper right hand side, while in the middle there is a sort of gray area and smaller cliques and quasi-cliques of 3-7 nodes float around.

We have tried to include as many algorithms as possible in this section[1]. We excluded reviewed methods for any of the following reasons: we were not able to find any implementation (or working implementation) freely available, the algorithm did not provide better knowledge regarding its category being very similar to another already included, or the algorithm was not suitable for real-world purposes, i.e. it was not able to provide a result on our example network in less than two hours and 1GB of memory occupation (for a 37kB input).

All of the evaluation measures used take a partition P of the network as input, i.e. a list of set of nodes which may or may not have common elements (i.e. overlap).

- **Modularity** (Q). Although there are overlapping definitions for this measure [203], the main version used is the standard one which is not defined for overlapping partitions. Therefore, we computed the original version of Modularity only for non-overlapping results.

- **Flake-ODF** (fl), introduced in [176], is defined as the fraction of nodes in a community that have fewer edges pointing inside than outside of the cluster. We calculate the average over all communities, i.e. $fl(p) = \sum_{k \in P} \frac{|\{u : u \in k, |\{(u,v) : v \in k\}| < deg(u)/2\}|}{|k|}$. In [176] many evaluation measures are presented in order to solve the monotonic increase in modularity (i.e. the resolution problem: bigger clusters tend to score better). However, we tested all of them in our experimental setting (some are not reported here for the sake of readability) and we found that all tend to assign constantly lower scores to overlapping partitions in the same network. Thus, these measures should be refined in order to be more general and to include the very common and popular overlap feature.

- **Reverse Conductance** (C^{-1}). Conductance is also presented in [176] as the fraction of total edge volume that points outside the cluster. We are interested in the reverse concept, i.e. the fraction of total edge volume that points inside the cluster, i.e. $C^{-1} = \frac{1}{|P|} \sum_{k \in P} \frac{m_k}{2c_k + m_k}$, where $m_k = |\{(u,v) \in m : u \in k \wedge v \in k\}|$ and $c_k = |\{(u,v) \in m : u \in k \wedge v \notin k\}|$.

- **Overlap Ratio** (o) is informally defined as the average number of communities that a node belongs to in the network, i.e. $o(p) = \sum_{n \in N} \frac{|\{k \in P : n \in k\}|}{|N|}$. While a non overlapping community discovery usually returns 1 in this metric, if an algorithm does not cluster all the vertices in the network then it may return a value less than 1.

[1] We are thankful to all the authors of the included algorithms for making them available or sending them to us.

We report the final results in Table 2.3, in which we have one row per algorithm and one column per measure. We added some simple statistics about the partitions, such as the number of communities and average number of nodes per community. For the measures, in Table 2.3 we use the same notation used in this section to present them.

We are now able to provide an additional reason for our classification by analyzing the presented results.

SocDim and Autopart belong to the Feature Distance category. As discussed previously, in this category we have a method with basically any feature (for example, SocDim is multidimensional and overlapping, while Autopart is parameter free and allows directed edges). The downside is the counter intuitive partition according to the graph topology. It is easy to see, in fact, how poorly Autopart scores in the Modularity test (Q). However, since we did not compute Modularity for the overlapping SocDim partition, we also used the Flake-ODF measure (fl). In this case too, both SocDim and Autopart got higher values, i.e. it is more frequent that a node has more edges pointing outside the cluster than pointing in. Overlap partitions usually have the lowest performance according to Flake-ODF, and to Conductance, since nodes in the overlap zone are densely connected to two or more clusters. However Autopart is not an overlapping method and SocDim turned out to be the worst of the other overlapping algorithms according to this evaluation.

For the Internal Density category we tested Modularity and Local Density algorithms. Their edge volume inside the community (Reverse Conductance C^{-1}) is high. For Modularity edge volume was the highest score, while Local Density scored well, although it did not come second for implementation reasons (the algorithm returns some communities with only one vertex which obviously contributes with zero to the sum).

As stated in the paragraph regarding the bridge detection community discovery, no assumptions about the density of the clusters are made. Thus these algorithms may have a high score on the inverse conductance (Edge Betweenness), or may not (CONGA).

Unfortunately our set of algorithms for the Diffusion category is very narrow and no conclusions can be drawn. Instead, Closeness algorithms Walktrap and Infomap highlight their independence from a simple density definition: Walktrap favors a few bigger (and denser) communities, while Infomap focuses on smaller and lower level sparser ones.

There is one clear downside to the Structure definition category: the K-Clique algorithm has an overlap ratio o less than one, since its structure definition is very strict and many nodes cannot satisfy it, ending up in no community.

Finally, algorithms in the Link Community category gave a very high overlap score (o). This proves that clustering edges is a natural and automatic way to get highly overlapping partitions.

2.3.12 Alternative Classifications

Over the last decade, several reviews of community discovery methods have been published. We would consider the most important to be [201, 66, 82, 95, 216, 231].

Fortunato and Castellano [95], hugely extended by Fortunato in [94], have published the most recent and probably the most comprehensive review on the community discovery problem. To tackle the problem they consider various definitions of community (local, global and vertex similarity), features of communities for extraction, and different categories. The number of algorithms and references they considered is impressive. We believe that a new review of this topic is needed because the authors analyze the main techniques of each method for community detection, however they do not build an organization of community definitions (while acknowledging that different ones exist). Their work does not include some more advanced features and definitions of community found in the literature, such as multidimensionality or an influence spread formulation of the problem.

Porter et al. [216] and Schaeffer [231] have also recently reviewed community discovery methods. In [231] they also introduced the problem of a comprehensive meta definition of community in a graph. Again, however, although they begin to provide different definitions of community, they do not create a classification of the community discovery algorithm based on such a community.

In Newman's pioneering work [201] we can find an organization of historical approaches to community discovery in complex networks following their traditional fields of application. Newman presents the most important classical approaches in computer science and sociology, enumerating algorithms such as spectral bisection [217] or hierarchical clustering [235]. He then reviews new physical approaches to the community discovery problem, including the known edge betweenness [106] and modularity [199]. His paper is very useful for a historical perspective, however it records few works and obviously does not take into account all the algorithms and categories of methods that have been developed since it was published.

Chakrabarti and Faloutsos [66] give a complete survey of many aspects of graph mining. One important chapter discusses community detection concepts, techniques and tools. The authors introduce the basic concepts of the classical notion of community structure based on edge density, along with other key concepts such as transitivity, edge betweenness and resilience. However, this survey is not explicitly devoted to the community discovery problem. It describes existing methods but does not investigate the possibility of different definitions of community or of a more complex analysis.

Danon et al. [82] test an impressive number of different community discovery algorithms. They compare the time complexity and performances of the methods considered. Furthermore, they define a heuristic to evaluate the results of each algorithm and also compare their performance. However, they focus more on a practical comparison of the methods, rather than a true classification, both in terms of a community definition and in the feature considered for the input network.

Various authors have also proposed a benchmark graph, which would be useful to test community discovery algorithms [164].

As we have seen, community discovery is a very complex task involving an incredible amount of techniques and desired features. To define and predict what will be the most important features in the future is an open question. As witnessed by this thesis and by many other publications [246, 36, 192, 25, 61] there is a growing interest in multidimensionality, perceived as a feature that is part of the solution and not only as an input to be preprocessed. In other words, we want not only to consider multidimensionality as an input, but also to extract truly multidimensional communities. But how to define what exactly a "multidimensional community" is? Are all the groups of nodes with dense multidimensional connections equal? We address these questions in the following section.

2.4 Generators

In this section we present the most common and basics generators of synthetic graphs. The general aim of the network modeling is to capture essential properties behind real-world phenomena, with simple assumptions over the mechanisms generating them (just like the rich-get-richer effect explanation for generating power law degree distributions). With these generators it is also possible to obtain a network respecting some of the properties of real world ones, presented in Section 2.2. The two different aims can cluster network generators into two different main categories: descriptive and generative models.

As we see, in the descriptive category generally each model focuses onto one or few essential properties of real world complex networks, trying to unveil knowledge about how that property came to life. None of these models has the aim of respecting all the properties of all the possible variants of real world networks. In the generative category, the models are more focused on providing as output data more similar to the actual networks, but the creation of a synthetic network that is able to represent all the rich semantic of real world data is still an open problem.

We now provide more details about each category in the two following subsections. Please note that an orthogonal categorization, namely one that divides models into static and dynamic, can be applied to models belonging either to the descriptive or to the generative one.

2.4.1 Descriptive Models

Solomonoff and Rapoport [239] and independently Erdös and Rényi [87] proposed the following extremely simple model of a network. Take some number n of vertices and connect each pair (or not) with probability p (or $1-p$). A slight variation of the same model is the following: consider all possible node pairs (that are $\frac{n(n-1)}{2}$), then choose randomly m node pairs. An equivalent definition: list all graphs with exactly n nodes and m edges and choose one randomly.

The random graph is the mathematically most well-studied and understood model. The random graph, while illuminating, is inadequate to describe some important properties of real-world networks. In fact, in a random graph all edges are equally probable and appear independently. This assumption is not true in almost all the networks in real world.

The structure of the random graph dramatically change by considering different values of the p and/or m parameter. The usual regime of interest is $p \sim \frac{1}{n}$ ($m \sim n$), when n is large. When $p > \frac{1}{n}$ the average degree z is greater than one and in the network there is a transition phase. A giant component, see Section 2.2, appears and the expected diameter of the network is equal to $\frac{\log n}{\log z}$ [195].

However in almost all other respects, the properties of the random graph do not match those of networks in the real world. It has a low clustering coefficient: the probability of connection of two vertices is p regardless whether they have a common neighbor or not [262]. As a consequence, a random graph does not show a community structure. The model also has a Poisson degree distribution, quite unlike the power law degree distributions present in real world networks.

Random graphs can be extended in a variety of ways to make them more realistic. The property of real graphs that is simplest to incorporate is the property of non-Poisson degree distributions, which leads us to the so-called configuration model [189]. In order to obtain a graph with the configuration model, we specify a degree distribution p_k, such that p_k is the fraction of vertices in the network having degree k or higher. We choose a degree sequence, which is a set of n values of the degrees k_i of vertices $i = 1...n$, from this distribution. We can think of this as giving each vertex i in our graph k_i "stubs" or "spokes" sticking out of it, which are the ends of edges-to-be. Then we choose pairs of stubs at random from the network and connect them together. In order to obtain a real power law degree distribution, one needs only to specify the correct degree sequence.

With this model is possible to generate scale free networks, although we do not know how this distribution has been generated. We know that the probability of attaching a new edge to a vertex is proportional to the vertex actual number of "stubs", but this is not sufficient in order to understand the power law degree distribution in real world networks. Further, the clustering coefficient of this model is still very low.

It is possible to define slight variants of the configuration model, that take into account also other characteristics such as the direction of the edges, the degree correlations or anticorrelations and the high clustering coefficient. An important class of these models are the so called exponential random graphs or Markov graphs [98]. In Markov graphs the presence or absence of an edge between two vertices in the graph is correlated only with those edges that share one of the same two vertices. Edge pairs that are disjoint (have no vertices in common) are uncorrelated. This model has a tendency to condense, i.e. it creates regions of the graph that are essentially complete cliques. Networks in the real world, however, do not seem to have this sort of "clumpy" transitivity: regions of cliquishness contributing heavily to the clustering coefficient, separated by other regions with few triangles.

However, all these variants share the common problem that are defined in order to capture one single real world network property. In order to obtain a greater accuracy, different graph generation criteria are needed.

Networks may have a geographical component attached to them; i.e. the vertices of the network have positions in space. In many cases it is reasonable to assume that geographical proximity will play a role in deciding which vertices are connected to which others. The small-world model [262] starts from this idea by positing a network built on a low-dimensional regular lattice and then adding or moving edges to create a low density of "shortcuts" that join remote parts of the lattice to one another.

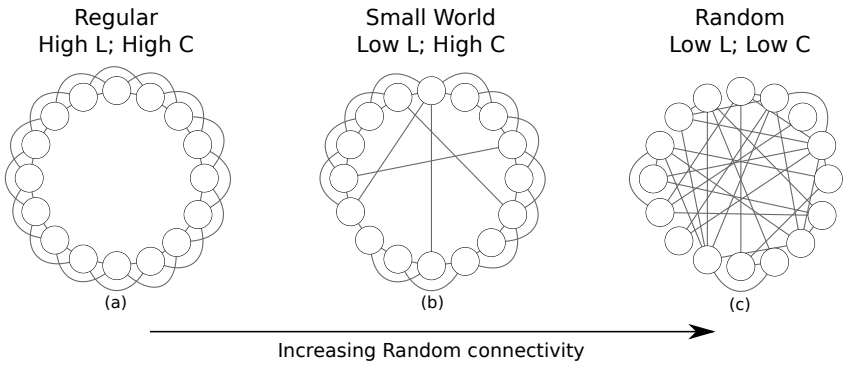


Figure 2.16: Three graphs generated with the small-world model: (a) $p = 0$; (b) $p = \frac{1}{4}$; (c) $p = 1$.

Small-world models can be built on lattices of any dimension or topology, but the best studied case so far is one-dimensional one. If we take a one-dimensional lattice of n vertices with periodic boundary conditions, i.e. a ring, and join each vertex to its (k or fewer) closest neighbors in the lattice, we get a system with nk edges. This system is depicted in Figure 2.16(a). The small-world model is then created by taking a small fraction of the edges in this graph and rewiring them. The rewiring procedure involves going through each edge in turn and, with probability p, moving one end of that edge to a new location chosen uniformly at random from the lattice, except that no double edges or self-edges are ever created. The output network of this procedure is represented in Figure 2.16(b).

The rewiring process allows the small-world model to interpolate between a regular lattice and something which is similar, though not identical, to a random graph. When $p = 0$, we have a regular lattice. It has been shown that the clustering coefficient of this regular lattice is $C = \frac{3k-3}{4k-2}$, which tends to $\frac{3}{4}$ for large k. The regular lattice, however, does not show the small-world effect. The average geodesic distance between vertices tend to $\frac{n}{4k}$ for large n. When $p = 1$, every edge is rewired to a new random location and the graph is almost a random graph, with typical geodesic distances on the order of $\frac{\log n}{\log k}$, but very low clustering $C = \frac{2k}{n}$. As Watts and Strogatz showed by numerical simulation, however, there exists a sizable region in between these two extremes for which the model has both low path lengths and high transitivity, as shown in Figure 2.16.

The model can be simplified considerably by rewiring both ends of each chosen edge, and by allowing both double and self edges. In another variant no edges are rewired. Instead "shortcuts" joining randomly chosen vertex pairs are added to the low-dimensional lattice. The parameter p governing the density of these shortcuts is defined so as to make it as similar as possible to the parameter p in the first version of the model: p is defined as the probability per edge of being randomly rewired to create a shortcut.

All the models discussed so far take observed properties of real-world networks, such as degree sequences or transitivity, and attempt to create networks that incorporate those properties. The models do not however help us to understand how networks come to have those properties in the first place. In this section we examine a class of models whose primary goal is to explain network properties. In these models, the networks typically grow by the gradual addition of vertices and edges in some manner intended to reflect growth processes that might be taking place on the real networks, and are these growth processes that lead to the characteristic structural features of the network.

A number of authors have studied models of network transitivity that make use of "triadic closure" processes, such as [257]. In these models, edges are added to the network preferentially between pairs of vertices that have another third vertex as a common neighbor. In other words, edges are added so as to complete triangles.

The best studied class of network growth models by far is the class of models aimed at explaining the origin of the power law degree distribution. The first archetypal model was the cumulative advantage model [218], defined in 1965 by Price. His work was built on ideas developed in the

1950s by Herbert Simon, who showed that power laws arise when "the rich get richer", when the amount you get goes up with the amount you already have. Price appears to have been the first to discuss cumulative advantage specifically in the context of networks, and in particular in the context of the network of citations between papers. His idea was that the rate at which a paper gets new citations should be proportional to the number that it already has.

This idea is now widely accepted as the probable explanation for the power-law degree distribution observed not only in citation networks but in a wide variety of other networks too, including the World Wide Web, collaboration networks, and so on. The idea of cumulative advantage is at the basis of the preferential attachment model [30] of Barabasi and Albert. In this model we have vertices that are added to the network with degree m, which is never changed thereafter. The other end of each edge is then being attached to another vertex with probability proportional to the degree of that end vertex. There are many different variants of this model that are defined in order to capture some differences in the event to be represented: for example this undirected graph can be considered directed when one needs to model the structure of the World Wide Web. However, all these variants do not alter the basic properties of this model.

The preferential attachment model is able to create networks with a power law degree distribution. In particular, its power law exponent α is fixed to 3, that is a good approximation for most real world networks [50].

There are other interesting properties of the model that has been studied so far. Some of them highlight issues of inaccuracy in the real world network representation of this model. First, the model has two important types of correlations. There is a correlation between the age of vertices and their degrees, with older vertices having higher mean degree. In other words the earliest vertices added have substantially higher expected degree than those added later, and the overall power-law degree distribution of the whole graph is a result primarily of the influence of these earliest vertices. This correlation between degree and age has been used to argue against the model as a model of the World Wide Web: it has been shown, using actual Web data, that there is no such correlation in the real Web [3]. The second correlation has been discovered between the degrees of adjacent vertices.

2.4.2 Generative Models

The study of temporal evolving networks has given new life blood also to a related problem: the definition of novel generator tools of synthetic complex networks, a well-known problem in literature even in the first years of graph theory, as presented in the previous section [30]. However, as we presented, the preferential attachment model focuses more on a description about how we obtain a power law degree distribution, rather than on the generation of a synthetic real world complex network. This step is tackled by models clustered in this category, and it relies mainly on making some hypothesis about what kind of microscopic node behavior would reproduce the observed macroscopic network structure.

Recently, it has been proposed a new way of looking at the evolution models. In this new approach the model does not try to represent the characteristics of the network as a whole, instead the focus is devoted to the microscopic level, i.e. what is interesting is the behavior of every node taken individually. The focus is to study the individual node arrival and edge creation processes that collectively lead to macroscopic properties of networks. Instead of only focusing on the global network structure and then hypothesizing about what kind of microscopic node behavior would reproduce the observed macroscopic network structure, works of this kind focus directly on the microscopic node behavior per se.

The main work in this class is based on the maximum-likelihood estimation (MLE) principle [172], that can be applied to compare a family of parameterized models in terms of their likelihood of generating the observed data, and as a result, pick the "best" model (and parameters) to explain the data. To apply the likelihood principle, authors consider the following setting: they evolve the network edge by edge and for every edge that arrives into the network they measure the networks likelihood that the particular edge endpoints would be chosen under some model. The product of these likelihoods over all edges will give the likelihood of the model. The mechanism of node

and edge arrivals considered are a given node and edge arrival process and the edge destination selection process. This microscopic evolution approach can outperform the preferential attachment in generating synthetic networks that are a better representation of the real world networks.

However, also in this case it is important to notice that these generators are defined only for a monodimensional dynamic setting. No consideration of the different dimensions in which a link can appear are represented in the model.

2.5 Link Analysis

Link analysis refers to a complex of techniques that explicitly consider the links when building predictive or descriptive models of the linked data. Commonly addressed link mining tasks include object ranking, collective classification, entity resolution and link prediction. The main reference present in literature about link analysis methods is [103].

Perhaps the most well known link analysis task is that of **link-based object ranking** (LBR), which is a primary focus of the link analysis community. The objective of LBR is to exploit the link structure of a graph to order or prioritize the set of objects within the graph. Much of this research focuses on graphs with a single object type and a single link type.

In the context of web information retrieval, the PageRank [208] and HITS [151] algorithms are the most notable approaches to LBR. **PageRank** models web surfing as a random walk where the surfer randomly selects and follows links and occasionally jumps to a new web page to start another traversal of the link structure. The rank of a given web page in this context is the fraction of time that the random web surfer would spend at the page if the random process were iterated ad infinitum. This can be determined by computing the steady-state distribution of the random process. **HITS** implements a slightly more complex process, modeling the web as being composed of two types of web pages: hubs and authorities. Hubs are web pages that link to many authoritative pages. Authorities are web pages that are linked by many hubs. Each page in the web is assigned hub and authority scores. These scores are computed by an iterative algorithm that updates the scores of a page based on the scores of pages in its immediate neighborhood. This approach bears a relation to PageRank with two separate random walks, one with hub transitions and one with authority transitions, on a corresponding bipartite graph of hubs and authorities. The hub and authority scores are the steady-state distributions of the respective random processes. While PageRank is currently used in real world applications, there are no information, to the best of our knowledge, about real world scalable implementation of HITS.

In the domain of social network analysis, LBR is a core analysis task, i.e. finding the most important (central) vertices inside the network. They range in complexity from local measures such as degree centrality, which is simply the vertex degree, to global measures such as eigenvector/power centrality, which use spectral methods to characterize the importance of individuals based on their connectedness to other important individuals [103].

Ranking objects in dynamic graphs that capture event data such as email, telephone calls, or publications introduces new challenges. In contrast to ranking methods for static settings that produce a single rank, the goal is to track the changes in object rank over time as new events unfold. Static ranking methods can be applied to aggregated event data over various time intervals, but this aggregation removes the time ordering of events, and the sparse link structure over a given time interval limits the utility of the resulting ranks.

In the **link-based object classification** (LBC) problem, a data graph $G = (O, L)$ is composed of a set objects O connected to each other via a set of links L. The task is to label the members of O from a finite set of categorical values. The discerning feature of LBC that makes it different from traditional classification is that, in many cases, the labels of related objects tend to be correlated. The challenge is to design algorithms for collective classification that exploit such correlations and jointly infer the categorical values associated with the objects in the graph. In addition to the machine learning community [181], the computer vision and natural language communities have also studied the LBC problem [159], thus highlighting the truly interdisciplinary nature of these fields of research.

The final object-centric task is **entity resolution**, which involves identifying the set of objects in a domain. The goal of entity resolution is to determine which references in the data refer to the same real-world entity. Examples of this problem arise in databases [141] (deduplication, data integration), natural language processing [44] (co-reference resolution, object consolidation), personal information management, and other fields. Entity resolution has been viewed as a pair-wise resolution problem, where each pair of references is independently resolved as being co-referent or otherwise, depending on the similarity of their attributes. Recently, there has been significant interest in the use of links for improved entity resolution. The central idea is to consider, in addition to the attributes of the references to be resolved, the other references to which these are linked. These links may be, for example, co-author links between author references in bibliographic data, hierarchical links between spatial references in geo-spatial data, or co-occurrence links between name references in natural language documents. However, while these approaches consider links for entity resolution, only the attributes of linked references are considered and different resolution decisions are still taken independently. In contrast, collective entity resolution approaches have also been proposed in databases [141], where one resolution decision affects another if they are linked.

Link prediction is an edge-oriented task and it is defined as the problem of predicting the existence of a link between two entities, based on attributes of the objects and other observed links. Examples include predicting links among actors in social networks, such as predicting friendships; predicting the participation of actors in events, such as email, telephone calls and co-authorship; and so on. This problem is often viewed as a simple binary classification problem: for any two potentially linked objects o_i and o_j, predict whether l_{ij} is 1 or 0. One approach is to make this prediction entirely based on structural properties of the network. Liben-Nowell and Kleinberg [207] present a survey of predictors based on different graph proximity measures. Other approaches make use of attribute information for link prediction. Just as in the community discovery research track, also in this case there are very recent works that use a tensor-based approach for link prediction [1]. A latter class of approaches is composed by systems of probabilistic inference about the links. This allows them to capture the correlations among the links. They can also be used for other tasks, such as link-based classification. Ideally this makes for more accurate predictions. However, model-based probabilistic approaches have a computational price: exact inference is generally intractable, so approximate inference techniques are necessary.

In this brief resume of the link analysis state of the art, we have described each link mining task in isolation. More generally, component link mining algorithms may be part of a larger knowledge discovery process [103]. As we move from one domain to another, the processing requirements will change, but the need to compose the algorithms in a unified process will remain. Ideally, as we move from data conditioning to more complex inference tasks, we would like to propagate uncertainty throughout the process. One approach that solves this problem, in theory, is to define a full probabilistic model. However, this approach is not always desirable or feasible. Moreover, many link analysis algorithms are defined in a static environment. But when considering the overall knowledge discovery process, it is important to keep in mind that many aspects of the process are intrinsically defined in an evolving setting.

2.6 Information Propagation

One of the most important event that we can represent with a network is the contagion model. Many researcher have worked defining different possible contagion dynamics that can take place in a network. The general model states that given a node n in state A, then all its neighbors directly connected with an edge to n have an increased probability of turning into state A. The different dynamics through this change of state may happen are modeled in different ways. We report for example the SIR/SIS model and the threshold models in social sciences.

We now focus on a particular type of contagion in complex networks, i.e. when what is spreading on a network is not a disease, but information. This particular problem definition, to understand the dynamics of the information propagation in a social network, is a problem particularly popular

in the data mining literature. We consider a social network in which users can view the action performed by their neighbors. Aim of the information propagation works is to investigate how influence (for performing certain actions) propagates from users to their network friends, potentially recursively, thus identifying a group of users that behaves homogeneously (i.e. a tribe, or a community). These works can be focused basically on two aspects of the information propagation: the temporal dimension intrinsically contained in the flow of information and the causes of the information propagation.

The works of the first class are focused on the topic and the general characteristic of the information propagation in different environments. Basically their aim is to answer the following questions: why certain topics are spread faster than others? What is the distribution of the temporal intervals among the "hops" that the information passes through? In this class can be clustered works like applications of graph mining [66] and temporally annotated sequences [105] techniques that aim to discover frequent patterns (both in graphs and in sequences that represents flows of communication) and then obtain a rich description of the information propagation process. Another work analyzes the efficiency of a recommendation system, trying to define a model of the user behavior [171]. The authors identify the recommendation system as a complex interaction of cascade behavior: one user influences, and is influenced by, a set of neighbors. Then they first analyze the probability of purchasing as one gets more and more recommendations. Next, they measure recommendation effectiveness as two people exchange more and more recommendations. Lastly, they observe the recommendation network from the perspective of the sender of the recommendation. The aim is to answer to the question: does a node that makes more recommendations also influence more purchases? Then authors present a model which characterizes product categories for which recommendations are more likely to be accepted. They use a regression of the product attributes to correlate them with recommendation success.

In the second class the focus is moved from the information propagation to the actors of this event. Aim is to identify in a social network the characteristics and the attribute that make a user a sort of power user, i.e. a leader in the information propagation. Here the focus is to answer these other questions: why certain discussions are passed over while others stop in two hops? What are the characteristics of the nodes that pass the information? The aim can be summarized as leader detection. An example of this kind of works can be represented by [113]. In this work, a node u can be defined as a leader if u performed a and within a chosen time bound after u performed a, a sufficient number of other users performed a. Furthermore these other users must be reachable from u thus capturing the role social ties may have played. A stronger notion of leadership might be based on requiring that w.r.t. each of a class of actions of interest, the set of influenced users must be the same: this will identify a tribe leader, meaning the user leads a fixed set of users (tribe) w.r.t. a set of actions. This variants can be viewed as an alternative definition of community discovery, as explored in Section 2.3. In some works the input includes just the network representation (which is not edge-weighted) [145], in other works this is enriched with an action table which plays a central role in the definition of leaders [113]. Another work that is included in this class aims to represent the information spread as an heat diffusion process [182]. In this way it is possible to model not only the spread of positive opinions about a specific product (or fact), but also the negative influence. Once modeled the information spread, it is possible to apply a marketing candidate selection that identify the characteristics of the user inside the network that can maximize the positive heat diffusion.

Both these classes of works are intrinsically focused on the temporal evolution of their data. In order to register the information spread and/or to identify leaders from a set of timestamped actions, the temporal analysis is a mandatory step. However, in literature the impression is that our knowledge about this phenomenon is still far from complete. Further, very little has been done in another crucial part of the information propagation analysis. It is a matter of common experience, in fact, that the form, the content, the lasting and all the characteristics of an information exchange are dramatically different when we communicate through different media. A phone call is different from an email, from a paper letter, from a chat via an instant messaging software. Thus, it is crucial to record this information and include it in the data representation. And this consideration holds also for contagion models: a virus can be airborne, waterborne or spread through different

media, and the underlying topology of the airborne networks are dramatically different from the waterborne ones. But the works focused on the multidimensional information spread are, to the best of our knowledge, very few.

2.7 Graph Mining

Given a complex network as a basic structure, traditionally a graph with labels attached both to vertices and to edges, aim of a graph mining algorithm is to find in this structure frequent subgraphs. An interesting and complete survey on graph mining tools and techniques can be found in [66]. Here we refer to the narrower definition of graph mining, namely the discovery of frequent patterns in the simple graph structure, i.e. nodes with the same labels connected by edges with the same label. More broadly, graph mining aims to find patterns of all kinds including, but not limited to, path lengths, communities, time evolution patterns, etc.

A subgraph p is said to be frequent in a graph database g if its support $\sigma_\wedge(p, g)$ is greater than or equal to a given threshold [268]. There are different definitions of support, depending on the specific format of data input. However, the definition of support should have three properties: anti-monotone, easy to compute and intuitive [56]. An anti-monotone support definition means that if a set cannot pass the frequency test, being found not frequent, all of its super-set will fail the same test as well. In our case this means that if a subgraph does not pass the threshold test then none of its extension will never pass the threshold test, thus dramatically narrowing the search space. It should be easy to compute because an $\mathcal{NP}$-hard support definition is a clear source of inefficiency. Finally, an intuitive support definition is necessary to understand clearly what the graph miner is actually computing.

The problem of frequent pattern discovery in graph databases can be split in two very different scenarios, namely the graph-transaction and the large single-graph setting [137]. In graph-transaction setting the graph database to be mined is a set of relatively small graphs, and the task of the mining process is to find frequently recurring graphs in this graph database. In single-graph setting the input of the mining system is one single graph with large number of nodes, and the task is to find frequent recurring subgraphs of the single input graph.

Thus in the **graph-transaction** setting the support of a subgraph is defined as the number of the graphs in the input database that contains at least one occurrence of the starting subgraph. The number of actual occurrences in the graph database is not considered. It is easy to note that the frequency is anti-monotone: when we extend our subgraph some of the structures in the database that contained the smallest subgraph may be no longer counted into the support of the new pattern. The kernel of frequent subgraph mining is the subgraph isomorphism test, i.e. the mechanism that allow to verify if two graphs are different instances of the same structure. Lots of well-known pair-wise isomorphism testing algorithms were developed.

These algorithms can be roughly divided by looking at their strategies in exploring the graph structure: they can implement a **breadth first search** [202] (quick, but expensive in terms of memory) or a **depth first search** [268] (slower, but requires a small amount of memory). Then each algorithm uses some heuristics to further cut the search space. One of the most important algorithms is gSpan [268], that represents the basic kernel also for techniques in other settings, such as link prediction. gSpan has introduced two concepts: the DFS lexicographic order and the minimum DFS code, that form a canonical labeling system to support DFS search. When exploring a graph with a depth-first approach, there might be different alternatives. The DFS lexicographic order is a technique to assign an order to all these alternatives, from the "smallest" to the "largest". The minimum DFS code of p is the representation of the smallest depth-first exploration of p. For any p there is one and only one minimum DFS code. Therefore we can solve the graph isomorphism test (i.e. decide if two graphs are the same structure) by confronting the minimum DFS code of two different graph patterns.

By exploiting the DFS lexicographic order, gSpan discovers all the frequent subgraphs without candidate generation and false positives pruning, needed, for example, in an Apriori approach. It combines the growing and checking of frequent subgraphs into one procedure, thus accelerating

the mining process. Currently some researchers are still looking for better heuristics in order to achieve faster implementations of a graph miner [16].

In the **single-graph** setting the frequency, or the number of occurrence, of a subgraph is not a good support function. This definition of support have the problem that it is not anti-monotone [56]; thus it cannot be used effectively in pattern mining, as anti-monotonicity is required to prune the search space. Anti-monotone support definitions currently accepted in literature are based on computing maximum independent sets in overlap graphs, or the minimum image based support. The second one is defined as follows, given a subgraph p that appears in our single input graph g, its support $\sigma_\wedge(p, g)$ is defined as follows: $\sigma_\wedge(p, g) = \min_{v \in V_p} |\{\varphi_i(v) : \varphi_i \textit{ is an occurrence of } p \textit{ in } g\}|$, where v is a vertex in the set V_p of vertices inside the pattern p [56]. Unfortunately this support definition is highly counter-intuitive.

Lately, it has been developed a further evolution in the single graph mining setting. In this evolution the labels attached to the edges represent the time step in which the edge has been created. From this setting it is possible to derive graph-evolution rules from frequent patterns [38]. In this work authors can provide not only frequent patterns with an absolute timestamp on the edges, but patterns in which the edges are labeled with a relative timestamp.

2.8 Privacy

Digital traces of human social interactions can now be found in a wide variety of on-line settings, and this has made them rich sources of data for large-scale studies of social networks. While a number of these on-line data sources are based on publicly crawlable blogging and social networking sites, where users have explicitly chosen to publish their links to others, many of the most promising opportunities for the study of social networks are emerging from data on domains where users have strong expectations of privacy, including e-mail and messaging networks, as well as the link structure of closed (i.e. members-only) on-line communities.

In designing studies of such systems, one needs to set up the data to protect the privacy of individual users while preserving the global network properties. This is typically done through anonymization, a simple procedure in which each individual's "name" (e.g., e-mail address, phone number, or actual name) is replaced by a random user ID, but the connections between the (now anonymized) people, encoding who spoke together on the phone, who corresponded with whom, or who instant-messaged whom, are revealed. The motivation behind anonymizing is roughly as follows: while the social network labeled with actual names is sensitive and cannot be released, there may be considerable value in allowing researchers to study its structure.

But anonymous social network data almost never exists in the absence of outside context, and an adversary can potentially combine this knowledge with the observed structure to begin compromising privacy, de-anonymizing nodes and even learning the edge relations between explicitly named (de-anonymized) individuals in the system. Moreover, such an adversary may in fact be a user (or set of users) of the system that is being anonymized.

In literature there are some papers focused on defining the type of attacks to the privacy of the users represented in a social network, such as [19], [194] and [206]. In short, these attacks are active (walk-based and cut-based) and passive attacks.

The structure of the active attack is roughly as follows. Before the anonymized graph is produced, the attacker creates k new user accounts (for a small parameter k), and it links them together to create a subgraph H. It then uses these accounts to create links (e.g. by sending messages or creating address book entries) to nodes in $\{w_1, ..., w_b\}$, and potentially other nodes as well. Now, when the anonymized copy of G is released, this subgraph H will be still present. The attacker finds the copy of H that it planted in G, and from this it locates $w_1, ..., w_b$. Having identified the true location of these targeted users in G, the attacker can then determine all the edges among them, thereby compromising privacy.

The passive attack is based on the observation that most nodes in real social network data already belong to a small uniquely identifiable subgraph. Hence, if a user u is able to collude with a coalition of $k - 1$ friends after the release of the network, he or she will be able to identify

additional nodes that are connected to this coalition, and thereby learn the edge relations among them.

The results of the attack survey in literature is that one cannot rely on anonymization to ensure individual privacy in social network data, in the presence of parties who may be trying to compromise this privacy. Still are missing some mathematically rigorous implementations of system that can ensure some robust countermeasures to one or more of these attacks.

Chapter 3

Multidimensional Network: Model Definition

In this chapter we briefly provide a formal definition of the concept of Multidimensional Network. We use this model for the remainder of the thesis.

The term *network* refers to the informal concept describing a structure composed of a set of elements and connections or interactions between them. In a real world network the entities may be connected by relations of different nature: for example, two persons may be linked because they are friends, colleagues, relatives or because they communicate to each other by phone, email, and so on. A network where a pair of entities may be linked by different kinds of links, having more than one connection between the two entites, is called a *multidimensional network*. We consider each possible type of relation between two entities a particular *dimension* of the network.

Often, the *graph* is used to model a network with its properties. In the graph model, the entities are represented by nodes while a relation is modeled by a (directed or undirected) edge. In the case of a multidimensional setting a convenient way to model a network is hence 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 any regular labeled graph, also labeled multigraph may be directed and undirected, thus we allow edges to be both directed and undirected, when the analytic aim requires it. However, in our context we do not consider node labels, thus we adopt a particular version of multigraph where only the edges are labeled. The model that we use in the remainder of the paper is hence the *edge-labeled multigraph*. We reserve the possibility of using in this thesis, where specified, multigraphs with labels attached also to nodes, or any other extension of the basic model here presented, to be more general. Formally, 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 triple 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). Moreover, as before reported, we are working with an directed graph, therefore the edges (u, v, d) and (v, u, d) are considered distinct, except when otherwise specified.

Thus, given $|D| = m$ each pair of nodes in G can be connected by at most m possible edges. In fact, it is useless to connect with two distinct edges in the same dimension, as they represent the same relation. When needed, however, we can allow weights to handle the particular situation where two or more relations of the same type are expressed in the data. Therefore, in those cases the edges are no more triplets, but quadruplets (u, v, d, w), where w is any real number representing the weight of the relation between nodes $u, v \in V$ and labeled with $d \in D$.

In the following, we denote by $\mathcal{P}(D)$ the power set of the label collection D and by $\mathcal{G}$ the set of graphs of the form $G = (V, E, D)$.

When we use edge-labeled multigraphs to model a multidimensional network, the set of nodes represents the set of entities or actors in the networks, the edges represent the interactions and relations between them and the edge labels describe the nature of the relations, i.e., the dimensions of the network. Given the strong correlation between labels and dimensions, in the following we use the term *dimension* in order to indicate *label*. Moreover, we denote by χ_E the characteristic function of E, which equals to 1 if a given edge (u, v, d) belongs to E, 0 otherwise. We also say that a node *belongs to* or *appears in* a given dimension d if it has at least one edge labeled with d. So, we define an operator $dim(v, d) : V \times D \to \{0, 1\}$ which equals to 1 if the node v appears in dimension d, 0 otherwise. Given a node $v \in V$, n_v is the number of dimensions in which v appears, i.e. $n_v = \mid \{d \in D \ s.t. \ dim(v, d) = 1\} \mid = \sum_{d \in D} dim(v, d)$. Similarly, given a pair of nodes $u, v \in V$, n_{uv} is the number of dimensions which label the edges between u and v, i.e., $n_{uv} = \mid \{d \in D \ s.t. \ \chi_E(u, v, d) = 1\} \mid = \sum_{d \in D} \chi_E(u, v, d)$.

Chapter 4

Related Work

In this chapter we present a subset of the state of the art of complex network analysis. In particular, here we focus on the study and analysis of networks with multiple kind of interactions in the broadest sense possible. We want to cluster all alternative models developed to express the complex interplay of multiple relations in the real world. We want to understand what are the common features of all these different models with our chosen idea of multidimensional network, also highlighting if and for which cases a mapping of these models cannot be done. We are able to divide the current examples in roughly three categories.

The first category is a collection of papers devoted to the analysis of layered (or interdependent) networks, and it is presented in Section 4.1. A second category is a mapping between tripartite networks and hypergraphs, in Section 4.2. In Section 4.3 we present those publications related to the first examples of multidimensional network analysis that are closest to our chosen model, that is the third category we can define. Finally, we take a look to the techniques of tensor decomposition in Section 4.4: in this case we are not dealing with an alternative model for multirelational networks, but to a toolbox that can be used in the presented frameworks.

4.1 Layered Networks

A layered network is a collection of different networks, the layers, whose nodes are interdependent to each other. In practice, nodes from one layer of the network depend or control nodes in a different layer. In the chosen representation, these dependencies are additional edges connecting the different layers. This structure, in between the network layers, is called "meso structure". A preliminary study about layered networks was presented in [158]. More recently, layered networks were used to study the interdependence of several real world infrastructure networks in [60] and [212], two works that also explore several statistical properties of these structures, there called interdependent networks, such as cascade failures and percolation.

In [60] the general concept of layered network is reduced for simplicity to a network with only two layers, but without loss of generality. The authors consider two networks, A and B, with the same number of nodes, N. The functioning of node A_i $(i = 1, 2, ..., N)$, in network A, depends on the ability of node B_i, in network B, to supply a critical resource, and vice versa. An example of this two-layers network is depicted in Figure 4.1a. This model was developed particularly to represent the real world fact that diverse infrastructures such as water supply, transportation, fuel and power stations are coupled together. Aim of [60] is to show that, owing to this coupling, interdependent networks are extremely sensitive to random failure, such that a random removal of a small fraction of nodes from one network can produce an iterative cascade of failures in several interdependent networks.

In the model, if node A_i stops functioning owing to attack or failure, node B_i stops functioning. Similarly, if node B_i stops functioning then node A_i stops functioning. The cascade failure generated by this process is depicted in Figures 4.1b, c and d. Authors denote such a dependence

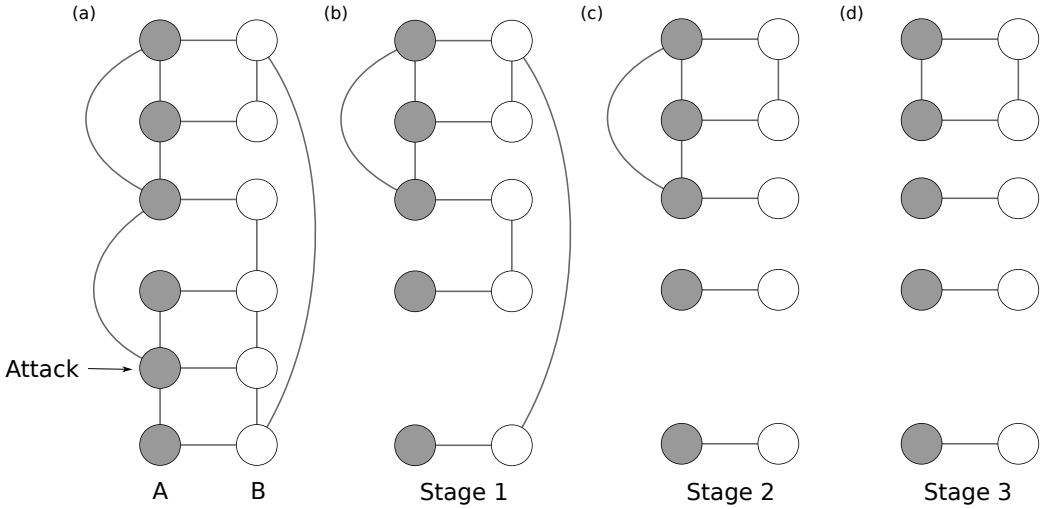


Figure 4.1: An example of layered network (a) and the process of a cascade failure involving the two different layers (b, c and d). In (a) the attacked grey node and its white dependent disappear, with all the edges attached to them, generating (b). Then, the cascade is triggered: the white nodes connected to the disappeared node lose their connections because they cannot sustain them anymore (b→c) and the same happens for the grey nodes (c→d).

by a bidirectional link in the meso structure, $A_i \leftrightarrow B_i$, that defines a one-to-one correspondence between nodes of network A and nodes of network B. Within network A, the nodes are randomly connected by A-links with degree distribution $P_A(k)$, where the degree, k, of each node is defined as the number of A-links connected to that node in network A. Analogously, within network B, the nodes are randomly connected by B-links with degree distribution $P_B(k)$.

Of course, the one-to-one correspondence does not hold in general. A single node in the layer A can be coupled with many different nodes in layer B. We will see that this makes the layered network the most natural way to represent interdependence.

The main finding of [60] is that for two interdependent scale-free networks with power-law degree distributions, $P_A(k) = P_B(k) \propto k^{-\lambda}$, the existence criteria for the giant component are quite different from those for a single network. As a consequence, two interdependent scale-free networks are not robust against random failures when a fraction lower than a critical value p_c does not survive to the first failure, while a single scale-free network does survive. This happens because high-degree nodes of one network can depend on low-degree nodes of the other. Moreover, authors show that the critical threshold p_c is generally higher for interdependent scale-free networks than for not interdependent networks (see the original paper for the mathematical details).

The same authors show with more mathematical details the presence of critical threshold in percolation processes in [212], creating the analogy between interdependent networks and ideal gases. In practice, introducing interactions between networks is analogous to introducing interactions among molecules in the ideal gas model. Interactions among molecules lead to the replacement of the ideal gas law by the van der Waals equation that predicts a liquid-gas first order phase transition line ending at a critical point characterized by a second order transition. Similarly, interactions between networks give rise to a first order percolation phase transition line that changes at the critical point to a second order transition, as the coupling strength between the networks is reduced.

Recently, another group of authors proposed a model called multilevel network [80]. We found that this model lies in between our proposed definition of multidimensional network and the layered networks, but it has more common points with the latter. The formal definition of a multilevel network is the following. Let $G = (V, E)$ be the network. A multilevel network is a triple $M = (V, E, S)$, where $S = \{S_1, S_2, ..., S_n\}$ is a family of subgraphs $S_q = (V_q, E_q)$ of G such that

$$G = \bigcup_{q=1}^{p} S_q,$$

i.e. $V = V_1 \cup V_2 \cup ... \cup V_p$ and $E = E_1 \cup E_2 \cup ... \cup E_p$. The network G is the projection network of M and each subgraph $S_j \in S$ is called a slice of the multilevel network M. Up until now, this definition is isomorphic to the one provided for the multidimensional network in Chapter 3. Let us consider a dimension d and the equivalent slice S_d. As presented in Chapter 3, dimension d is the collection of edges labeled with the label d, or $D = \{(u, v, x) \mid x = d \in D\}$, while S_d is the collection of nodes and edges of relation d, i.e. all the u and v that are present in at least one edge labeled with d.

However, to perform more advanced analysis such as the shortest path detection, authors introduce also a meso structure that they call auxiliary graph. Every vertex of the multilevel network M is represented by a vertex in the auxiliary graph and, if a vertex in M belongs to two or more slice graphs in M, then it is duplicated as many times as the number of slide graphs it belongs to. Every edge of E is an edge in the auxiliary graph and there is one more (weighted) edge for each vertex duplication between the duplicated vertex and the original one. This operation breaks the isomorphism with multidimensional networks and brings this representation very close to a layered network. However, these two models are not completely equivalent, since in the multilevel network the one-to-one correspondence of nodes in different slices is strict, while this condition does not hold for layered networks.

In [80] authors then define the extension of classical complex network measures for multilevel networks, such as the slice clustering coefficient; and more advanced concepts, such as the efficiency of a multilevel network, as well as a collection of network random generators for multilevel structures.

The main advantage of layered network is the ability of mapping a node in one relation with many different nodes in another relation. The meso structure can connect a single node in network A to several different nodes in network B. This operation is not possible in a traditional multidimensional networks, since there is no explicit meso structure defined and therefore a node in a multidimensional network is a single and a not divisible entity. This advantage is very convenient in the interdependence studies, since it provides a more natural way to deal with multiple dependencies in between the multiple relations.

However, it is possible to map the layered network model on the multidimensional network model, by imposing several constraints. Firstly, each different layer (A, B, C, ...) of the layered network is a different dimension of the network, i.e. it is represented in the multigraph with a different label. Then, we can define an additional dimension (an additional label in L) whose function is to represent the meso structure. Finally, we need to impose a set of constraints for the network connectivity. In particular, between two nodes there can be one and only one dimension: if the two nodes are part of the same layer this dimension must be the dimension representing that particular layer. If the two nodes are part of different layers, the dimension must be the one representing the meso structure. To derive our collection of measures and analytic procedures in the layered setting, and vice versa, is a possible future research scenario.

The procedure described to derive the layered model as a special case of the multidimensional model highlights also what is the main limitation of using interdependent networks with a meso structure. This limitation lies in the connectivity constraints. If between two nodes there can be one and only one dimension, then all the interplay among different relations cannot be derived easily. For instance, we know that in a social network the friendship relation may be influenced by a working connection. How can we easily derive this influence if between the same two nodes we can have either friendship or work, but not friendship and work together? In other words, layered networks work very well in establishing dependencies among nodes, but not among edges.

In literature, it is possible to find already some advanced works that make use of layered networks. The main example we discuss is multidimensional community discovery. In [192], authors extend the popular modularity function for community discovery to adapt its implicit null model to fit a layered network, that they call multiplex. As we presented in Section 2.3, the main idea is

to represent each layer (that can be a snapshot, as well as a different relation) with a slice. Each slice s of a network is represented by adjacency A_{ijs} between nodes i and j. The authors also specify the meso structure, that they call "inter-slice couplings", as C_{jrs} that connect node j in slice r to itself in slice s. They notate the strengths of each node individually in each slice, so that $k_{js} = \sum_i A_{ijs}$ and $c_{js} = \sum_r C_{jsr}$, and define the multislice strength $\kappa_{js} = k_{js} + c_{js}$. The authors then specify an associated multislice null model. They then derive a resulting multislice extended definition of modularity, that we reported in Section 2.3.4.

4.2 Hypergraphs

A hypergraph is a generalization of a regular graph in the sense that an edge can connect multiple vertices. Thus, unlike in a regular graph where an edge connects two vertices, in a hypergraph a hyperedge is a collection of an arbitrary number of vertices. These vertices can be of the same or different types, and hyperedges can vary in the number of vertices they connect.

In [272], the authors represent the network as tripartite graphs consisting of three different types of vertices. The edges represent three-way hyperedges that each connect exactly three vertices.

This representation corresponds to the case of a tripartite hypergraph $G = (V, H)$ which can be defined as a pair of sets V and H, that satisfy the following conditions: firstly, the set $V = \{V_1 \cup V_2 \cup V_3 | V_i \cap V_j = \emptyset\}$ is formed by the union of three disjoint sets of vertices. Secondly, the set $H \subset \{(v_1 \in V_1, v_2 \in V_2, v_3 \in V_3)\}$ of hyperedges is a set of triangles connecting elements of these three sets.

The case study of the paper involved modeling folksonomies with hypergraphs. In practice, they have three sets of entities: users, content and tags. Each hyperedge connects an user and a piece of content with a tag, used by the user to annotate the content. Also in this case, there exists a mapping between this representation and the multidimensional networks. This mapping consists in projecting one set of entities as dimensions and recreating a regular bipartite multigraph. If we project over tags, we get a bipartite graph that connects users to content. Each user can use multiple tags to annotate the same piece of content, and this lead to multidimensionality. In other words, the tag "temples" or "Rome" are two different dimensions. If the set of tags is very big, this procedure is not feasible, making the projected graph crowded with too many dimensions.

In [272] the authors provide a collection of useful metrics on tripartite hypergraphs, making this model very robust for basic analysis. This is the second most important advantage of this model, the first being explained before as a very natural way to model tripartite interactions in folksonomies. Along with the extension of the classical node degree, authors provide also an edge degree, namely the number of hyperedges a couple of vertices is part of. Authors test the distribution of both these definitions of degree in real world networks extracted from Flickr and CiteULike. The vertex degree is, as common in social networks, a fat-tailed distribution, and also the edge degree distributions are right skewed. Authors also adapt the clustering coefficient to this setting, via what they call hyperedge density $D_h(k)$, and the vertex to vertex distance, a measure needed to develop algorithms to navigate into the hypergraph structure. Finally, they also define a vertex similarity measure that is used for a preliminary community discovery algorithm on hypergraphs.

In a previous work [104], the authors investigate the theory of random tripartite hypergraphs with given degree distributions. An extension of the configuration model for traditional complex networks is provided. Then, a theoretical framework in which analytical properties of these random graphs are investigated is provided. In particular, authors analyze the conditions that allow the creation of giant components in tripartite hypergraphs, along with classical percolation events in these complex structures. Other research groups has devoted some attention also to this model [81].

Thanks to the work of these authors, tripartite hypergraphs and their properties are now well understood. However, this model is tailored on folksonomies and on a very particular class of phenomena, namely all the phenomena that involves complex interactions among three class of entities. This scenario is far from universal. If we are interested in a phenomenon that involves

only users interacting with each other through different media, then this model is no more suitable because it is too complex: one would need to add at least one hypothetical class of entities besides users and media themselves, that in multidimensional networks are represented naturally with nodes and edge types.

We continue to use multidimensional community discovery as the main test case for the applications of the models presented in this chapter. One of the main examples of hypergraph analysis is represented by Metafac [178]. Also in this case, we will present more details in Section 2.3. In this work, the authors use relational hypergraphs to represent multi-relational and multi-dimensional social data. Instead of using tripartite hypergraphs, authors use a more general model composed by M-way hyperedges to represent the interactions of M facets (groups of entities of the same type, i.e. users, contents, tags...). The metagraph is defined as a set of data tensors. Then tensor decomposition and factorization operations are defined in order to unveil the community organization among facets.

4.3 Multidimensional Networks

In this section we focus on the part of the literature that uses multigraphs to model network with multiple kind of interactions, i.e. the model proposed and studied in this thesis. In sub section 4.3.1 we present the examples of multidimensional community discovery, i.e. detecting the modular structure of a network in which multiple relations are expressed at the same time. Another interesting problem is the prediction of multidimensional links. In Section 4.3.2 we briefly present one recent approach for this problem. We found that researchers in this field focus particularly on signed networks, or networks in which the multiple relations can be classified in "positive" and "negative" relations. We analyze those publications in Section 4.3.3.

It is important to note that the term "multidimensional", that we use here to describe the particular representation we are interested in, is far from universally accepted. The terminology referring to networks with multiple different relations has not reached a consensus yet, even among those researchers that are actually using the same model to tackle the problem of multiple interactions. Moreover, different scientists from different fields used similar terminologies for different models and different names for the same model. Thus, what we call "multidimensional" is often referred as multiplex, multislice, multirelational, multifaceted and this list is far from complete. However, the analysis we present in this section can be mapped entirely with the labeled multigraph we use as representation of multidimensional networks.

4.3.1 Multidimensional Community Discovery

To find densely connected modules in a complex multidimensional network is not an easy problem. We will see in Section 8.1 that the concept of "multidimensional density" is intrinsically ambiguous. Nevertheless, there are two important research tracks on this topic (also in this case, more details will be provided in Section 2.3).

The first one is represented by a collection of papers that investigate the possibility of extracting the latent social dimensions from real world networks [245, 246]. The dimensions are extracted using a classifier which not only considers the connectivity of a node, but assigns additional information to its connection i.e. a description of a likely affiliation between social actors. The basic assumption is borrowed from the concept of homophily, which states that actors sharing certain properties tend to form groups [185].

The second one [61] aims to discover hidden multidimensional communities. The authors start from the basic assumption that each relation (explicit or implicit) plays a different role in different tasks. Therefore they allow an analyst to specify custom community definitions. Then, all the dimensions have a different importance according to the community definition (some are important, some others are noise). All the relations are then weighted accordingly to how they reflect the community definition using a regression model. Finally, the communities are extracted.

4.3.2 Multidimensional Link Prediction

Given a pair of nodes in an evolving network, the literature on monodimensional network analysis defines link prediction as the problem of estimating the likelihood that an edge will form between two nodes [207]; we have discussed this problem in Section 2.5. In multidimensional networks this translates in estimating the likelihood that an edge will appear between two nodes in a specific dimension. In practice an additional degree of freedom is added to the classical definition.

A preliminary work [226] explores how useful a set of simple multidimensional measures can be if added to well-known link predictors. Authors show that using the established link predictors and adding a correction for multidimensionality and temporal evolution is sufficient to get slightly improved performances, even if the need of a truly multidimensional link predictor is proved. We will analyze more deeply this case in Section 8.3.

But multidimensional link prediction is obtaining an increasing attention in literature, and novel algorithms are being proposed. One of the main examples is [243]. The goal is to systematically define the relations between entities encoded in different paths using the metastructure of these paths, that authors call the "meta paths". For example, in the real world we have authors publishing papers in different venues. If two authors published two different papers in the same venue, we may have the chain "Jim-P5-SIGMOD-P6-Mike". This chain is translated in a meta-path "author-paper-venue-paper-author". Then, several measures are proposed to quantify the meta path-based relations, each of which quantifies the relation in a different way. In other words "paper-venue-paper" is simply a dimension that may connect two authors. This relation and several different other relations are studied in the paper. In our case, the relation is defined as co-venue publication (we will see in Chapter 5 that this dimension definition is used for the DBLP Conference multidimensional network). Authors then use a supervised learning framework to learn the best weights associated with each topological feature.

4.3.3 Signed Networks

A sub-problem in link prediction for multidimensional networks considers some specific properties of the dimensions. For example, each relation in a multidimensional network can be tagged as "positive" or "negative". Our main reference for this particular branch of research is [244], but other research groups share part of both methodology and problem definition with this work, such as [170].

In [244] six dimensions are extracted from the actions of tens of thousands players of a massive multiplayer online game. These six dimensions are either positive (friendship, trade, communication) or negative (attack, enmity, bounty). Then statistical properties, such as dimension interactions or degree distributions, are studied, highlighting the fact that truly different dynamics are working behind the curtain to shape the evolution of these relations.

Then, authors focus on the structural balance problem, particularly on the triangle, a very basic network structure. When we have positive and negative links, there are four possible different triangle configurations. The edges can be all positive, all negative, two positive and one negative and vice versa. Social balance theory, in its strong form [63], claims that there are "balanced" triads, where the links are all positive or there are two negative links (i.e. one element is enemy of both two allies) and "unbalanced" triads, where the links are all negative or there are two positive links (one element is allied with two enemies). Unbalanced triads are sources of stress and therefore tend to be avoided by agents when they update their personal relationships. In fact, the authors are able to prove that balanced triads are heavily over represented in the network, while unbalanced triads are very rare.

These results are confirmed by [170]. In this case the data sources are different, namely a trust network in which users connect with each other by deciding if a particular user is trustworthy or not, a social network that allows to create "enemy" links, and the register of Wikipedia votes (positive or negative) expressed by the community for the promotion of users to administration tasks. In all these different kinds of networks, all with positive and negative relations, the social balance proves to be one of the main features in order to increase the link predictor performances.

4.4 Tensor Decomposition

In this section we briefly provide an overview of mathematical tensors. Also, tensors are not a general model for multidimensional networks. Multidimensional networks, and hypergraphs, can be represented with tensors, but tensors are just the tool in which different models could be implemented. However, tensors are used for several other purposes and their usefulness is not bounded to their contribution to multidimensional network analysis. Our main reference is [153].

A tensor is a multidimensional array. More formally, an N-way or Nth-order tensor is an element of the tensor product of N vector spaces, each of which has its own coordinate system. The order of a tensor is the number of dimensions, also known as ways or modes. Scalars are tensors of order zero. A vector is a tensor of order one. A matrix is a tensor of order two. Tensors with order three or higher are called higher-order tensors. Of course, if a network can be represented with a two-order tensor (a matrix), then we can represent a multidimensional network with a three order tensor.

Some basic operations on tensors are defined such as the extraction of fibers and slices. A fiber is defined by fixing every index but one. Slices are two-dimensional sections of a tensor, defined by fixing all but two indexes. Other operations are defined, such as matricization (transforming a tensor into a matrix) and tensor multiplication. We are particularly interested in tensor decomposition, since many real world problems can be expressed in a tensor form and then solved by tensor decomposition.

Several different decomposition strategies and applications has been studied since decades. We recall the CANDECOMP (canonical decomposition) by Carroll and Chang [62] and PARAFAC (parallel factors) by Harshman [125]. One of the most popular strategies is actually the combination of the two, the CANDECOMP/PARAFAC (CP). Very briefly, the CP decomposition factorizes a tensor into a sum of component rank-one tensors. In a computer science perspective, the main problem is that there is no finite algorithm for determining the rank of a tensor. Consequently, the first issue that arises in computing a CP decomposition is how to choose the number of rank-one components. Most procedures fit multiple CP decompositions with different numbers of components until one is "good". Only recently some memory efficient and low time complexity frameworks has been proposed [154]. In this last work, and in [1], some examples of applications of tensor decomposition to complex network analysis have been provided.

Chapter 5

Real World Multidimensional Networks

In this section we present a collection of real world datasets from which it is possible to extract multidimensional networks. For each source of data we briefly describe the structure of the real world entities they are describing. Then, we present how we extracted a network representation of the phenomenon we want to analyze. Finally, we also describe the different dimensions of this phenomenon, that constitute the different relations inside the network.

For the last point, a caveat is needed. Dimensions in network data can be either *explicit* or *implicit*. In the first case the dimensions directly reflect the various interactions in reality; in the second case, the dimensions are defined by the analyst to reflect different interesting qualities of the interactions, that can be inferred from the available data. The formalization of this distinction is not our invention. The distinction is already proposed in [192], where the authors deal with the problem of community discovery. In their paper, our conception of multidimensional network is referred as *multislice*, networks with explicit dimensions are named *multiplex*, and also the temporal information is used to derive dimensions for the network. For more information, see Chapter 4.

Examples of networks with explicit dimensions are social networks where the dimensions of the interactions are a representation of communications through different means: email, instant messaging services and so on. An example of network with implicit dimensions is a co-authorship network where an interaction between two authors may happen in different years and the year when the collaboration took place is our dimension.

5.1 Facebook

This network is a small ego-centered network extracted from the popular social media site[1]. It was built considering the direct neighbors of the author of this thesis. We took the direct friends and then we built all the connections among them removing the ego node (which is obviously connected to everyone, thus creating noise). These connections are established through 10 different dimensions. We end up with 228 nodes and more than 3k edges (more topological statistics, namely n for the number of nodes, m for the number of edges and $\bar{k}$ for the average degree, are reported in Table 5.1). Our dimensions are: the direct friendship, the affiliation to a common group or event, the co-appearance in a photo, the co-comment or co-"like" (a function of the social media) about a particular object, or to be tagged in the same message or video. With the exception of the friendship dimension, all the other dimensions are built with a "tf-idf" approach [229], i.e. the groups, or events, too popular in this network are penalized and do not lead to the creation of an edge in that particular dimension.

It is well known that many people belongs to many communities. Extracting the ego network of

[1] http://www.facebook.com/

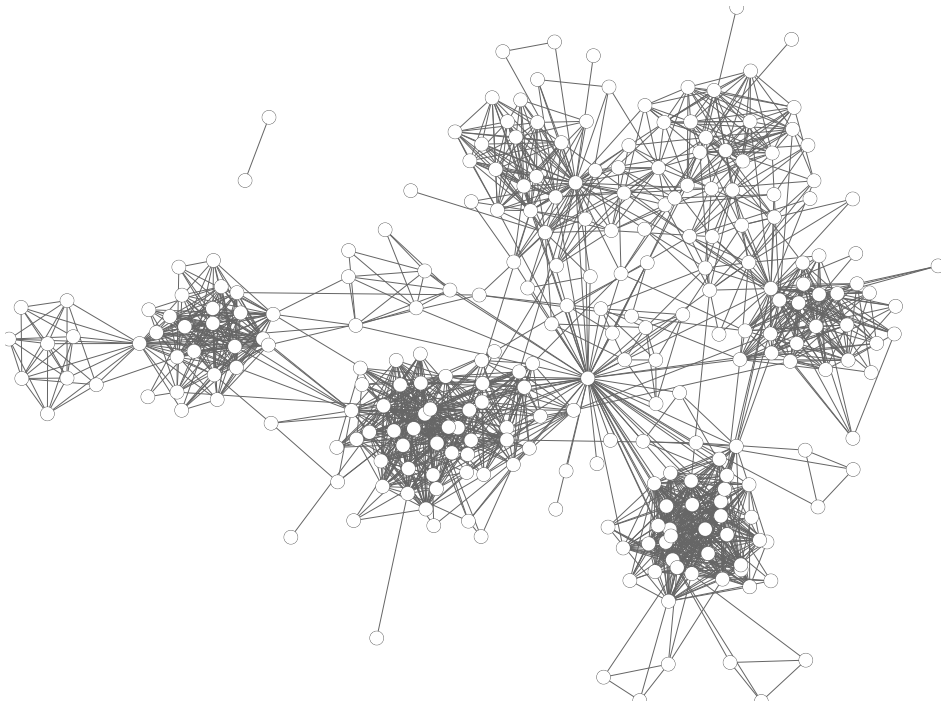


Figure 5.1: The friendship dimension in Facebook network.

a Facebook user, it is possible to easily detect the various communities he/she belongs. To clarify this concept, consider Figure 5.1: here we have depicted our Facebook network considering only the friendship dimension. It is straightforward to identify denser parts of the graph, corresponding to the various communities of the center node: the high school community, the university community, the working community, an online community to which he belongs, and so on. What is not trivial is to discern among different "types" of communities, or to find actors inside these communities which establish among themselves multidimensional connections.

These applications and research questions are the criteria driving the choice about which dimension is needed to be included in the network. In our case, including co-tagging is a way to obtain explicit and implicit information that should strengthen the concept of "friendship". Instead, using the "group affiliation" dimension is a way to capture an orthogonal information w.r.t friendship, as to be part of the same online group is not necessary to have some kind of real world relationship (that is intuitively more important to be tagged in the same photo). These analytic choices are the basis of the dimension definition for all the networks presented in this chapter and used in this thesis.

This is a small dataset and a not real world scale network. We use it as a test dataset because we know almost everything about the entities inside this network. Our aim in the analysis of this network is to verify that the extracted knowledge is robust and reflects actual real world relationships. Once we have verified it, we can apply our techniques to other real world scale networks, on which we can perform our robust knowledge extraction process even if we don't know almost anything about the entities populating them.

5.2 Supermarket

This network has been created starting from the sales data of a chain of Italian supermarkets. In this network the nodes are the customers of the supermarket, linked to each other if they share the same buying behavior. We started with 90M original transactions during a year of 838k customers and 318k different products. We used the 732 marketing categories used by the supermarket owners

Dataset	Dimension	n	m	$\bar{k}$
Facebook	Friendship	225	1,371	12.186
	Group	118	494	8.372
	Comment	64	92	2.875
	Likes	83	337	8.120
	Photo Tag	154	439	5.701
	Status Comment	133	236	3.548
	Status Tag	14	15	2.142
	Video Tag	17	18	2.117
	Event	88	259	5.886
	Note Comment	48	50	2.083
	Global	**228**	**3,311**	**29.043**
Supermarket	Mozzarella Cheese	1,578	8,162	10.344
	Bread	1,749	6,912	7.903
	Clementine	1,101	4,061	7.376
	Bananas	1,291	5,282	8.182
	Short Pasta	1,420	6,741	9.494
	Red Meat	1,329	5,081	7.646
	Canned Vegetables	1,320	4,808	7.284
	Long Pasta (Spaghetti)	1,312	5,187	7.907
	Milk UHT	1,665	7,202	8.651
	Mineral Water	1,998	12,141	12.153
	Global	**4,463**	**65,577**	**29.386**

Table 5.1: Main statistics about Facebook and Supermarket networks, and their dimensions.

to cluster the products. We then selected a period of two weeks, 4k random customers and 10 marketing categories, which are our dimensions. A report of basic topological statistics for this network is provided by Table 5.1.

"To share the same buying behavior" does not mean that two customers buy an high or the same quantity of a particular good. It means that a particular good has the same importance with respect to the total purchases of the customers. Thus a similarity measure for the triple $\{customer, customer, product\}$ is needed. We chose as similarity measure the Revealed Comparative Advantage. For each couple $\{customer, product\}$, the Revealed Comparative Advantage is defined as

$$RCA(c,p) = \frac{\left(\frac{x(c,p)}{\sum_p x(c,p)}\right)}{\left(\frac{\sum_c x(c,p)}{\sum_{c,p} x(c,p)}\right)},$$

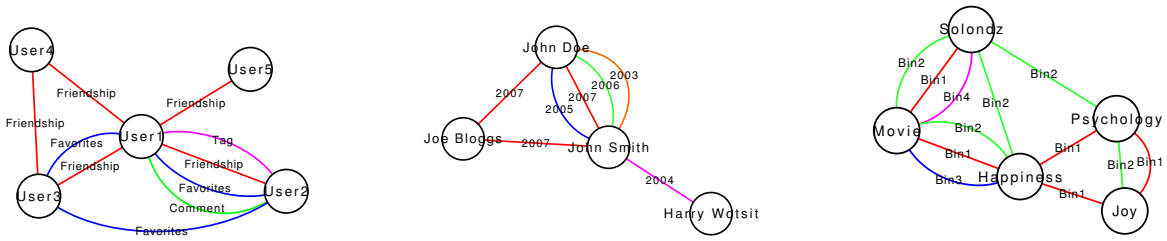
where $x(c,p)$ is the total amount purchased by the customer c of the product p [131]. RCA is larger than one when the share of purchases of a customer of a given product is larger than the share of that product on the global supermarket purchases. This measure is an equivalent to the lift, a known concept in association rules mining [54].

We can now define the similarity $\phi(i,j,p)$ between the customers i and j on a particular product (dimension) p as

$$\phi(i,j,p) = \min\left\{\frac{RCA(i,p)}{RCA(j,p)}, \frac{RCA(j,p)}{RCA(i,p)}\right\}.$$

$\phi(i,j,p)$ is one if the RCAs for i and j are the same and tends to zero the more the RCAs differs. We can create an edge if $\phi(i,j,p)$ is above a given threshold. For our purposes we found convenient $\phi(i,j,p) > 0.9$.

Using marketing categories as dimensions is useful for a particular problem definition. A supermarket may be interesting in spotting incomplete customer profiles. In other words, customers



(a) (b) (c)

Figure 5.2: Small extracts of the three real multidimensional networks.

may buy from a supermarket a collection of very specific products, but then buy other products in other shops. For example, one may particularly trust a store across the street exclusively on meat products more than the meat supermarket department. In this case, by grouping together densely connected nodes in many redundant dimensions and spotting some dimensions that are unexpectedly absent in these groups, we can identify these weak marketing categories on which the supermarket may want to establish more competitive offers.

5.3 Flickr

This dataset comes from the well known photo sharing service[2], and was obtained by crawling the data via the available APIs. We extracted both implicit and explicit dimensions of the social network represented in this data. For each picture, we extracted the list of all the users related to it and from these users we completed the social network by adding edges if two users commented, tagged or set the same picture as favorite, or if they had each other as a contact. From roughly 1.3M users we obtained slightly more than 900M edges, distributed on four dimensions.

The resulting network is a person-person network, where each dimension is one of the "Friendship", "Tag", "Favorites", or "Comment", representing if the users are friends, tagged the same picture, marked the same picture as favorite, or commented on the same picture. A small extract of this network is represented in Figure 5.2(a). Also, some basic statistics about the connectivity of the network, and each dimension taken singularly, are provided in Table 5.2 (also in this case, n is used for the number of nodes, m for the number of edges and $\bar{k}$ for the average degree).

The multidimensional representation of user interactions in a social media service is important for improving, and keeping alive, the service. We already depicted a possible scientific study about people behavior in Section 5.1. Here, we instead focus on a service study scenario. If we consider the interplay among dimensions, several different behaviors can be profiled. In particular, it is easy to identify users that only connect to each other via friendship dimension. Or they interact exclusively with social channels and do not care about tag or favorite photographs. These sets of users may need to obtain some kind of incentives to start to fully use the social media platform. Further, it is possible to connect these behaviors with service abandon ratio, thus providing a tool to prevent the website to burn out all its user capital.

5.4 DBLP

This dataset comes from the popular bibliographic database[3]. We built a co-authorship network of authors (nodes) connected by an edge if they wrote a paper together. The choice for the dimension definition in this case needs a careful explanation. We want to use the DBLP dataset as proof that the distinction between explicit and implicit dimensions is not only possible, as we have already shown, and the two different dimension classes are not mutually exclusive in the same network. In

[2]http://www.flickr.com
[3]http://www.informatik.uni-trier.de/~ley/db

Dataset	Dimension	n	m	k	Density
Flickr	Friendship	984,919	48,723,010	98.938	$1.00e^{-4}$
	Comment	930,526	198,309,709	426.231	$4.58e^{-4}$
	Favorite	380,992	674,488,956	3540.698	$9.29e^{-3}$
	Tag	91,690	715,447	15.605	$1.70e^{-4}$
	Global	**1,186,895**	**922,237,122**	**1554.033**	$3.27e^{-4}$
DBLP-C	**Global**	**30,177**	**84,531**	**5.60**	$5.98e^{-6}$
DBLP-Y	**Global**	**582,201**	**2,648,845**	**9.09**	$7.81e^{-6}$
DBLP	**Global**	**582,201**	**2,733,376**	**9.38**	$1.68e^{-7}$
QueryLog	Bin 1	138,992	1,104,581	15.894	$1.14e^{-4}$
	Bin 2	108,439	878,136	16.195	$1.49e^{-4}$
	Bin 3	89,418	708,897	15.855	$1.77e^{-4}$
	Bin 4	75,846	583,774	15.393	$2.02e^{-4}$
	Bin 5	42,951	253,976	11.826	$2.75e^{-4}$
	Bin 6	12,236	36,456	5.958	$4.87e^{-4}$
	Global	**184,760**	**3,565,820**	**38.599**	$3.48e^{-5}$

Table 5.2: Summary of the datasets extracted from Flickr, DBLP and Querylog. Column 1 specifies the dataset; Column 2 the dimension into account; Columns 3 and 4 the number of nodes and edges; Column 5 the average degree; Column 6 the density computed as number of edges out of number of total possible edges in all the dimensions

other words, in the same network, explicit and implicit dimensions can co-exist. Of course, they should co-exist if and only if they are both necessary to describe a particular phenomenon.

Let us now define these two different classes of dimensions in our DBLP scenario. Firstly, we use years as dimensions, and any pair of authors was connected in a specific dimension if they wrote at least one paper together in the corresponding year. We obtained roughly 600k nodes connected by 2.6M edges, distributed over 65 dimensions. The resulting network is a person-person network, where each dimension is on the years from 1938 to 2008 (with some gaps at the beginning), indicating whether the authors had a collaboration in the corresponding year. A small extract of this network is represented in Figure 5.2(b). The temporal dimension is implicit, since the relation per se is the same (co-authorship) and what does change is only a "quantity" of co-authorship. We refer to this network as DBLP-Y.

Secondly, we use the different publication venues as dimensions. We took only the publications in the most important 31 conferences in computer science, which include VLDB, SIGKDD, WWW, AAAI and more. Please note that in this case we are not constraining on the years. Also in this case the relation is, at its basis, the same (co-authorship), but the venue of a publication is a very strong distinction. To publish a paper in VLDB conference requires different competences, expertise and, possibly, even procedures, than publishing in a computer vision conference. We also may define the dimensions as keywords present in the paper title and/or abstract, but the venue is explicitly hard coded in our data source and we decided to use this as dimension, to reduce noise. We refer to this network as DBLP-C.

Aggregated statistics of both versions of DBLP networks (DBLP-Y, DBLP-C and aggregation) is reported in Table 5.2. Since the total number of dimensions defined on DBLP is close to 100, we do not report single statistics about the dimension connectivity for space constraints.

To have both explicit and implicit dimension in this setting is very useful to better characterize the research groups in DBLP. Without them, a clique of authors is simply a collaboration group and nothing more can be said about it. With this multifaceted data, we can distinguish occasional collaboration from persistent research groups, multidisciplinary collaborators from mono-thematic research tracks, and any combination of the four.

5.5 Querylog

This network was constructed from a query-log of approximately 20 millions web-search queries submitted by 650,000 users over a period of time[4], and was described in [213]. Each record of this dataset stores an anonymous user ID, the query terms, the date and hour of the query, the rank position of the result visited by the user on each record and the host portion of the URL of the visited result. From this dataset, we extracted a word-word network of query terms, consisting of roughly 200k words (nodes), after removing stop-words.

We connected two words if they appeared together in a query, producing roughly 2M edges. Dimensions are defined as the rank positions of the results, grouped into six almost equi-populated bins: "Bin1" for rank 1, "Bin2" for ranks 2-3, "Bin3" for ranks 4-6, "Bin4" for ranks 7-10, "Bin5" for ranks 11-58, "Bin6" for ranks 59-500. Hence two words appeared together in a query for which the user clicked on a resulting url ranked #4 will produce a link in dimension "Bin3" between the two words. The result is a word-word network, for which we give a small extract in Figure 5.2(c).

The Querylog network is different from what we have seen until now because its nodes are not people and the definition of its implicit dimension may seem obscure. However, it is tailored on a particular problem definition. Aim of this network definition is to provide a field where to test the performances of the search engine ranking system. The simple link distribution among the dimensions is already an interesting measure of the search engine performances: the more links appear exclusively in dimension "Bin1" the better. But since we are in a network, we are not interested in a single word query but in the complex interaction between query terms. Therefore, to find terms surrounded mainly by dimensions "Bin4", "Bin5" and "Bin6" raises an alert about the way the search engine is operating for those particular ambiguous terms.

5.6 GTD

From this database of global terrorism[5], we created a group-group graph for the years 1969-2008, where each node represents a terrorist group or organization, and two groups are connected if they participated in a terrorist attack to the same country (note that the two groups only attacked the same country, but they do not need to have collaborated to the attack in order to be connected). We then considered each year as temporal snapshot, generating 40 snapshots. As for DBLP, the snapshots are non-cumulative, and we ended up with 2,279 nodes and 31,843 total edges.

This network is useful to characterize the history of global terrorism. An important observation, considering implicit dimensions defined with a quantitative logic, is that we can define an dimension order (just like in the DBLP-Y network). The year as dimension is an unambiguous quantity. Therefore, it is natural to order the dimension "2006" before dimension "2007" and after dimension "2005". By studying the relationships between consecutive dimensions, it is possible to extract uniform periods, and to characterize them using the nodes (the groups active in those years), the length of the period itself (the edge dimensions) and the hot areas (by checking the edge creation criterion).

5.7 IMDb

From the Internet Movie Database[6], we created a collaboration graph for the years 1899-2010, where each node represents a person who took part in the realization of a movie (directors, cast, song writers, and so on), and two persons are connected if they participated to the realization of the same movie. We considered each year as temporal snapshot, i.e. an implicit dimension, generating 112 of them. As for DBLP-Y, the temporal snapshots are non-cumulative, for a total number of 57,457 nodes and 13,047,319 edges.

[4] http://www.gregsadetsky.com/aol-data
[5] http://www.start.umd.edu/gtd
[6] http://www.imdb.com

This network has been used mainly together with both DBLP-Y and GTD networks. IMDb network shares with DBLP-Y network its general scenario: it is a structure that we use to investigate mainly co-authorship. This is simply an alternative setting in which collaboration takes place. The application we are interested in is shared with GTD network: we are mainly interested in analyze a very long collaboration history. IMDb data is very rich since the beginning of movie industry. Starting from the year 1900, the data is very reliable and the quantity is enough to apply complex network and data mining analysis. The data quality and the wide temporal window (more than a century) makes IMDb a unique data source.

5.8 Classical Archaeology

Lastly, we present the dataset we will use in the third part of this thesis to show a real world complex scenario for multidimensional network analysis. We will provide more details about this data source in Chapter 10.

We made use of Archäologische Bibliographie, i.e. a bibliographic database that collects and classifies literature in classical archaeology since 1956 [234]. Analyzing the state of 2007, our source data includes about 370,000 classified publications authored by circa 88,000 archaeologists that are connected to about 45,000 classification criteria, via 670,000 classification links. The classification criteria themselves are manually grouped into different categories: subject themes, locations, periods, persons and objects.

Firstly we generate a classification co-occurrence network from the classification link between publications and classification criteria. In this case we already have a multidimensional network with implicit dimensions, since we use the year of the publication as dimension for the co-classification link.

However, this is not the only structure we extract from this data source. After performing an overlapping community discovery on this structure, we are able to group together classification criteria into groups. Since each classification criterion may be part of more than one group (for instance, the classification “Paestum” is part of 12 communities), we are able to draw a link between two different groups, weighted accordingly to the number of shared classifications. This community network is not only weighted, it is also multidimensional. The classification criteria, as stated before, are clustered into categories. Each one of this categories defines then a link type, i.e. a dimension, in this structure.

We will see in Chapter 10 how these two structures can be analyzed and used to solve some critical problem typical of classical archaeology, namely the difficulty of finding related works for an authors and of navigating through literature. These problems have nothing to do, apparently, with complex network theory in general and multidimensional networks in particular, but the techniques and measures defined in this thesis play a crucial role in their solutions.

Part II

Multidimensional Network Analysis

Chapter 6

Extension of Classical Measures

In this chapter we discuss the most straightforward and basic changes in the playing field of complex network analysis when allowing multidimensionality. This chapter is mainly focused on verifying what happens to the degree and to the shortest path, and their derivable measures, allowing multiple kinds of edges. Besides describing how the analytical measures defined on standard graphs can be extended to cope with multiple dimensions, we also define new aggregate functions induced by some local or global measures.

In general, in order to adapt the classical measures to the multidimensional setting we need to extend the domain of each function in order to specify the set of dimensions for which they are calculated. Intuitively, when a measure considers a specific set of dimensions, a filter is applied on the multigraph to produce a view of it considering only that specific set, and then the measure is calculated over this view. In the following, we redefine some of the classical measures on graphs and networks, using the presented approach. After this set of measures, in Chapter 7 we present the new measures we introduce in the multidimensional setting, that are meaningful only in this scenario.

The general notation is the following, and it is consistent with the notation used in Chapter 3. V is the set of nodes of the network, E is the set of edges, D is the set of all dimensions of the network and, when needed, D' indicates a subset of D.

6.1 Degree Related Measures

We start by considering what happens to the degree in a multidimensional network. We use this section also to provide the general operation for the extension of traditional measures into the novel setting. In order to cope with the multidimensional setting, we have to define the degree of a node w.r.t a single dimension, w.r.t a set of dimensions and we need also to analyze the average degree of a node within the network. To this end we have to redefine the domain of the classical degree function by including also the dimensions. This operation is standard and general and, as we will see, it also holds for more complex traditional measures such as the closeness centrality.

Definition 1 (Degree) *Let $v \in V$ be a node of a network G. The function $Degree : V \times \mathcal{P}(D) \rightarrow \mathbb{N}$ defined as*

$$Degree(v, D') = |\{(u, v, d) \in E \ s.t. \ u \in V \ \wedge \ d \in D'\}|$$

computes the number of edges between v and any other node labeled with one of the dimensions in D'. □

As it can be done for most of the measures that we present further, for this measure we can consider two particular cases: when $D' = D$ we have the degree of the node v within the whole network, while when the set of dimensions D' contains only one dimension d we have the degree of v in the dimension d, which is the classical degree of a node in a monodimensional network. This

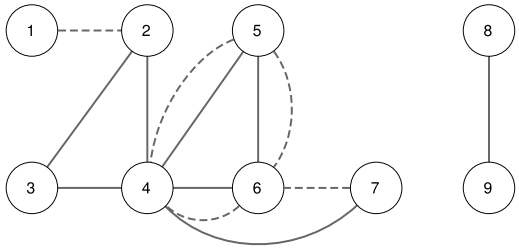


Figure 6.1: A toy example. The solid line is dimension 1, the dashed line is dimension 2.

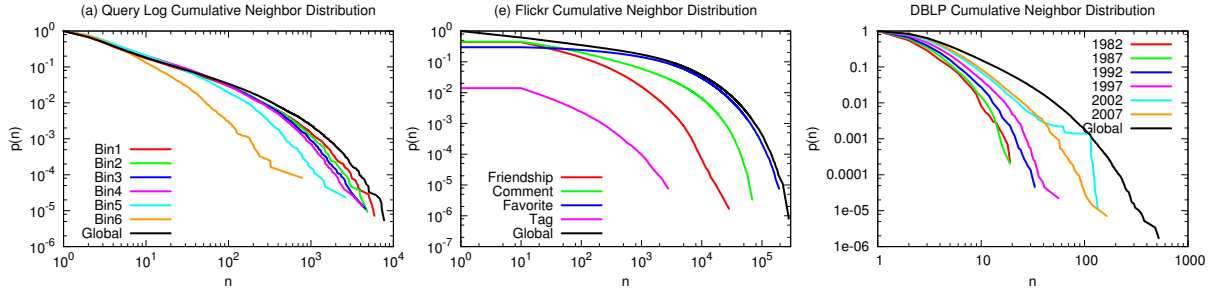


Figure 6.2: Cumulative distributions of degree per dimension and global degree for Querylog, Flickr and DBLP-Y.

consideration also holds for all the measures below extending the monodimensional case, thus we avoid to repeat it for each of them.

Besides computing the average degree of the network, by summing all the degrees of the nodes and dividing by the number of nodes, we can also induce an aggregate function that computes the average of the degrees of a node v computed in different dimensions, dividing by the number of dimensions considered.

Definition 2 (Average of the Degrees over dimensions) *Let $v \in V$ be a node of a network G. The function $AvgDegree : V \times \mathcal{P}(D) \to \mathbb{R}$ defined as*

$$AvgDegree(v, D') = \frac{Degree(v, D')}{|D'|}$$

computes the average degree of a node v over the specific set of dimensions D' of the network G.□

To illustrate the measures we define in this paper, we use a toy example, depicted in Figure 6.1, to show the application of the metrics on it.

Example 1 *Consider the multigraph in Figure 6.1 that models a multidimensional network with 2 dimensions: dimension d_1 represented by a solid line, and dimension d_2 represented by the dashed line. In this multigraph we have:*

- $Degree(3, \{d_1\}) = 2$
- $Degree(3, \{d_2\}) = 0$
- $AvgDegree(3, \{d_1, d_2\}) = (2 + 0)/2 = 1$
- $AvgDegree(3, \{d_1\}) = 2/1 = 2$

In Figure 6.2 we show that with this function it is possible to identify dimensions that are good representatives of the global degree distribution of the network, thus allowing tasks such as focused sampling or (lossy) compression of the graph. Consider the case of Flickr network (Figure 6.2b): "Comment" dimension shows a very similar exponential cutoff of the general degree distribution, but it has almost an order of magnitude less edges. Also consider "Tag" dimension for Flickr or "Bin 6" dimension for Querylog (Figure 6.2a). What is happening is that even inside a multidimensional network that does not present a power law degree distribution, since both the general degree distribution in Querylog and Flickr present a very strong exponential cutoff, there can be scale free dimensions, where the cutoff is evidently weaker.

Traditionally in complex network studies, degree distributions and average degree are theoretically linked with the study of the connected components of the network, focusing particularly on the preconditions for observing a giant component. We are not tackling the problem from a theoretical point of view, but we just provide some useful tools to derive an easily computable way to better understand the topology of the network. In the following we compute the number of connected components of a multidimensional network, including also a set of dimensions into account.

Definition 3 (Connected Components) *The function* $CC : \mathcal{G} \times \mathcal{P}(D) \to \mathbb{N}$*, called* Connected Components*, computes the number of connected component of a graph considering only the edges labeled with dimensions included in a given set* D'*. It counts the number of the maximal set of nodes that can be reached through paths only using the edges belonging to any* $d \in D'$*.* □

Definition 4 (Average of the Connected Components over Dimensions) *The function* $AvgCC : \mathcal{G} \times \mathcal{P}(D) \to \mathbb{R}$*, called* Average of the Connected Components over Dimensions*, is defined as*

$$AvgCC(G, D') = \frac{\sum_{d \in D'} CC(v, \{d\})}{|D'|}$$

and computes the average of the number of connected components over the specific set of dimensions D' *of a given network* G*.* □

Example 2 *Considering the multidimensional network of the Figure 6.1:*

- *if we consider only the dimension* d_1 *we have 3 components, as we can consider node 1 as a component composed by only one node*
- *if we consider only the dimension* d_2 *we have 5 components, as we can consider nodes 3, 8 and 9 as above*
- *if we consider all the dimensions of the network we have 2 components*
- *the connected component average of the network are* $(3 + 5)/2 = 4$

6.2 Shortest Path Related Measures

As done for the degree, the classical shortest path definition has to be revisited in order to deal with the multidimensional setting, by extending the domain with a set of dimensions.

Definition 5 (Shortest Path) *Let* $u, v \in V$ *be two nodes of a network* G*. The function* $ShortestPath : V \times V \times \mathcal{P}(D) \to \mathbb{N}$ *computes the length of the shortest path (in terms of number of edges) between* u *and* v*, where the edges are labeled with dimensions in* D'*.* □

As in the classical definition, if are no paths between two nodes then the distance between them is ∞. We also define the *Average Shortest Path* and the *Average of the Shortest Paths over dimensions*, which are two aggregate functions.

Definition 6 (Average Shortest Path) *The function* $ShortestPath_{AVG} : \mathcal{P}(D) \rightarrow \mathbb{R}$ *is defined as*

$$ShortestPath_{AVG}(D') = \frac{\sum_{\forall p \in SP_{D'}} Length(p)}{|SP_{D'}|}$$

where:

- $SP_{D'}$ *denotes the set of shortest paths having only edges labeled with dimensions belonging to* D'*, between node* v *and any node* u *reachable from it.*
- $Length(p)$ *denotes the length of the shortest path* p *in terms of number of edges.*

It computes the average shortest path considering only the set of dimensions D'*.* □

Definition 7 (Average of the Shortest Paths over dimensions) *Let* $u, v \in V$ *be two nodes of a network* G *such that* v *is reachable from* u*. The function* $AvgShortestPath : V \times V \times \mathcal{P}(D) \rightarrow \mathbb{R}$ *defined as*

$$AvgShortestPath(u, v, D') = \frac{\sum_{d \in D'} ShortestPath(u,v,\{d\})}{|D'|}$$

computes the average of the lengths of the shortest paths between two nodes u *and* v *over the specific set of dimensions* D' *of the network* G*.* □

An interesting analysis that can be done when taking the dimensions into account in the definition of the shortest path is to verify the heterogeneity of the shortest path, i.e. verifying how many dimensions are traversed by a given shortest path. To this end, we define a function that computes the heterogeneity of a path.

Definition 8 (Path Heterogeneity) *Let* P *be a path between two nodes of a multidimensional network, i.e. a sequence of labeled edges. The* Path Heterogeneity *function computes the ratio of dimensions in* P *with respect to the dimensions in the whole network, i.e.* $PathHeterogeneity = \frac{|\{d \mid \exists (u,v,d) \in P\}|}{|D|}$ □

Given a set of paths it is possible to compute aggregate functions of the path heterogeneity measure, such as the average, the maximum and the minimum. In a multidimensional network, it is interesting to apply this measure on the set of the shortest paths: considering a transportation network, this would translate in knowing how many different trains, or tickets, a persons has to take to get to the destination. A possible variant of this is to count the number of "changes" of dimensions: even though the number of different crossed dimensions can be only two, it may happen that in order to go from one node to another one in the network, the shortest path is a sequence of $d_1 - d_2 - d_1 - ... - d_2$, which, in the transportation network, would mean to change train at every station. An interesting problem would be to modify Dijkstra algorithm for computing the shortest path [77] to include also the change of edge label (i.e. dimension) as additional cost of the shortest path. We investigate more deeply this problem in Section 8.4.

Example 3 *Continuing with the example of Figure 6.1 we have:*

- *the* $ShortestPath(1, 7, \{d_1, d_2\}) = 3$ *and its Heterogeneity is equal to 1, as this shortest path contains 2 edges of the dimension* d_2 *and 1 edge of the dimension* d_1*.*
- $ShortestPath(1, 7, \{d_1\}) = \infty$
- $ShortestPath(6, 7, \{d_1, d_2\}) = 1$
- $ShortestPath(6, 7, \{d_1\}) = 2$
- $ShortestPath(6, 7, \{d_2\}) = 1$

- $AvgShortestPath(6, 7, \{d_1, d_2\}) = 1.5$
- $ShortestPath_{AVG}(\{d_1, d_2\}) = 1.6$

The concept of shortest path has been used in complex networks to derive a collection of centrality measures, such as the closeness and the betweenness centrality. We now take a look what does change for these two measures with a multidimensional formulation of the shortest path.

The closeness centrality describes a particular kind of "importance" of a node within a network, in terms of distance of it from all the other nodes. In the standard definition this measure is only defined on nodes. As done for the above measures, we modify the definition introducing the dimensions. Please note that there are several different definitions of closeness centrality (such as the random-walk or the information centrality). The underlying logic needed to extend all the variants in the multidimensional case is fairly similar to the one presented here, and this is the reason why we present only this closeness version. In practice, what is needed is to select only the edges belonging to the set D of dimensions we are interested in, collapse them into a monodimensional view and apply the standard definition.

Definition 9 (Closeness Centrality) *Let $v \in V$ be a node of a network G. The function $Closeness : V \times \mathcal{P}(D) \to [0, 1]$ is defined as*

$$Closeness(v, D') = \frac{|\bar{V}|}{\sum_{u \in \bar{V}} ShortestPath(v, u, D')}$$

where $\bar{V}$ denotes the set of nodes reachable from v by a path, excluding v itself. □

Moreover, we define an aggregate function that computes the average of the closeness centrality computed over different dimensions.

Definition 10 (Average of the Closeness Centralities over Dimensions) *Let $v \in V$ a node of a network G. The function $AvgCloseness : V \times \mathcal{P}(D) \to [0, 1]$ defined as*

$$AvgCloseness(v, D') = \frac{\sum_{d \in D'} Closeness(v, \{d\})}{|D'|}$$

computes the average of the closeness centralities of a node v over the specific set of dimensions D' of the network G. □

Please note that in this measure we explicitly indicate the set D_v, as it is not meaningful to consider the closeness in dimensions where the node does not appear.

Example 4 *In the multidimensional network of the Figure 6.1:*

- *if we consider the entire dimension set and the node 7 we have six nodes reachable with a total number of 11 edges: $Closeness(7, \{d_1, d_2\}) = 6/11 = 0.54$*
- *if we consider only the dimension d_1 and the node 7 we have $Closeness(7, \{d_1\}) = 5/9 = 0.55$*
- *if we consider only the dimension d_2 and the node 7 we have $Closeness(7, \{d_2\}) = 3/5 = 0.6$*
- *the average of the closeness on the all the dimensions of the node 7 is $AvgCloseness(7, \{d_1, d_2\}) = (0.55 + 0.6)/2 = 0.57$*

While the closeness centrality takes into account the distance between a node and all the other nodes in a network, the betweenness centrality considers the number of shortest paths passing through a node, thus emphasizing the analysis of the resilience of the network to the removal of important nodes. Also in this case, we would like to include a set of dimensions into account, hence we modify the standard definitions, introducing the followings.

Definition 11 (Betweenness Centrality) *Let $v \in V$ be a node of a network G. The function $Betweenness : V \times \mathcal{P}(D) \to [0, 1]$ defined as*

$$Betweenness(v, D') = \frac{\sum_{s,t \in V} SP_{svt}(D')}{SP_{st}(D')}$$

where:

- *$SP_{svt}(D')$ denotes the number of shortest path between the nodes s and t passing through v, only considering edges belonging to the set of dimensions D'*
- *$SP_{st}(D')$ denotes the number of shortest path between the nodes s and t, considering only considering edges belonging to the set of dimensions D'.* □

We can also define an aggregate function that computes the average of the betweenness centralities computed on different dimensions.

Definition 12 (Average of the Betweenness Centralities over Dimensions) *Let $v \in V$ be a node of a network G. The function $AvgBetweenness : V \times \mathcal{P}(D) \to [0, 1]$ defined as*

$$AvgBetweenness(v, D') = \frac{\sum_{d \in D'} Betweenness(v, \{d\})}{|D'|}$$

computes the average of the betweenness centralities of a node v computed over the specific set of dimensions D' of the network G. □

Finally, the last measure influenced by the novel definition of the shortest path is the diameter. The classical definition of the diameter is the length of the longest shortest path between any pair of nodes in the network. Having re-defined the notion of "shortest path" we can define the concept of diameter in terms of it.

Definition 13 (Diameter) *The function $Diameter : \mathcal{G} \times \mathcal{P}(D) \to \mathbb{N}$ computes the length of the longest shortest path of a network G considering only edges labeled with dimension belonging to a specific set D'.* □

Clearly, on the diameter it is possible to define aggregate functions as for the shortest path. It is also interesting to measure the difference between the diameter computed considering a given set of dimensions and the diameter of the whole network.

Example 5 *On Figure 6.1:*

- $Diameter(G, D) = 3$
- $Diameter(G, \{d_1\}) = 2$

Chapter 7

Novel Measures

In this chapter, we present a set of basic measures. Differently from the previous chapter, the measures here presented make sense only in the multidimensional case, since by their formulation they have a trivial solution in a network with only one dimension. We provide examples and a case study for all of them to demonstrate their usefulness. The measures here presented are: the Dimension Relevance (Section 7.1), a class of measures to quantify and understand the importance of a dimension for the connectivity of a node; the Dimension Correlation (Section 7.2), whose aim is to unveil relationships and dependencies between the different dimensions in a network; and Dimension Connectivity (Section 7.3), a final class of measures to assess the importance of a dimension in the general ecology of a complex multidimensional network.

7.1 Dimension Relevance

Given a multidimensional network, one natural question is related to the importance of each single dimension in the economy of the connections of the network. An interesting question is then how to quantify this importance. The quantification can be calculated at two different granularity levels: at the global level of the network or at the local level of the single node. The first case provides a general and aggregate answer to the question "How much in general a dimension is important in the network?" and we explore this branch in Section 7.3. The second case, that is explored in this section, is focused on how much the dimension is important for the connectivity of a particular node. We want to be able to discern quantitatively if a node is exclusively connected with one single dimension, if this dimension connects it to its entire neighborhood but there are alternative dimensions or if it is structurally meaningless. We develop a new class of measures to quantify these different configurations and we call it "Dimension Relevance".

We introduce a case study in which we show the usefulness of the Dimension Relevance class of measures. One classical topic of research in complex network science is to find and to analyze *hubs*, i.e. nodes with a large number of neighboring nodes. Hubs are a typical class of nodes in scale-free networks. As we have seen, scale-free networks, i.e. networks with the degree distribution following a power law, have been studied for many years. The first study introducing the term "scale-free" was [30], where the authors discovered that the structure of the Web shows the presence of a few highly connected nodes, the hubs, and many nodes with a low degree. Other papers studied the same concept and tried to capture the "importance" of a node in a network: [151] is a well known example.

Since then, many papers have considered scale-free networks in several different areas of research. However, most of these studies are related to monodimensional networks. In this setting the concept of hub has been widely studied, and is at the basis of many important applications, ranging from analysis of the structure of the Internet to web searches, from peer-to-peer network analysis to social networks, from Viral Marketing to analysis of the Blogosphere, from outbreaks of epidemics to metabolic network analysis [30, 151, 5, 139, 99, 238, 171, 184].

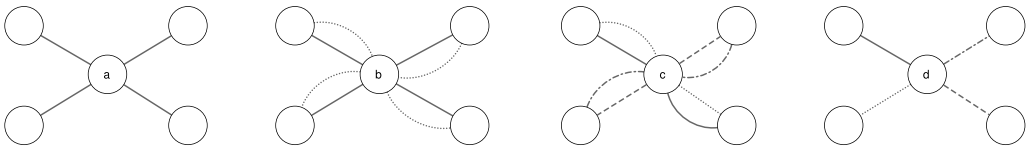


Figure 7.1: Example of different multidimensional hubs.

In [65], the authors analyzed the spread of viruses in real networks, showing that the best nodes to immunize in order to prevent the spread are not necessarily hubs. In social networks, many studies have analyzed the power of highly connected and influential nodes from different points of view: [99, 238, 52] are just a few, describing how having highly connected nodes affects the social behavior of the networks. An interesting study on citation and collaboration networks is presented in [269], where the authors use heterogeneous networks, which can be considered very similar to our multidimensional setting. In communication networks, the authors of [5] showed how to make use of hubs in peer-to-peer networks for fast and efficient searches. In relation to hubs in networks it is impossible not to mention previous approaches like PageRank [208] or HITS [151].

All the previous methods disregard the possibility of enhancing their analysis with the power of a multidimensional investigation, which can be extended in order to consider this more complex scenario. As we have seen, in the real world networks are often multidimesional, i.e there might be multiple connections between any pair of nodes. Therefore, multidimensional analysis is needed to distinguish among different kinds of interactions, or equivalently to look at interactions from different perspectives.

In this section, we propose to deal with the following question to show the usefulness of the Dimension Relevance class of measures: *how does the concept of hub change in multidimensional network analysis?* Figure 7.1 depicts a possible hub in a monodimensional network (Figure 7.1a) and three possible hubs in a multidimensional setting (Figure 7.1b-d). The four cases show different hub configurations: while the first is simply a node with a high degree (thus connectivity), and nothing else can really be said about it purely on the basis of this figure, the other three represent a different scenario. We can see that the hub in Figure 7.1b is connected by two dimensions (solid and dotted line) to all the other nodes, while this is not true for the other hubs. Neither the degree of the node nor the number of neighbors that could be reached from it would give us any more information. The third and fourth case give other two possible scenarios where, if we take into account each dimension individually, the node in the center has a low degree (and number of neighbors); however, the co-existence of many dimensions where this happens makes it possible to consider the central node as a hub (this is particularly true for the hub in Figure 7.1d).

Can the four hubs be considered in the same way, or can we say something specific about each one? In a multidimensional setting, are all hubs equivalent to each other? Can we say something about the importance of a specific dimension for the connectivity of a node? Finally, can we reason on hubs' behavior by looking at how relevant a dimension is for the connectivity of the hubs?

As these questions suggest, analyzing hubs in multidimensional networks basically introduces a new degree of freedom: the set of dimensions of the network. However, we believe that the current analytical tools are not able to capture the interplay among these dimensions. New measures, that we propose under the class of Dimension Relevance, need to be introduced to overcome this problem.

In this section we then address the problem of finding and analyzing multidimensional hubs in real networks by defining suitable analytical tools. We introduce two analytical tools needed in order to perform such an analysis. Firstly we need a multidimensional generalization of the degree, different from the one proposed in Section 6.1, namely the number of neighbors of a node, in Section 7.1.1. Secondly we will define the brand new class of measures, the Dimension Relevance in Section 7.1.2. The aim of these measures is to exploit the additional degree of freedom that multidimensionality adds to the problem of analyzing hubs in networks. Finally, in Section 7.1.3 we show a multidimensional hub analysis case study on the proposed real world networks, supporting

the meaningfulness of the problem introduced, the effectiveness of the measures defined, and a few practical applications intended to demonstrate the power of our approach.

The most important insights of this section are: (1) we show that multidimensional hubs exist, and can be found and analyzed using our introduced measures of interplay of the different dimensions; (2) we show that the characterization of multidimensional hubs highlights interesting analytical properties, and (3) thanks to our measures, we discover and quantify the importance of every single dimension with respect to the others, generally unknown a priori.

7.1.1 Neighbors

Now, we define the *Neighbors* class of measures, needed for the creation of the analytical ground of the Dimension Relevance measures. Neighbors is an extension of the degree in the multidimensional setting. We also give its interpretation and we show a toy example illustrating its behavior for a few nodes.

In classical graph theory the *Degree* of a node refers to the connections of a node in a network: it is defined, in fact, as the number of edges adjacent to a node. In a simple graph, each edge is the sole connection to an adjacent node. In multidimensional networks the degree of a node (i.e., the number of the connections of that node in a network) and the number of nodes adjacent to it are no longer related, since there may be more than one edge between any two nodes. For instance, in Figure 7.1, all nodes have four neighbors, but they have a very different degree, especially in every single dimension.

In order to capture this difference, we define a measure concerning the *neighbors* of a node.

Definition 14 (Neighbors) *Let $v \in V$ and $D' \subseteq D$ be a node and a set of dimensions of a network $G = (V, E, D)$, respectively. The function $Neighbors : V \times \mathcal{P}(d) \to \mathbb{N}$ is defined as*

$$Neighbors(v, D') = |NeighborSet(v, D')|$$

where $NeighborSet(v, D') = \{u \in V \mid \exists (u, v, d) \in E \ \wedge \ d \in D'\}$. This function computes the number of all the nodes directly reachable from node v by edges labeled with dimensions belonging to D'. □

Note that, in the monodimensional case, the value of this measure corresponds to the degree. It is easy to see that $Neighbors(v, D') \leq Degree(v)$, but we can also easily say something about the ratio $\frac{Neighbors(v,D')}{Degree(v)}$. When the number of neighbors is small, but each one is connected by many edges to v, we have low values for this ratio, which means that the set of dimensions is somehow redundant with respect to the connectivity of that node. This is the case of node 2 in the toy example illustrated in Figure 6.1. On the opposite extreme, the two measures coincide, and this ratio is equal to 1, which means that each dimension in which v has a neighbor is necessary (and not redundant) for the connectivity of that node: removing any of these dimensions would disconnect (directly) that node from some of its neighbors. This is the case of node 5 in Figure 7.2.

We also define a variant of the Neighbors function, which takes into account only the adjacent nodes that are connected by edges belonging exclusively to a given set of dimensions.

Definition 15 (Neighbors$_{XOR}$) *Let $v \in V$ and $D' \subseteq D$ be a node and a set of dimensions of a network $G = (V, E, D)$, respectively. The function $Neighbors_{XOR} : V \times \mathcal{P}(D) \to \mathbb{N}$ is defined as*

$$Neighbors_{XOR}(v, D') = |\{u \in V | \ \exists d \in D' : (u, v, d) \in E \wedge \nexists d' \notin D' : (u, v, d') \in E\}|$$

It computes the number of neighboring nodes connected by edges belonging exclusively to dimensions in D'. □

7.1.2 Dimension Relevance

As already mentioned, while performing hub analysis it is important to understand how important a particular dimension is over the others for the connectivity of a node, i.e. what happens to the connectivity of the node if we remove that dimension. In order to answer these questions, we define the new concept of *Dimension Relevance*. Also in this case, we refer to the toy example in Figure 7.2 for its intuition and interpretation.

Definition 16 (Dimension Relevance) *Let $v \in V$ and $d \in D$ be a node and a dimension of a network $G = (V, E, D)$, respectively. The function $DimRelevance : V \times D \to [0, 1]$ is defined as*

$$DimRelevance(v, d) = \frac{Neighbors(v, d)}{Neighbors(v, D)}$$

and computes the ratio between the neighbors of a node v connected by edges labeled with a specific dimension d and the total number of its neighbors. □

Clearly, the above function can be defined taking into account a set of dimensions instead of a single dimension. In other words, we can generalize Definition 16 as follows:

Definition 17 (Dimension Relevance) *Let $v \in V$ and $D' \subseteq D$ be a node and a set of dimensions of a network $G = (V, E, D)$, respectively. The function $DimRelevance : V \times \mathcal{P}(D) \to [0, 1]$ is defined as*

$$DimRelevance(v, D') = \frac{Neighbors(v, D')}{Neighbors(v, D)}$$

and computes the ratio between the neighbors of a node v connected by edges belonging to a specific set of dimensions in D' and the total number of its neighbors. □

Note that, the case of a single dimension (Definition 16) is a particular case of that in Definition 17, where the set of dimensions D contains only the dimension d. In the remaining of the paper we define the others measures considering a set of dimensions.

However, in a multidimensional setting, this measure may still not cover important information about the connectivity of a node. Figure 7.1 shows three nodes (a, b and c) with a high dimension relevance for the dimension represented by a solid line. In the first two cases the dimension relevance is equal to one, but the complete set of connections they present is different: if we remove the solid line dimension the node a will be completely disconnected while the node b can still reach all its neighbors. To capture these possible different cases we introduce a variant of this metric.

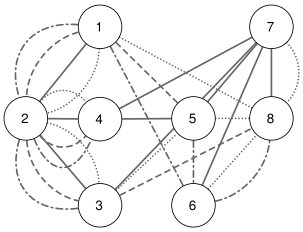
Definition 18 (Dimension Relevance XOR) *Let $v \in V$ and $D' \subseteq D$ be a node and a set of dimensions of a network $G = (V, E, D)$, respectively.*
The function $DimRelevance_{XOR} : V \times \mathcal{P}(D) \to [0, 1]$ defined as

$$DimRelevance_{XOR}(v, D') = \frac{Neighbors_{XOR}(v, D')}{Neighbors(v, D)}$$

computes the fraction of neighbors directly reachable from node v following edges belonging only to dimensions D. □

We can easily calculate the above metric in the examples in Figure 7.1. For the node a there is no difference with the *Dimension Relevance* (Definition 17): all its neighbors are only reachable by solid edges. In node b we have the opposite situation: all its neighbors are reachable by solid edges, but we always have an alternative edge. So the *Dimension Relevance XOR* of the solid line dimension is equal to zero.
In the following, we want to capture the intuitive intermediate value, i.e. the number of neighbors reachable through a dimension, taking into account all the possible alternatives.

Definition 19 (Weighted Dimension Relevance)
Let $v \in V$ and $D' \subseteq D$ be a node and a set of dimensions of a network $G = (V, E, D)$, respectively. The function $DimRelevance_W : V \times \mathcal{P}(D) \to [0, 1]$, called Weighted Dimension Relevance*, is defined as*



Id	Deg	Neigh	DR				DR_W				DR_{XOR}			
			dim1	dim2	dim3	dim4	dim1	dim2	dim3	dim4	dim1	dim2	dim3	dim4
1	7	4	0.250	0.500	0.500	0.500	0.062	0.312	0.312	0.312	0.000	0.250	0.250	0.250
2	12	3	1.000	1.000	1.000	1.000	0.250	0.250	0.250	0.250	0.000	0.000	0.000	0.000
3	7	4	0.250	0.250	0.750	0.250	0.312	0.062	0.562	0.062	0.250	0.000	0.500	0.000
4	7	4	0.750	0.250	0.250	0.500	0.562	0.062	0.062	0.312	0.500	0.000	0.000	0.250
5	6	6	0.333	0.166	0.333	0.166	0.333	0.166	0.333	0.166	0.333	0.166	0.333	0.166
6	6	5	0.200	0.000	0.200	0.800	0.200	0.000	0.100	0.700	0.200	0.000	0.000	0.600
7	6	5	1.000	0.000	0.200	0.000	0.900	0.000	0.100	0.000	0.800	0.000	0.000	0.000
8	7	5	0.200	0.000	1.000	0.200	0.100	0.000	0.800	0.100	0.000	0.000	0.600	0.000

Figure 7.2: Toy example and computed measures. Lines: solid = dim 1, dashed = dim 2, dotted = dim 3, dash-dotted = dim 4.

$$DimRelevance_W(v, D') = \frac{\sum_{u \in NeighborSet(v,D')} \frac{n_{uvd}}{n_{uv}}}{Neighbors(v, D)}$$

where: n_{uvd} is the number of dimensions in which there is an edge between two nodes u and v and that belong to D'; n_{uv} is the number of dimensions in which there is an edge between two nodes u and v. □

Hereafter we occasionally use DR to stand for Dimensional Relevance. In our toy example in Figure 7.2, the nodes 6, 7 and 8 have five neighbors, quite a large number in this example, but their values of Dimension Relevance are very different since they are connected in different dimensions.

The Dimension Relevance XOR behaves in a different way. A value equal to zero does not necessary imply that the node is not connected to a particular dimension. It represents a situation where the node has not a single neighbor that can be reached exclusively through that particular dimension. So it is possible to reach it by alternative ways. In Figure 7.2, node 3 is an example of this, when considering the dashed line dimension.

The Weighted Dimension Relevance takes into account both the situations modeled by the previous two definitions. Low values of $DimRelevance_W$ for a particular set of dimensions D are typical of nodes that have a large number of alternative dimensions through which they can reach their neighbors. High values, on the other hand, mean that there are fewer alternatives. Our example shows the case of node 4 when considering the solid line dimension: its Weighted Dimension Relevance is clearly the highest, although the dot-dashed line dimension has a high value of Dimension Relevance (as in Definition 17).

The table in Figure 7.2 shows the values of all the above metrics for all the dimensions computed in the toy example. Each value is computed taking into account a single dimension. In our analysis we will apply our metrics on a single dimension to better highlight and show the use, the effects and the power of proposed measures.

The following theorem states the relations among the above three definitions. We state and prove the theorem using only one dimension d, but the proof holds using any set of dimensions $D' \subseteq D$.

Theorem 1 *Let $v \in V$ and $d \in D$ be a node and a set of dimensions in a multidimensional network $G = (V, E, D)$, respectively. It holds:*

$$DimRelevance_{XOR}(v, d) \leq DimRelevance_W(v, d) \leq DimRelevance(v, d).$$

□

Proof In order to prove this theorem it is sufficient to show that

$$Neighbors_{XOR}(v, d) \leq \sum_{u \in NeighborSet(v,d)} \frac{n_{uvd}}{n_{uv}} \quad (1)$$

and

$$\sum_{u \in NeighborSet(v,d)} \frac{n_{uvd}}{n_{uv}} \leq Neighbor(v, d) \quad (2)$$

as $DimRelevance_{XOR}(v, d)$, $DimRelevance_W(v, d)$ and $DimRelevance(v, d)$ have the same denominator. Let:

$A = Neighbors_{XOR}(v, d)$
$B = \sum_{u \in NeighborSet(v,d)} \frac{n_{uvd}}{n_{uv}}$
$C = Neighbors(v, d)$.

First of all, we prove the inequality (1). If node v is connected to a neighbor u only by edges labeled with dimension d then in both A and B, u contributes with 1; if they are connected only by edges labeled with dimensions different than d then in both the formulas, A and B, u contributes with 0; lastly, if they are connected by some edges labeled with dimension in d and some edges labeled with dimensions different than d then in A the node u contributes with a value equal to 0 while in B it contributes with a value greater than 0. Thus, we have that $A \leq B$.

Now, we prove the inequality (2). If node v is connected to a neighbor u only labeled with dimension d then in both the formula B and C it contributes with 1; if they are connected only by edges labeled with dimensions different than d then in A and B u contributes with 0; lastly, if they are connected by some edges labeled with dimensions different than d and some edges labeled with dimension d then in B the node u contributes with a value equal to $\frac{n_{uvd}}{n_{uv}} < 1$ while in C it contributes with 1. Thus, we have that $B \leq C$. □

7.1.3 Finding and Characterizing Hubs

Many interesting network analytic concepts, both at the global and at the local level, such as connectivity, centrality, diameter, etc., developed for standard, monodimensional networks, come under a different light when seen in the multidimensional setting. At the global level, for example, the connectivity of the whole network changes if we see a single dimension as a separate network, with respect to the network formed by all the edges in the entire set of dimensions. Also at the local level, it is possible to analyze many other examples. One such example is the concept of a hub, i.e., a node with a very high degree, substantially higher than the average degree of all nodes. When considering a multidimensional network, such simple concept becomes subtler and multifaceted: first, the definition of a multidimensional hub is parametric with respect to a set of dimensions and secondly, the relevance of a node depends on the interplay among the different dimensions and their impact on the connectivity of the node. Here, a multidimensional hub is a node with high connectivity in the sub-network obtained by considering only the edges from some specified dimensions (we give a formal definition later in this section). As evidence of how subtle the characterization of a multidimensional hub is, we found in all our real-world networks that the population of hubs obtained while neglecting the dimensions, differs substantially from that of hubs obtained taking dimensions into account (see Table 7.1 and its discussion later in this section): some (sometimes many) monodimensional hubs are not multidimensional hubs, and vice versa (we provide later on further analysis of this phenomenon). In this section we use mainly the Querylog, Flickr and DBLP-Y networks presented in Chapter 5.

This led us to conclude that analyzing hubs in multidimensional networks is not a trivial extension of the standard case. In other words, it requires techniques and measures of node connectivity across different dimensions, able to highlight the interplay among (sets of) dimensions and their impact on node connectivity. Therefore, the problem that we dealt with in this paper can be defined as follows:

Definition 20 (Problem Definition) *Given a large multidimensional network, find and characterize the multidimensional hubs.*

Measuring the Hubbiness

We now formally define the concept of multidimensional hub and a possible characterization for it.

Definition 21 (Multidimensional Hub) *Let v be a node and D the set of dimensions in a multidimensional network. Given a threshold δ the node v is a multidimensional hub iff $Neighbors(v, D) \geq \delta$.*

In general, the threshold δ depends on the specific network, although there are empirical rules in the literature to determine it (one example is the classical 80-20 rule [221]). This is why hereafter we omit this threshold, saying only that a hub is a node with a high number of neighbors.

At this point, one question arises: can we give a formal characterization of multidimensional hubs? The set of measures to assess the relevance of a dimension for a given node allows to characterize some kinds of hubs. In particular, by combining the two following notions of multidimensional hub and relevance of a dimension for a node we are able to identify, within a set of multidimensional hubs, those for which a specific dimension d is relevant (Definition 22) or irrelevant (Definition 23).

Definition 22 (D-supported Hub) *Let $v \in V$ and $D' \subseteq D$ be a node and a set of dimensions of a network $G = (V, E, D)$, respectively. The node v is D-supported if v is a multidimensional hub with respect to the set of network dimensions D and $R(v, D') \geq \epsilon$, with $R \in \{DR,\ DR_{XOR},\ DR_W\}$*

Definition 23 (D-unsupported Hub) *Let $v \in V$ and $D' \subseteq D$ be a node and a set of dimensions of a network $G = (V, E, D)$, respectively. The node v is D-unsupported if v is a multidimensional hub with respect to a set of dimensions D and $R(v, D') < \epsilon$, with $R \in \{DR,\ DR_{XOR},\ DR_W\}$*

As one can see, the difference between the two resides only in the direction of the inequality. They are equivalently "powerful" nodes of the network, as they are hubs thus very highly connected, but they have a totally opposed connection patterns w.r.t the dimensions of the network. We choose to use the term *nemesis* to address them, and we use this term hereafter to refer to hubs that play the opposite role of other ones (fixing D', if v_1 is a D-supported Hub for the set of dimensions D' and v_2 is a D-unsupported Hub for the same set of dimensions, then v_1 and v_2 are the nemesis of each other). An interesting future work is to have a multidimensional definition for "date" and "party" hubs [7], and to study the relationship between this four classes (i.e. are date hubs significantly more represented in the D-unsupported class for the entire set of dimensions D? Are the classes completely orthogonal?).

There are two *caveats* in the above definitions. First, the definitions are generic for any set of dimensions D', where D' might even contain only a single dimension. When analyzing real networks, a specific target of analysis might be to find the set of d-supported hubs for one single specific dimension d.

Second, the choice among the various DRs allows to find D-supported (D-unsupported) hubs with very different multidimensional characteristics. The choice is ad-hoc, and only depends on the analysis that one might want to perform, hence there is no better choice among the others. For example, by choosing the DR_{XOR}, and looking for the d-unsupported hubs for a specific dimension d, we are looking for hubs that would be hubs even without the connections provided by dimension d.

Note that the above characterization in a network whose set of dimensions D would contain only a single dimension d, would not make any sense, for the following reasons: (1) all the values for the DRs would be 1, making the distinction between D-supported and D-unsupported vain, and (2) there would be no distinction among the three DRs, making thus the characterization leading to only one possible type of hubs, which is, obviously, the traditional concept of monodimensional hub.

Given all the above, building a multidimensional analysis aimed at extracting and characterizing a multidimensional hub is relatively easy: the analyst defines the desired analysis, translates it in terms of a filter on the values of Dimension Relevance and then selects, among the nodes with high number of neighbors, the ones satisfying the filter, leading to D-supported or D-unsupported hubs, according to the most appropriate choice of DR and parameters.

Example 6 (Airline Network) *Without looking at the complete structure of the multidimensional network of airlines (each airline company taken as a dimension), we selected two European multidimensional hubs ($\geq$ 100 connected cities): Dublin and Madrid. We found that the Ryanair airline has a DR of 0.54 for Dublin and 0.27 for Madrid, while it has a DR_{XOR} of 0.31 for the former, and 0.09 for the latter. This means that, while the Ryanair's importance seems to be*

double for Dublin w.r.t Madrid in terms of connected cities, its importance as sole connection is more than triple. Dublin is then a Ryanair-supported hub, according to both DR and DR_{XOR}.

Computing the Dimension Relevance

The complexity required to compute the Dimension Relevance measure set is low. The procedure we used is the following. First, we are considering an undirected network, thus the edge set E is a sequence of triplets (u, v, d), where each element is a numerical is for nodes and dimensions, and $u < v$, i.e. the numerical id of the first node is always lower than the second node (no self loops allowed).

We then sort $|E|$, with a sub-quadratic complexity of $\mathcal{O}(|E| \log |E|)$. All the Dimension Relevance variants can be now computed by a simple scan of the sorted edge list, with complexity $\mathcal{O}(|E|)$, that is dominated by the sorting complexity (in case the edge list is already sorted, to calculate the Dimension relevance is linear). When cycling over the edge list, the following criteria are used to update the values of the Dimension Relevance measures. When we found the first triple (u, v, d), we increase by one the $Neighbors(u, D)$ and $Neighbors(v, D)$ values. For each triple (u, v, d), we also update the $Neighbors(u, d)$ and $Neighbors(v, d)$ values, needed for the plain Dimension Relevance, and the n_{uvd} and n_{uv} values, needed for the Weighted Dimension Relevance. We then update $NeighborsXOR(u, d)$ and $NeighborsXOR(v, d)$, needed for the Dimension Relevance XOR, if and only if the triple (u, v, d) is the only one connecting nodes u and v (and we ave this information since all the triplets involving directly u and v are clustered together, due to the sorting).

We now want to answer the following:

Q1. Are the presented multidimensional measures able to make important latent knowledge emerge from the data?

Q2. Would it be possible to extract (part of) this knowledge with non-multidimensional techniques with the same degree of complexity?

Q3. What kind of knowledge would the measures make emerge on null models?

We now provide an answer for Q1 and Q2. Q3 is a particular case, requiring the definition of multidimensional null models and/or generators for complex networks. Since this is an interesting problem per se, we address its solution in a following section (Section 8.2). For clarity, we report here the main findings: more sophisticated multidimensional network generators are able to better represent the complex dynamics of the distribution of the Dimension Relevance class of measures. However, significant differences still emerge between the most advanced null models and the real world networks, proving that Dimension Relevance measures are able to unveil complex dynamics at the local level that are not fully understandable with assumptions at the global level (we refer in particular to the comparison of DR distributions from the original networks in Figures 7.3(a-i) and from the null models in Figures 8.7, 8.8, 8.9 and 8.10).

Q1: Multidimensional Measures on Real Networks

Here, we want to study the power of our multidimensional tools in letting latent knowledge emerge from the data. Figures 6.2(a)-(c) show, for the three datasets, the cumulative neighbor distributions in log-log scale. Consider the curve corresponding to the global network, i.e. the distribution of neighbors computed over all the dimensions. The DBLP-Y network shows a behavior similar to the "the rich gets richer", with very different cut-offs, while the other networks behave differently. The figures show that the behavior of this measure resembles the one of the degree in the monodimensional setting, even without being completely similar. To support this, in Figure 6.2(a)-(c) we report also the cumulative neighbor distribution per dimension (which, in turn, is the degree per dimension) of the three networks, and we compare them with the global neighbors distribution. For DBLP-Y, we chose only six representative dimensions out of the original 65.

Network	Multi → Mono	Mono → Multi
QueryLog	75.69%	99.85%
Flickr	70.87%	46.43%
DBLP-Y	31.08%	70.87%

Table 7.1: Relationship between mono and multidimensional hubbiness of a node

In Figure 7.3(a-i) we report the distributions of Dimension Relevance in the three dataset. The strong differences among the three networks highlight the presence, in the real world, of networks with different multidimensional structure.

We then believe that the three DRs are able to make the interplay among the dimensions emerge from the data, extracting the knowledge at the center of investigation in Q1, that we now consider successfully answered.

Q2: Finding multidimensional hubs with monodimensional techniques

In order to answer the question "can we extract multidimensional hubs with monodimensional techniques?", the first question to answer is "are multidimensional hub necessarily monodimensional and vice versa?"

Table 7.1 answers this for our three networks. For each dataset we extracted the top 20% monodimensional hubs (nodes with a high degree in one dimension) and the 20% multidimensional hubs (only taking into account the total number of neighbors considering all the dimensions). The columns of the table report the probability of being a multidimensional hub given that a node is a monodimensional hub and vice versa. We can see from DBLP-Y and Flickr dataset that being a monodimensional hub does not entail being a multidimensional hub and vice versa.

However, one can argue that finding 46% of multidimensional hubs by extracting monodimensional hubs could be sufficient. To prove that this is not true, we show that two multidimensional hubs may look very different when their multidimensional connectivity is examined, or, in other words: the fact that two hubs are multidimensional does not entail that these two nodes have the same importance and show the same behavior. This is based on the intuition that, in the multidimensional setting, two different multidimensional hubs may exhibit a different interplay among the dimensions in which they appear. In order to show this, we report in Figure 7.3(m-o) the cumulative standard deviation of the three measures for each hub on the different dimensions. The high values of the standard deviation obtained highlight a high diversity of relevance for each of the dimensions in which a node is connected. All the networks show high values of these metrics for a large fraction of nodes. As a result, two multidimensional hubs may look very differently when their multidimensional connectivity is examined.

Consider Figure 7.4. Here we report the size of the overlap among two sets of hubs: the ones extracted with our filters defined later in this section for our analysis and the ones having only a high monodimensional degree. Note that the set of hubs extracted in our analysis here is a subset of the total set of multidimensional hubs. Therefore the set of nodes used for Figure 7.4 highly differs from the one used for Table 7.1. The overlap between the two sets is computed after increasing the number of hubs extracted from the network. We started extracting the 0.25% of high degree nodes and we ended extracting the 2.5% top hubs. The plot highlights two different things. The first is related to Flickr and QueryLog datasets. In these datasets it is fairly impossible to extract the desired set of hubs, answering to our precise analytical questions expressed later in this section, without any multidimensional information. In order to extract less than 1% of the nodes with the desired multidimensional properties, the analyst must extract the 2.5% of the network's hubs. This means, for example, that in order to obtain 7 hubs in the QueryLog dataset the analyst has to extract 5000 hubs and for 200 Flickr hubs this number raises up to 30000. Furthermore there is no way to distinguish the desired hubs from the other ones. The DBLP-Y dataset behaves differently. In DBLP-Y we can obtain almost all (99%) the interesting hubs defined according to our analytical questions by extracting the 1.5% of the hubs of the network (9000 nodes). However, this ratio decreases as we enlarge the set of hubs extracted. This happens because 8774 is the exact number of nodes in DBLP-Y having the desired characteristics. Thus they are not hubs: we are

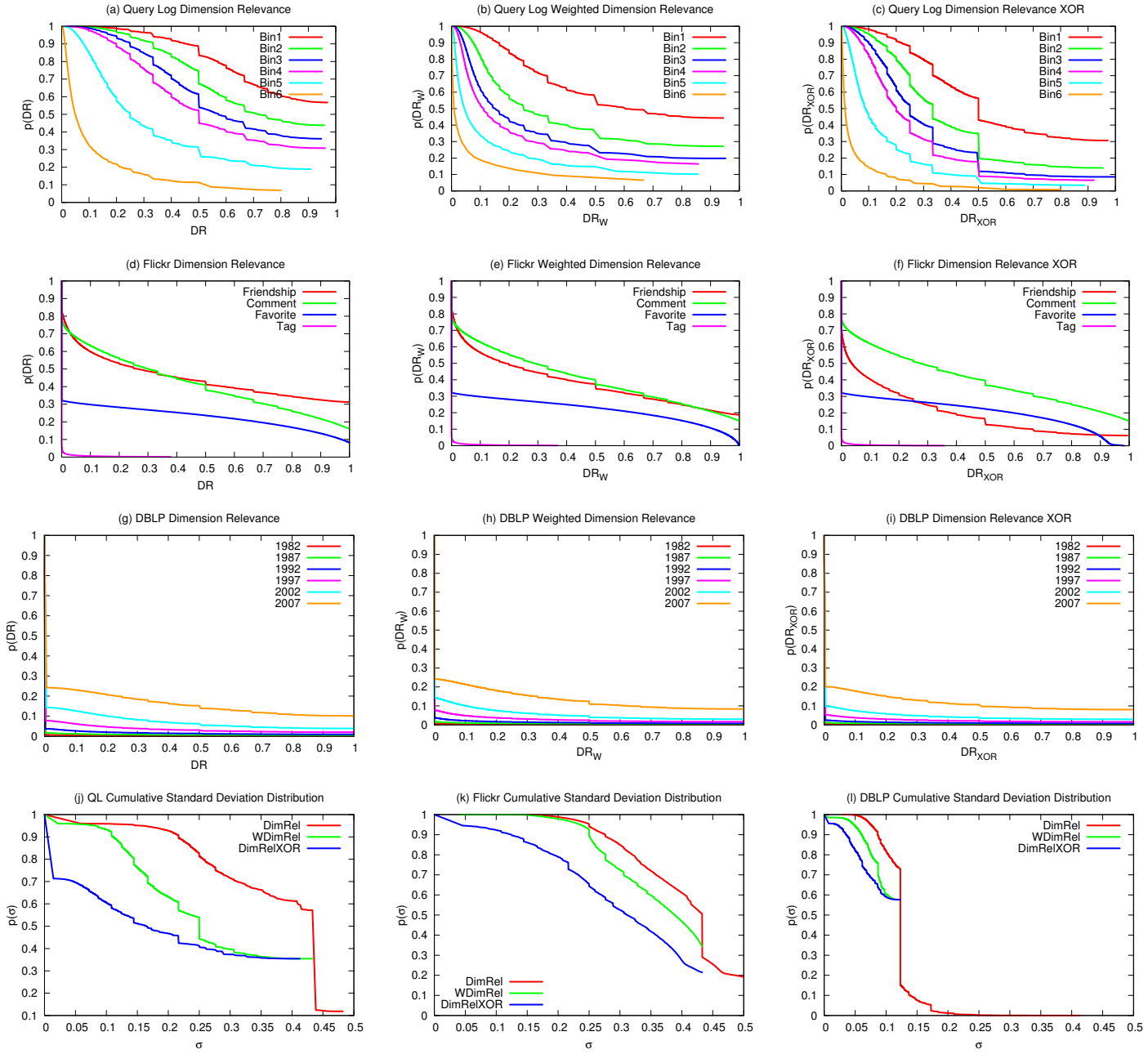


Figure 7.3: The metrics computed on the three networks (color image).

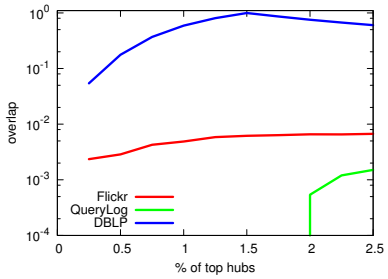
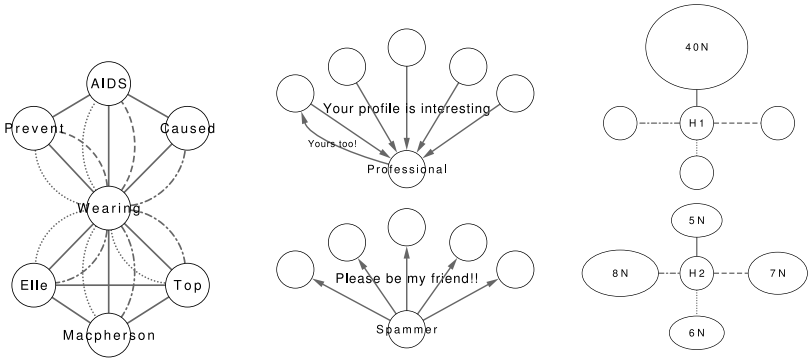


Figure 7.4: The overlap ratio between monodimensional and multidimensional hubs



(a) (b) (c)

Figure 7.5: Some of the multidimensional hubs extracted

dealing with all the nodes, regardless their connectivity. For this analysis it is a coincidence that all these nodes are also monodimensional hubs, but, as one can expect, this is not always true.

In conclusion, we have provided a motivated answer for question **Q2**, that makes it clear the need for these multidimensional techniques.

Hub Characterization in Real Networks

We now show how, by exploiting the characteristics and the semantic of the real networks described in Chapter 5 and of their dimensions, we are able to assign a name to some of the possible characterizations of the hubs. We then find hubs that are interesting w.r.t three simple analytical examples. In our networks, we found convenient to use the dimension relevances as a powerful filter for characterizing a narrow set of hubs, due to the distributions of these measures. In particular, in QueryLog, only 100 hubs have a Weighted Dimension Relevance lower than 0.25 or higher than 0.5. The vast majority of hubs lays on a very narrow interval of values, thus becoming clearly irrelevant for the analysis, more focused on the outliers. This holds also for the other two networks.

The following three examples are meant to be only a sample of possible real-life applications in which our techniques may be helpful. In the future, we plan to expand the direction of finding interesting real-life problems in multidimensional network analysis, in which our techniques might be used as a support for a more complete understanding of real phenomena. Just to give an example of this, we will very briefly present also the *nemesis* of our extracted hubs, i.e. hubs with very similar number of neighbors, but extracted with a specular filter on the Dimension Relevance. This will help to better characterize the extracted hubs and will give a further idea of the degree of freedom of the analyst in using these analytical tools.

In the following, we consider hubs the nodes with a high number of neighbors taking into account the complete set of available dimensions, i.e., in definitions 22 and 23, we put $D' = L$.

Detection of Ambiguous Query Terms. In the QueryLog network we applied our measures to find ambiguous query terms. In order to do so, we selected the query terms that are: 1) used in

conjunction with many other terms (high number of neighbors) and 2) generally connected with their neighbors in queries that led to low rank results (low Weighted Dimension Relevance for the first rank bin, i.e. the neighboring terms are often found in queries that do not provide good results for the user).

Then, we are saying that being an ambiguous query term translates into being a D-irrelevant hub, where $D = \{\text{“Bin1”}\}$ and the proper dimension relevance measure is the DR_W. Note this choice: minimizing the DR_{XOR} of dimension “Bin1” would have selected terms that generally do not produce good results at all, while the pure DR would not have specified the interplay with the other dimensions.

Given the hubs extracted with the above characterization, we wanted to go further, trying to explain why the terms led also to good results in a few cases. We then considered the small communities of words surrounding the hubs extracted, where we looked for the reasons for a very good or very bad query result. We selected the neighbors with the highest Dimension Relevance for dimension 1, to see why, with a generally bad query term, sometimes we find good results.

A possible example found to satisfy these criteria is the word “Wearing” (a simplified view of its neighborhood is depicted in Figure 7.5a). This term shows here poor semantics, which needs a disambiguation. Moreover, the clusters surrounding this word are very clear: words in either cluster are not really expected to be in the other one. The first group of queries was apparently generated by users looking for information about AIDS and how to prevent it. In the second cluster we see people interested in Elle MacPherson’s dressing habits.

The nemesis of this hub, i.e. words which always lead to good results with a very high number of other words (D-supported hub, where $D = \{\text{“Bin1”}\}$ using DR_W), are the words “Wikipedia” and “Amazon”: a possible explanation for this is that a user looking either for many different words in an encyclopedia or for products in a store is likely to find the first results to be the best matches.

Outlier Detection. Here we analyzed hubs in a totally different context, i.e. a network of social connections. The aim of this analysis is to find outliers, i.e. users behaving in a strong different fashion than everybody else. In this scenario we are able to present one of the strongest advantages of using a multidimensional network perspective. In a single dimension, we can define an outlier in few different ways (hubs in general, or particularly central or marginal nodes). In a multidimensional network, instead, we can apply those definition for each dimension but, more importantly, we can create brand new definitions. In this case, we define an outlier as a user that connects himself to his neighborhood mainly through one dimension, ignoring almost completely the others. More precisely, we select users that are connected to the network mainly via the Friendship dimension, thus giving less importance to the Comment, Favorite and Tag features of the social network.

Thus, in this analysis we focused on the Dimension Relevance XOR and considered the head of its distribution for the Friendship dimension: high values of this metric mean that the node is connected with its neighborhood exclusively via Friendship links.

Hence, in this analysis, we can characterize as *outliers* the D-relevant hubs, where D={”Friendship”} and the dimension relevance is the DR_{XOR}.

We wanted to go further, by identifying two subcategories of our outliers: *professionals* and *spammers*, for which Figure 7.5b gives a possible representation. The first can be identified due to their high number of ingoing edges and the low number of outgoing ones (to do this, we extracted a posteriori the direction of ever edge, distinguish then between ingoing and outgoing ones). This behavior is classic in social networks: if a person has an interesting profile, many people will ask for friendship. We found two instances of this kind of profile[1,2]. On the other hand, the owner of an interesting profile could not be interested in having so many friends. The opposite observation can be made for spammers: they can be detected by a high number of outgoing edges but no one is interested in returning the friendship link to a spammer (we found two examples of these hubs[3,4]).

[1] http://www.flickr.com/photos/38687875@N00
[2] http://www.flickr.com/photos/20532904@N00
[3] http://www.flickr.com/photos/10539246@N05
[4] http://www.flickr.com/photos/23941584@N08

As nemesis (D-unsupported hubs, where D={ "Friendship"} and using DR_{XOR}), we found three profiles[5,6,7]. All these profile presented, at the time of the download of the network, a very high number of neighbors and no one exclusively through the Friendship dimension: at the moment of writing this paper, all the profiles are closed. Therefore, the nemesis of both *spammers* and *professionals* are the *quitters* (and this is really interesting in the perspective of the service providers).

Analyzing Temporal Behaviors. In this section, we go beyond the theory presented so far. Consider definitions 22 and 23. It is clear that real networks might express rich semantic, and that even powerful tools and characterizations as defined so far could not cover the complete set of analysis that one might want to perform. In this perspective, we want to show how, by substituting the usage of the DRs in the two mentioned definitions with any of the possible aggregates computed on their values, it is possible to expand the class of phenomena that can be studied with our tools.

In this context, an interesting application of our approach is to analyze the temporal behavior of multidimensional hubs on evolving networks. In this section we show the results obtained on DBLP-Y, whose dimensions are the years of publications. The specific object of our analysis is to find authors of scientific papers who tend to change the authors with whom they collaborate possibly every year. Note that we are not focusing on just new collaborations, but we want also to see the old ones to disappear. In order to do so, we found hubs v maximizing the number of dimensions d for which $\mathrm{DR}_{XOR}(v, d) > 0$ (maximizing this value means maximizing the number of years in which the author had collaborations that took place only in a specific year and not in others).

In this scenario then, we call then *dynamic researchers* the D-relevant hubs v, where $D = L$ (where L contains all the years) and, instead of the any of the simple DRs, we maximize $|\{d : DR_{XOR}(v, d) > 0\}|$.

Figure 7.5c reports two representations of hubs extracted in this way: the hubs behaving as H1 and the ones behaving as H2. To be more precise, a deeper classification among them might be performed by looking also at the standard deviation of the DR_{XOR} computed in all the dimensions. The example H2 in the right of that Figure, in fact, represents a hub minimizing the standard deviation. H1 hubs are collaborators in high effort publications such as books (such as Maxine D. Brown or Steffen Schulze-Kremer); while H2 hubs are authors who tend work with many different people, rarely keeping these collaborations alive, such as Ming Xu or Jakob Nielsen.

Finally, if we minimize the $\mathrm{DR}_{XOR}(v, d)$ we find the nemesis of these hubs. The list of these hubs includes many relevant names in Computer Science: Allan Borodin, Richard M. Karp, Robert Endre Tarjan, Godfried T. Toussaint, and Jeffrey D. Ullman fall in this category.

To conclude the analysis in this section we would like to sum up our final remarks.

We applied our scalable methodology to large real networks and showed that such hubs do exist and they can be found and studied by using our measures of interplay of the different dimensions. Moreover, our measures allow to discover and quantify the importance of every single dimension above the others.

Many other questions on multidimensional network hubs remain unanswered, and call for further research; we mention two such lines briefly here.

First, we did not consider, in our approach, the possible structure or semantics of the specific set of dimensions under analysis: each different dimension is a distinct categorical value, and used as such in the multidimensional measures; however, such dimension values can be meaningfully sorted (as, e.g., in the QueryLog network, where dimensions are associated to quality levels) or may have a temporal or spatial semantics (as, e.g., in the DBLP-Y network, where dimensions are associated to years). How can our measures be extended to fully exploit this additional structure?

Second, it would be interesting to devise a generalized query framework for the discovery and analysis of hubs in multidimensional networks, based on the proposed measures, capable of supporting the analyst in expressing the desired queries (e.g., top-k hubs according to some

[5]http://www.flickr.com/photos/21700048@N04
[6]http://www.flickr.com/photos/22045276@N00
[7]http://www.flickr.com/photos/53654438@N00

specified hubbiness and relevance constraints), in finding appropriate parameters and thresholds for the involved measures on the basis of the available network data.

7.2 Dimension Correlation

In this section we present a second class of measures that makes sense only in multidimensional networks. An interesting question in multidimensional networks involves the investigation whether a couple or a set of different relations is actually influencing or not each other. In a social network, being friends is probably a strong prerequisite to the communication relation, i.e. sending each other messages. To be able to quantify this "correlation" is our aim. Naturally, this class of measure is called "Dimension Correlation". We firstly provide a general formulation for this class of measures. Then, we are interested in providing a scenario in which Dimension Correlation measures show their analytic usefulness. We chose to avoid social network analysis in its pure sense by using a very particular definition of dimension for our networks. In fact, what we analyze is the temporal evolution of several different evolving networks. In this way, we are able to show that Dimension Correlation, and multidimensional network analysis in general, is able to include in its scope also temporal analysis. In particular, Dimension Correlation is a key element to detect periods in network evolution via a hierarchical clustering technique.

Dimension Correlation class of measures is composed by the following two definitions of correlation. Intuitively, they give an idea of how redundant are two dimensions, if we can expect two nodes to be connected by a given dimension when they are found to be connected by a specific one, and so on. They are based on the Jaccard coefficient, computed between two (sets of) dimensions using in the first case the sets of nodes (Node Correlation) and in the second case the sets of connected couples (Edge Correlation).

Definition 24 (Node Correlation) *Let $d_1, d_2 \in D$ be two dimensions of a network $G = (V, E, D)$. The* Node Correlation *is the function $\rho_{node} : D \times D \to [0, 1]$ defined as*

$$\rho_{node}(d_1, d_2) = \frac{|V_{d_1} \cap V_{d_2}|}{|V_{d_1} \cup V_{d_2}|}$$

where V_{d_1} and V_{d_2} denote the nodes belonging to dimensions d_1 and d_2, respectively. It computes the ratio of nodes belonging to both the dimensions over the total number of nodes belonging to at least one of them. □

Definition 25 (Edge Correlation) *Let $d_1, d_2 \in D$ be two dimensions of a network $G = (V, E, D)$. The* Edge Correlation *is the function $\rho_{edge} : D \times D \to [0, 1]$ defined as*

$$\rho_{edge}(d_1, d_2) = \frac{|E_{d_1} \cap E_{d_2}|}{|E_{d_1} \cup E_{d_2}|}$$

where E_{d_1} and E_{d_2} denote the edges belonging to dimensions d_1 and d_2, respectively. It computes the ratio of edges belonging to both the dimensions over the total number of edges belonging to at least one of them. □

We now present one of the possible applications of these two measures, namely the detection of eras in evolving networks.

7.2.1 Finding Eras in Evolving networks

We are given an evolving network G, whose evolution is described by a temporally ordered sequence of temporal snapshots $T = \{t_1, t_2, \ldots, t_n\}$, where t_i represents the i-th snapshot. Without any loss of generality, G is represented by a multidimensional network, whose dimensions represent each one a single snapshot. T can be either defined on the sets of nodes, i.e. each snapshot t_i is represented

by the set of nodes involved, or on the sets of edges, i.e. each snapshot is represented by the set of edges in it.

Based on a distance function $f : (t_i, t_{i+1}) \rightarrow]-\infty, +\infty[$, we want to find a hierarchical clustering on T, returning clusters $C_i = \{t_j, \ldots, t_{j+k}\}$, with $j \geq 1$, and $0 \leq k \leq n - j$.

Each cluster represents then an era of evolution. Due to the global evolution of real-life networks, we do allow alterations of the structure of the network among snapshots of the same cluster, as long as they follow a constant trend. As soon as this trend changes, we want to set the corresponding snapshot as the first of a new era, i.e. a *turning point*. The stronger is the change, the higher should be the dissimilarity of that snapshot with the previous one. The definition of the dissimilarity function should reflect this intuition.

We then want to assign to each cluster C_i a set of labels describing the represented era. This step adds a semantic dimension to our framework.

To provide a solution to the general problem presented we need to: (a) define and compute a dissimilarity measure on the temporal snapshots; (b) merge the snapshots into clusters; (c) assign semantics to the clusters based on frequent labels.

(A) Dissimilarity. To perform clustering, the first step is to define a measure of dissimilarity among elements that we want to cluster. As stated before, we use the Dimension Correlation class of measures between each snapshot of the network. In a generic network, we can easily apply both the Node and the Edge Correlation, where each dimension corresponds to a temporal snapshot of the network. The coefficient would then tell us how each snapshot is correlated to the previous one, helping in detecting turning points along the evolution. As we show later in the paper, clustering temporal snapshots actually corresponds to perform a segmentation of the sequence of the snapshots, thus we are interested only in computing the Dimension Correlation for every pair of consecutive snapshots. Note that the Dimension Correlation could be computed between any pair of dimensions, thus corresponding also to non-consecutive snapshots. We are, however, not interested in a two-dimensional clustering of its values, which would lead to eras formed by potentially non-consecutive years; rather we want to perform monodimensional clustering of the temporal evolution of the Dimension Correlation. In the following section we also show how, for the networks we use, this intuition is also supported by the values of the Dimension Correlation: every snapshot is more correlated with its precedent and consecutive ones, than with any other else, justifying eras formed by consecutive snapshots.

Many real-life networks are characterized by a global evolutionary trend, then if we plot the Dimension Correlation for each snapshot, either the Node or the Edge definition, we shall see a global trend, characterized by an almost constant slope of the Dimension Correlation plot, alternated by (moderate to high) changes of this slope. An immediate way to define starting point of new eras is to detect the snapshots corresponding to these changes. This could be done by computing the second derivative of the Dimension Correlation and finding values different from zero. However, the Dimension Correlation is continuous but not derivable exactly in the points we need. To overcome this problem, we consider an approximation of the second derivative defined as follows. We take triples of consecutive years, and trace the segment that has, as endpoints, the Dimension Correlation computed for the first and the third snapshot. If the middle point is distant from the segment, the corresponding snapshot should be considered as the start of a new era. The Euclidean distance between the middle point and the segment also gives us a quantitative analysis of how important is the change: the higher the distance, the stronger the change.

Formally, given a temporal snapshot t_j, we define the following measure:

$$s_N(t_j) = \frac{|c_N(t_j) - (m \times j) - q|}{\sqrt{(1 + (m^2))}}$$

where $m = \frac{c_N(t_{j-1}) - c_N(t_{j+1})}{t_{j-1} - t_{j+1}}$, $q = (-(j+1) \times m) + c_N(t_{j+1})$, and $c_N(t_k) = \frac{|N_{k-1} \cap N_k|}{|N_{k-1} \cup N_k|}$ is the Node Correlation.

Defining s_E, which is the counterpart computed on the set of edges, requires to consider c_E instead of c_N, where c_E is the Edge Correlation.

However, this measure takes, formally, only one snapshot as input, thus it is not intuitive to use as basis for a clustering methodology. In order to tackle this problem, we define a dissimilarity between any two snapshots as follows.

Definition 26 (Era Clustering Distance Function) *Given an ordered sequence $t_1, t_2, \ldots, t_n$ of temporal snapshots of a network G, the distance function between any two snapshots t_i and t_j computed on their node sets (f_N) is defined as*

$$f_N(t_i, t_j) = \begin{cases} s_N(t_{max(i,j)}) & if\ |i-j| = 1 \\ undefined & otherwise \end{cases}$$

Defining the similarity on the edges f_E requires to consider s_E instead of s_N.

Moreover, this dissimilarity measure allows for a straightforward hierarchical clustering: an higher dissimilarity corresponds to a stronger separation between two consecutive eras. This means that by setting a fixed threshold, we can draw a dendrogram of the hierarchical clustering, driven by this dissimilarity as a criterion for merging two consecutive clusters in a bigger one. Note that the hierarchy among clusters permits to analyze the eras with a different granularity, allowing different sensibility of the framework to the changes of the network structure.

(B) Hierarchical clustering. Having defined a measure of dissimilarity, we are now ready to group together our snapshots into clusters, starting from single-member ones, and then merging, driven by increasing values of dissimilarity.

In hierarchical clustering, when merging clusters, there are various main approaches followed in the literature to define the distance between two clusters: the maximum distance between any two points belonging to the two clusters (complete linkage), the minimum (single linkage), the average (average linkage), the sum of all the intra-cluster variance, and so on.

Given two clusters $C_i = \{t_1, t_2, \ldots, t_k\}$ and $C_j = \{t_{k+1}, t_{k+2}, \ldots, t_{k+p}\}$, in order to define the distance between two clusters, we shall first compute all the distances between every pair (t_i, t_j), with $1 \leq i \leq k$ and $k+1 \leq j \leq k+p$.

However, according to Definition 26, only one pair of snapshots has a dissimilarity defined: (t_k, t_{k+1}). At this point, we use this dissimilarity as inter-cluster distance. As one can see, taking the only available dissimilarity value as distance between clusters actually corresponds not only to both the complete linkage and the single linkage, but also to the average. In our case, thus, the three of them are identical.

(C) Semantic enrichment of clusters. Once we have computed the cluster hierarchy, we want to add a description of every era. In order to do so, in analogy with the TF-IDF approach used in the Information Retrieval literature [229], we label each cluster with the nodes (or edges, or a property of it), that maximizes the ratio between its relative frequency in that cluster, and its relative frequency in the entire network. This strategy may produce several values equal to 1 (identical numerators and denominators). In order to discern among these cases, we weight the numerator by multiplying it again for the relative frequency in the cluster under analysis. In this way, we give more importance to 1s deriving from nodes (or edges) with a higher number of occurrences in the cluster.

With this frequency based strategy, we are assigning labels that truly characterize each cluster, as each label is particularly relevant in that cluster, but less relevant for the entire network.

One important caveat in this methodology is what to take as label for the edges. In fact, while for the nodes it is straightforward to consider the identity of the corresponding entity of the network as candidate label, the edge expresses a relationship with a semantic meaning, thus each network requires some effort in defining exactly which label could be applied to a cluster computed on edges. For example, in a co-authorship network, where two authors are connected by the papers that they have written together, a possible strategy is to take every keyword in the title of the papers as possible label. In the experimental section we show three different sets of properties used as labels for our networks.

We now discuss the time complexity of our Dimension Correlation based era discovery. The entire framework requires to compute several Jaccard indexes, the dissimilarity measure and the frequencies of the labels. The computation of the Jaccard between two sets A and B requires

$O(|A| + |B|)$. Thus, when computed on the sets of nodes and edges, for each network with $|T|$ snapshots, we have a complexity of $O(\sum_{i=1}^{i<|T|}(|N_i| + |N_{i+1}|) + \sum_{i=1}^{i<|T|}(|E_i| + |E_{i+1}|))$, where N_i is the set of nodes of the i^{th} snapshot, and E_i is the set of edges of the i^{th} snapshot. To this, we have to add $O(2|T|)$ to compute the dissimilarities on both nodes and edges. We then have to add $O(|T| - 1)$ for merging the clusters. Given W the multiset of node and edge labels, we finally have to add $O(|W|)$ to assign labels to clusters. To summarize, for each network, we have a total complexity of

$$O(\sum_{i=1}^{i<|T|}(|N_i| + |N_{i+1}|) + \sum_{i=1}^{i<|T|}(|E_i| + |E_{i+1}|) + 2|T| + |T| - 1 + |W|)$$

$$= O(\sum_{i=1}^{i<|T|}(|N_i| + |N_{i+1}|) + \sum_{i=1}^{i<|T|}(|E_i| + |E_{i+1}|) + |W|)$$

$$= O(|N| + |E| + |W|),$$

where N is the multiset[8] of all the nodes appearing in any of the snapshots and E is the multiset of all the edges appearing in any of the snapshots, which leads to a scalable framework.

7.2.2 Experiments

We make use of three of the presented datasets with a temporal definition of dimensions, namely DBLP-Y, IMDb and GTD. For each of the networks we also built synthetic null models reflecting the global statistics of the network, in terms of number of snapshots and number of edges per snapshot. We created two different null models for each network:
Random. Nodes and edges are placed at random, only the number of nodes and edges of the original network snapshots were preserved.
Preferential attachment. While preserving the number of snapshots and the number of edges per snapshot, each snapshot is created following the preferential attachment model[30], i.e. the probability of connecting two nodes is directly proportional to their degrees.

Dimension Correlation Distribution

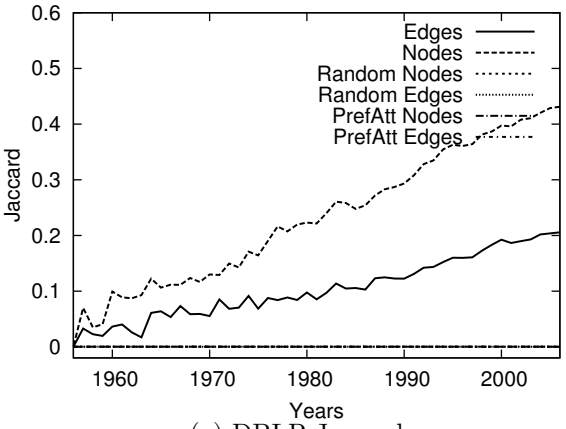
Figures 7.6(a,c,e) show both the Node and Edge Dimension Correlation. These plots report a general increasing behavior of the Dimension Correlation during time in DBLP, both on nodes and on edges, broken by short series of years in which people acted in counter-trend. On the other hand, for the other two networks the temporal behavior seems not to follow a specific trend, while, in particular, GTD presents a hole of two years in the history of the network.

Two questions might be raised on the effectiveness of following a Dimension Correlation-based approach for clustering eras: what would the Dimension Correlation computed on non consecutive snapshot tell us? Are we dealing with some random or real phenomena?

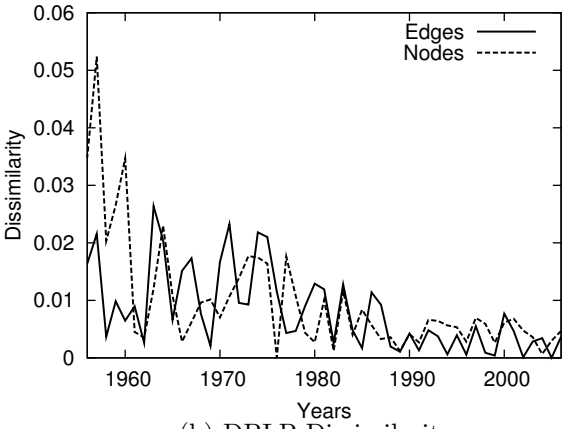
We start answering the first question by plotting the coefficient computed for every pair of snapshots: Figure 7.7 shows that the Dimension Correlation decreases when computed between snapshots more distant in time. As stated in the previous section, this observation justifies a dissimilarity measure that takes into account only consecutive snapshots, as two distant snapshots are not likely to be similar, thus they will belong to different clusters. Temporal segmentation is then a good model for clustering real-life evolving networks, which is a consideration well accepted in the literature regarding evolving networks [38, 174].

Answering the second question requires to compare the knowledge extracted with our methodology on real and random networks. If such knowledge is similar, we might conclude that our methodology is not able to extract any useful, non-random, information. We then followed an approach which is common in the network analysis literature [73]: building random networks as null models and testing the framework on them. In order to do so, we created random and preferential

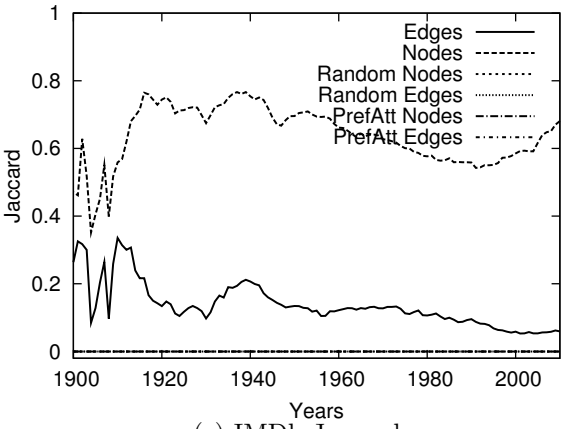
[8] We have multisets because every node or edge can be found in more than one snapshot



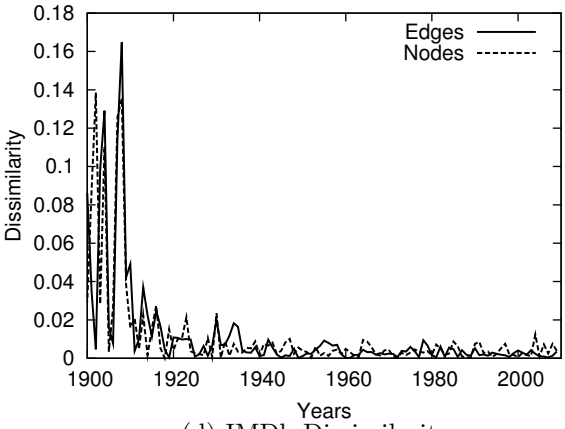
(a) DBLP Jaccard



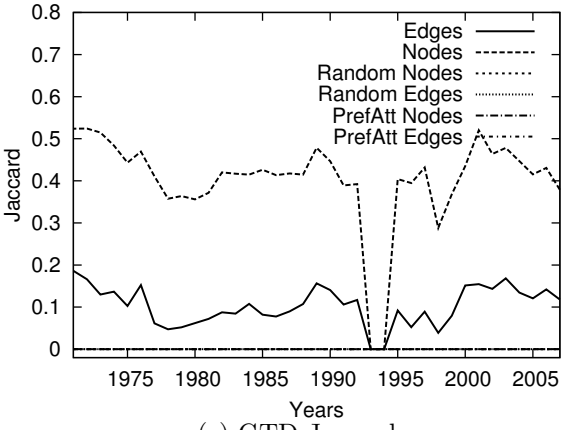
(b) DBLP Dissimilarity



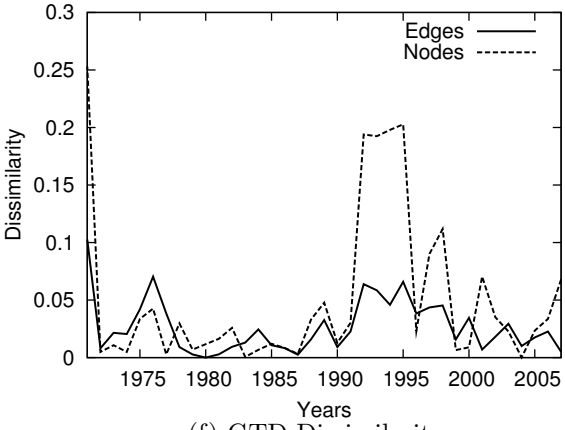
(c) IMDb Jaccard



(d) IMDb Dissimilarity



(e) GTD Jaccard



(f) GTD Dissimilarity

Figure 7.6: The Dimension Correlation computed only between subsequent snapshots (a,c,e) and the corresponding dissimilarities computed on it (b,d,f). We recall that values for the Random and Preferential attachment models are reported but not visible, as they are constant on the 0 line.

attachment null models, as stated at the beginning of this section, and computed the Dimension Correlation on them. As we see in figures 7.6(a,c,e), the random component of both the null models makes the framework not meaningful on them (all the Dimension Correlations for the null models are zero), and shows that the defined random models are not an accurate description of the dynamics of Dimension Correlation in real world networks.

The second step of the framework requires to compute our dissimilarity on the basis of the

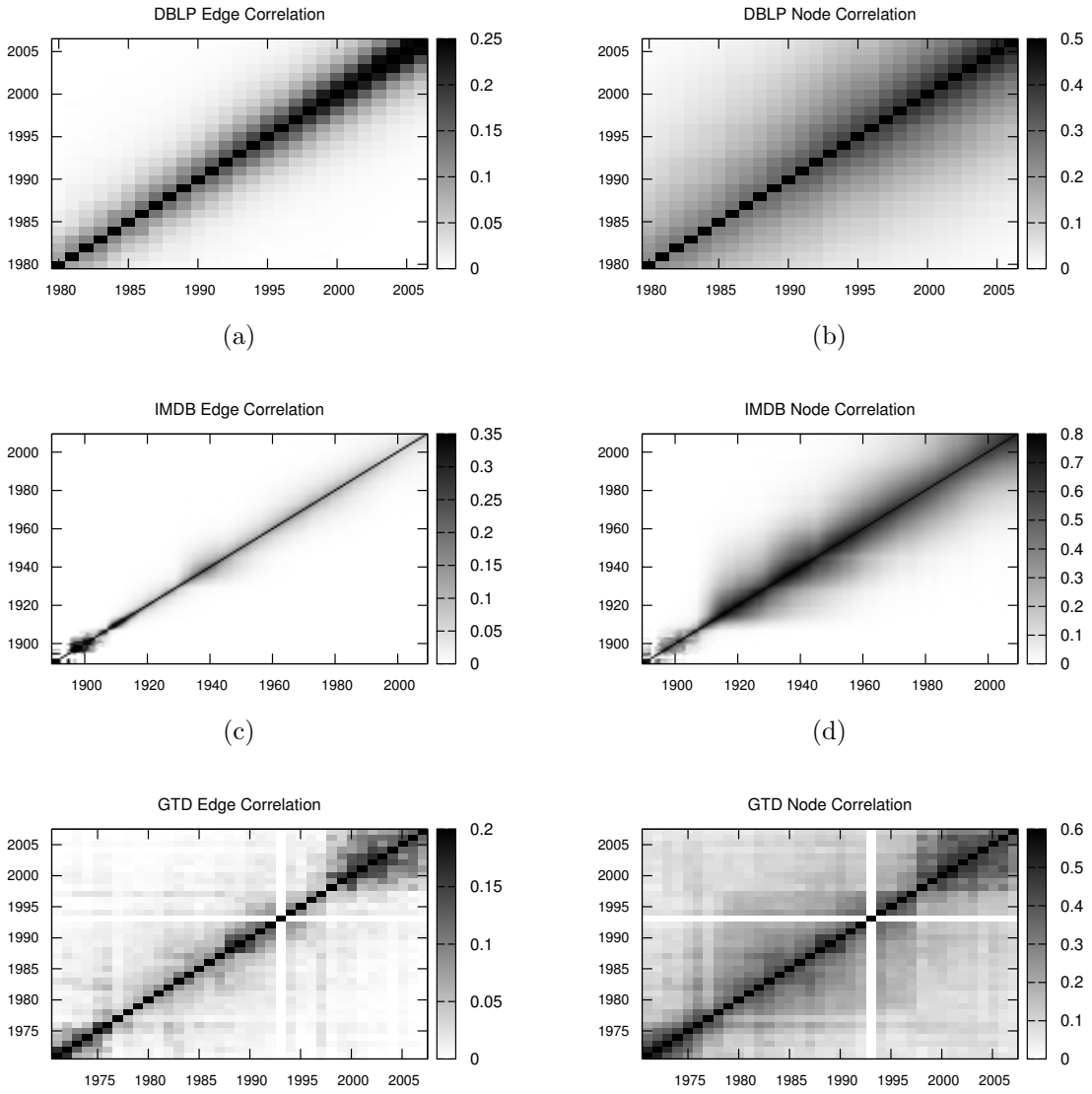
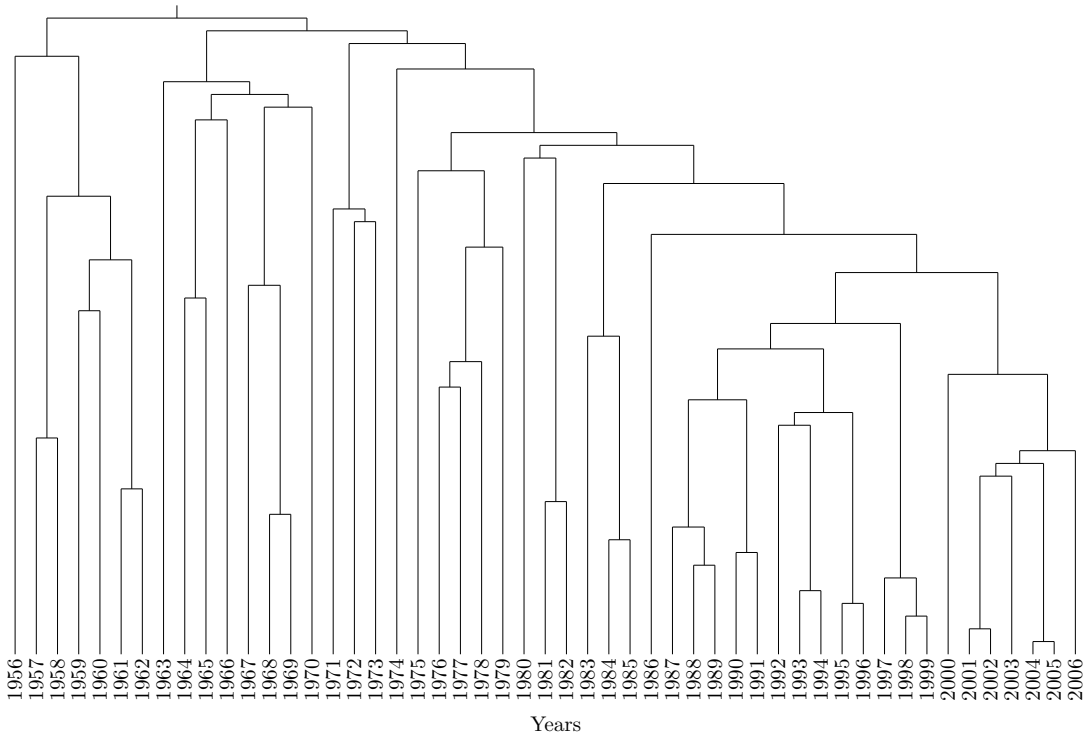


Figure 7.7: The Node and Edge Correlation computed among all the snapshots.

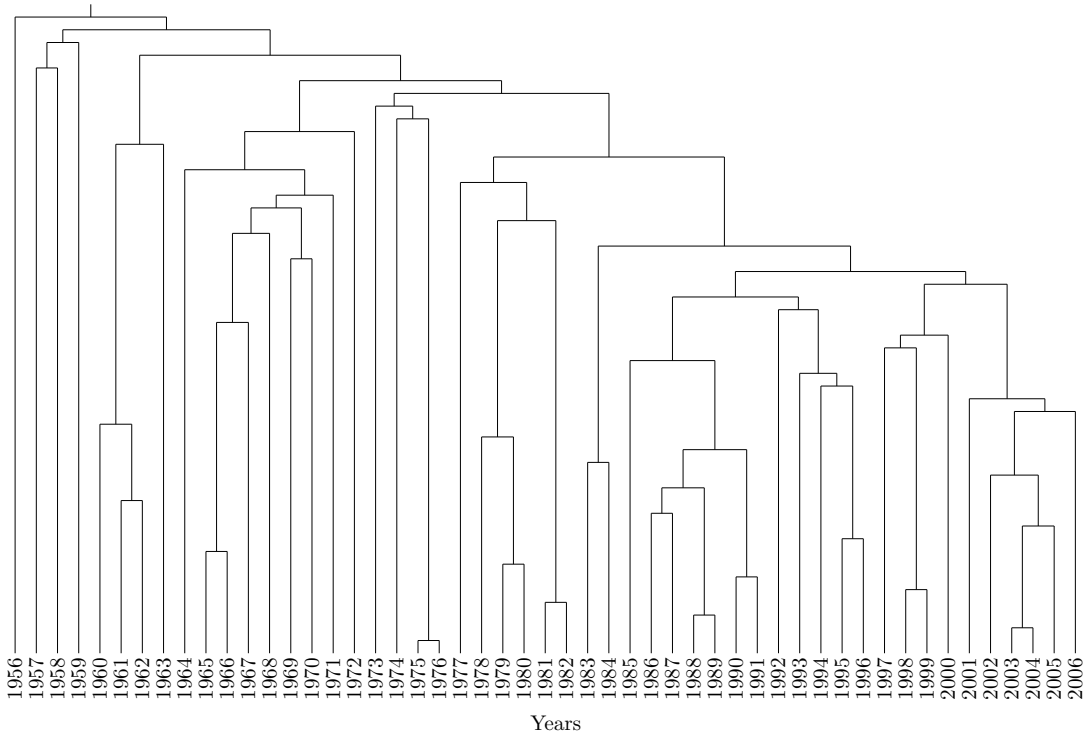
Dimension Correlation coefficient computed on the network. Figures 7.6(b,d,f) report the values of the Era Clustering Distance Function (Definition 26), both in the edge and node definitions, for each network. As one can see, the quantitative analysis of our dissimilarity measure is effective: its values have a considerable standard deviation. That is, we can effectively perform hierarchical clustering finding a well distributed strength of starting snapshots for new eras of evolution.

Another observation that can be done is that while the Dimension Correlation values computed on nodes or edges show similar trends, stronger differences can be found in the dissimilarity plots. That is, we expect the eras computed on nodes slightly differ from the ones computed on the edges.

As last note, we see that in the first years, although not always noticeable from the Dimension Correlation plots, the dissimilarity spots very unstable behavior. This could be mainly explained by two considerations: first, at the beginning of the history of every network, the network structure is still very little, and a change of a few nodes or edges may result in a strong change of the Dimension Correlation values; second, even though a network follows one clear model of evolution (DBLP is well known to follow the preferential attachment model [38, 55]), the model itself takes a few years to warm up and to be fully functional (note that in the preferential attachment this means that nodes are still not affected by the aging effects).



(a) Edge eras



(b) Node eras

Figure 7.8: Eras on both edge and node evolutions in DBLP

We then started to compute the clusters on the sequences of temporal snapshots. We started from clusters containing only one year and then, driven by the dissimilarity values computed in

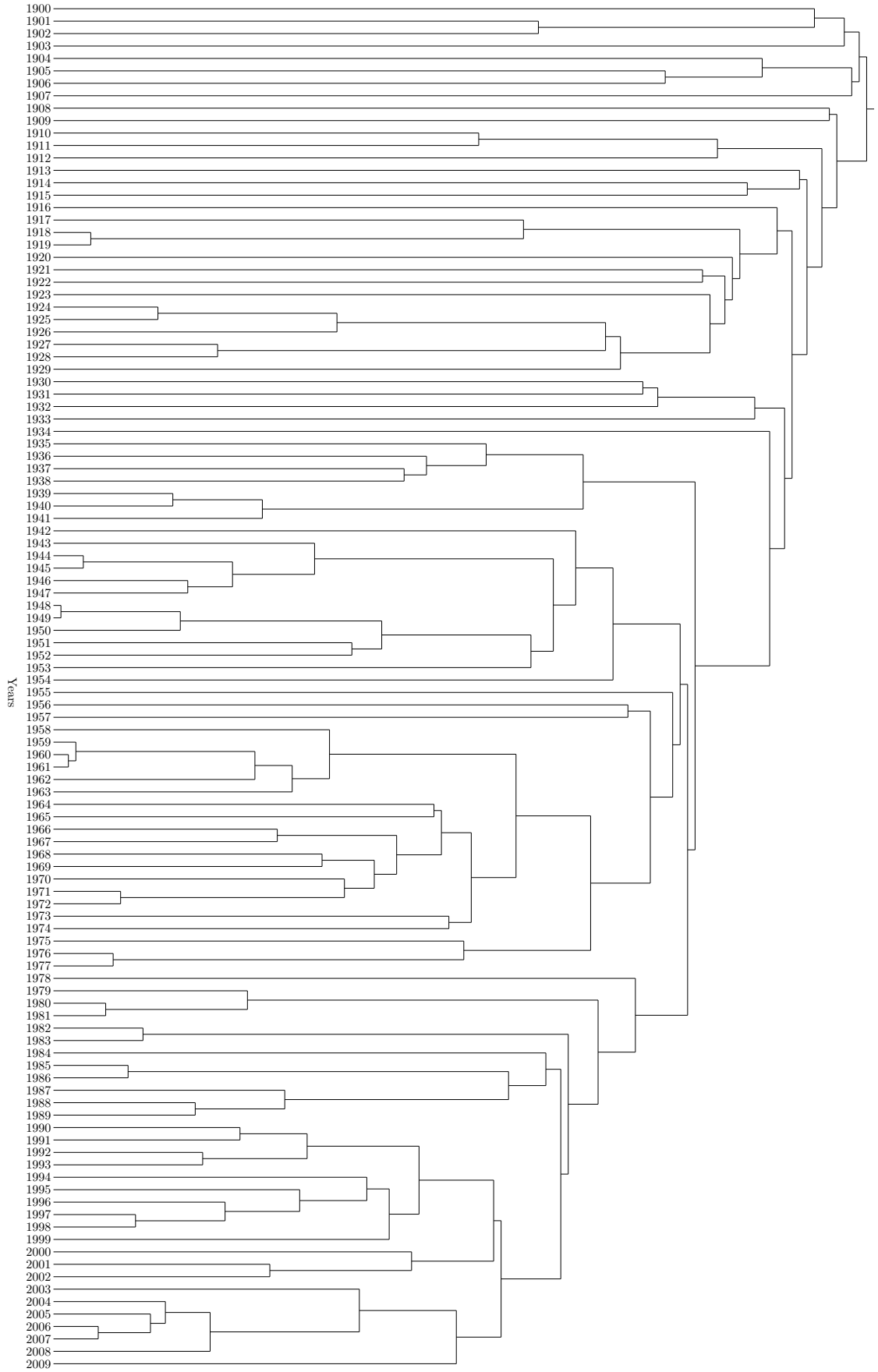


Figure 7.9: Eras in IMDb edge evolution

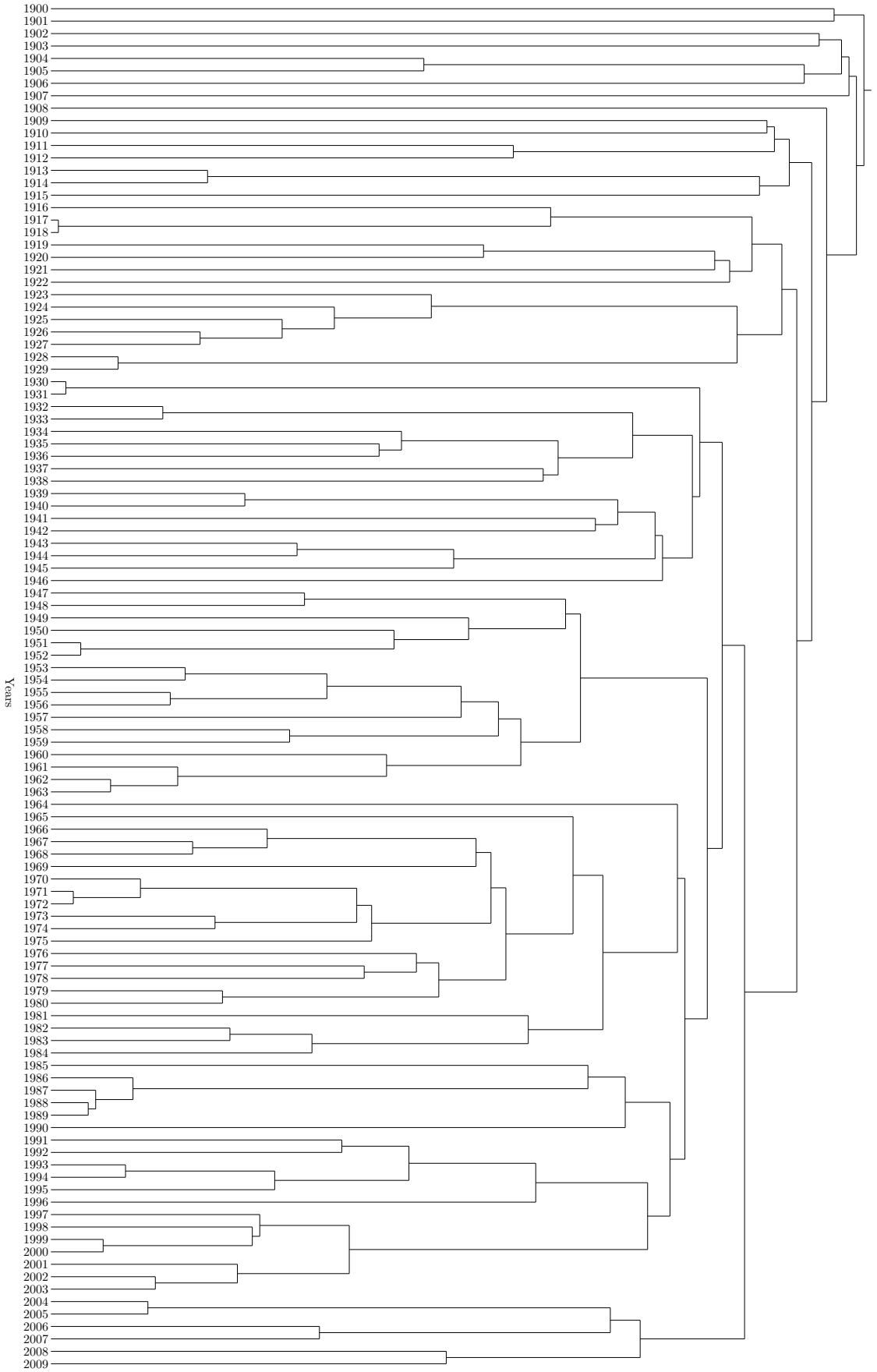


Figure 7.10: Eras in IMDb node evolution

DBLP - Edge labels		
Start	End	Labels
1956	1962	tunnel diode, q-d-algorithm, megabits-sec, four megacycles, bounded transition
1963	1970	prediscuss, algol, machine to man, ssdl, tree manipulation
1971	1973	lr0, word functional, optimal, virtualize, syntax analysis
1975	1979	data, language, program, computer, codasyl
1980	1982	pascal, language, database, data, micro-computer
1983	1985	prolog, database, online, abstract, expert
1987	1991	parallel, program, logic, abstract, database
1992	1996	parallel, program, logic, object oriented, computer
1997	1999	model, parallel, design, distributed, image
2001	2003	model, data, network, design, image
2004	2005	network, model, algorithm, web, data

DBLP - Node labels		
Start	End	Labels
1957	1959	Yu. A. Shreider, I. Y. Akushsky, Howard H. Aiken, D. G. Hays, W. L. van der Poel
1960	1963	Calvin C. Elgot, W. D. Frazer, Roger E. Levien, Robert O. Winder, Lorenzo Calabi
1964	1972	R. L. Beurle, Sheila A. Greibach, Rina S. Cohen, Karl K. Pingle, James L. Parker
1973	1976	Raymond F. Boyce, Michael Ian Shamos, Matthew M. Geller, Louis Pouzin, Irving L. Traiger
1977	1982	Peter Raulefs, Gary G. Hendrix, Helmut K. Berg, Nathan Goodman, S. Bing Yao
1983	1984	Hans Bekic, Gunter Spur, Werner Frey, Frank-Lothar Krause, Ashok K. Thareja
1985	1991	Walter Ameling, Ehud Y. Shapiro, David Chaum, Setrag Khoshafian, David W. Stemple
1992	1996	Robert K. Brayton, Alberto L. Sangiovanni-Vincentelli, Terence C. Fogarty, Janak H. Patel, Martin Kummer
1997	2000	Miodrag Potkonjak, Bruce Schneier, Christopher J. Taylor, Alok N. Choudhary, Prithviraj Banerjee
2001	2006	Mahmut T. Kandemir, Zhaohui Wu, HongJiang Zhang, Wei-Ying Ma, Wen Gao

Table 7.2: Era labels on both DBLP edges and nodes

the previous step, we merged similar consecutive clusters, with increasing values of dissimilarity.

Figures 7.8,7.9,7.10,7.11 report all the dendrograms of the extracted eras for each network. On the x axis we ordered the yearly snapshots of our networks. On the y axis we connect two snapshots, or era clusters, at a height proportional to their distance. The higher the connection, the more distant are the eras. Note that, due to the large number of snapshots and to the wide range of values taken by the dissimilarity, we could not plot the dendrograms with the height proportional to the dissimilarity values itself, but rather we just connected in sequence the eras with an increasing dissimilarity at regular intervals.

A few considerations can be done by looking at the dendrograms. First, in all the three networks, as expected by looking at the dissimilarity plots, the first years tend to form eras by themselves, and this is true both for nodes and for edges.

Second, while, as we said above, the Dimension Correlation plots of nodes and edges for each network tend to look similar, and the differences are then emphasized in the dissimilarity plots, by looking at the shape of the dendrograms, discerning between eras that coincides for both nodes and edges, and eras that include different years for the two sets, appears to be easier. For example, look at the years 2001-2006 in both DBLP nodes and edges: it is easier to see those years grouped in the same era in the dendrograms in Figure 7.8 than in the dissimilarity plot in Figure 7.6(b). Same discussion for the eras 1995-1997 and 1998-2007 in the GTD network, that are both similar when comparing nodes and edges, but for which the dissimilarity plot does not clearly reflect this situation.

Third, while the dendrograms can spot situations as above, they can also highlight differences in the node and edge evolutions. Take, for example, years 1930-2009 in both nodes and edges in

IMDb, as reported in Figure 7.9 and 7.10. This era presents very different sub-eras when looking at nodes or edges, and this is because of the different importance, given during time, to new nodes or new edges over old ones.

As last step in our framework, we computed the labels for each cluster obtained. We recall that for each cluster C_i we assign the set of the k labels maximizing the ratio between their frequency in C_i and their frequency in the entire network. Tables 7.2, 7.3 and 7.4 report a few of the most characterizing labels for some of the eras of each network. Due to the impressive number of total eras and labels, we could not report all the labels for all the eras, but instead we chose, for each network a selection of interesting eras (covering the entire network history), and a selection of the most representative labels for them. The DBLP keywords were pre-processed using the Porter's stemming algorithm [224].

We chose some relatively small eras in order to cover approximately the entire time span of the dataset. The start and end years of an era were selected where the inclusion of the following, or preceding, year would have caused the merging of two eras resulting in a selected period of many years not strongly correlated each other, according to the dendrogram. Verifying the labels of the extracted eras provides two benefits: it is useful to evaluate our results, as we refer to fields in which there is a ground truth about periods, and may lead to novel points of view about the history of our data sources.

For DBLP, we present the labels for the node and edge eras in Table 7.2. It is possible to spot some interesting eras, such as the ALGOL era from 1963 (the year of one major revision of ALGOL60[9]) to 1970. In the 70s many popular programming languages were developed, such as C (developed from 1969 to 1973[10]), Prolog (which was born in 1972 from a project aimed not at producing a programming language but at processing natural languages [76]) and Pascal (standardized in 1983[11], and this might explain also its era from 1980 to 1982).

Interesting enough, from 2004 we are witnessing a brand new era, made of networks and the increasing complexity of web technologies. Node era labels for DBLP let emerge some other key research results: we can recognize the huge work made by David Chaum (1985-1991) in the field of cryptography, the basis of the electronic currency system, culminating in 1990 with the foundation of his electronic cash company; another example is Raymond F. Boyce, a key researcher for the development of SQL [69], died in 1974.

For IMDb, we present the era labels in Table 7.3. It is possible to perform an analysis at two different granularity levels. At a high level, one may notice that the keywords for periods before 1975 are very specific and referring to precise concepts in movie history (such as the sound synchronized to record, referring to the very first movies with sound, or heimatfilm, such as "Lassie come home"), while after 1975 keywords are simpler and less specific (love, death, murder, blood). This is due to the fact that the keywords are user-assigned, thus very old movies are only watched (and tagged) by a niche of expert cinephiles, while the mass tags recent blockbusters. Note also that the vast majority of IMDb users are Western and particularly American, thus the keywords are heavily unbalanced on Hollywood and European industry, disgreaging other filmographies such as Japan, Hong Kong and the prolific Bollywood. At a lower level of granularity, our technique is able to spot actual eras or sub-eras of movie history, such as the "pre code" era (from 1930, the year in which the Motion Picture Production Code was written, to 1933, when the code become effectively enforced[12]).

In IMDb node eras we see the most prolific people in movie industry. Especially in latter years, counter-intuitively, instead of finding movie stars, which are involved in leading roles in big productions (thus it is impossible for them to participate to more than 4-5 movies a year), we see actors that are prolific in minor roles, or producers (Andreas Schmid, producer from 2004 of movies like "The Punisher", "Lord of war" and "Perfume: The Story of a Murderer", before stopping his career in 2007[13]), directors (Peter Elfelt, very well known for many experimental documentary

[9] http://www.masswerk.at/algol60/report.htm
[10] http://cm.bell-labs.com/cm/cs/who/dmr/chist.html
[11] ISO 7185, http://www.pascal-central.com/iso7185.html
[12] Mick LaSalle, "Complicated Women: Sex and Power in Pre-Code Hollywood"
[13] http://www.imdb.com/name/nm1209077/

IMDb - Edge labels		
Start	End	Labels
1900	1907	spanish-american-war, early-sound, america's-cup, synchronized-to-record, trick-film
1908	1909	synchronized-to-record, film-d'art, william-shakespeare, early-sound, te-deum
1910	1912	trick-photography, broncho-billy, animal-actor, melodrama, law-enforcer
1913	1915	broncho-billy, mister-jarr, universal-ike-series, americana, ham-and-bud-series
1917	1929	melodrama, society, mutt-and-jeff, fable, world-war-one
1930	1933	pre-code, bimbo-the-dog, talkartoon, flip-the-frog, two-reeler
1935	1941	1930s, gunfire, b-movie, beautiful-woman, stock-footage
1942	1954	beautiful-woman, 1940s, usa, world-war-two, series
1956	1957	beautiful-woman, heimatfilm, 1950s, mr-magoo, sportscope
1958	1963	peplum, loopy-de-loop, modern-madcaps, independent-film, nudie-cutie
1964	1965	swifty-and-shorty, beautiful-woman, independent-film, nudie-cutie, peplum
1966	1972	female-nudity, independent-film, spaghetti-western, beautiful-woman, hippie
1973	1974	female-nudity, blaxploitation, hoot-kloot, grindhouse, martial-arts
1975	1977	independent-film, erotic-70s, poliziottesco, italian-sex-comedy, naziploitation
1979	1989	nudity, cult-favorite, murder, electronic-music-score, violence
1990	1993	murder, sequel, male-female-relationship, family-relationships, police
1994	1999	independent-film, female-nudity, gay-interest, love, friendship
2000	2002	independent-film, gay-interest, friendship, female-nudity, flashback
2004	2008	love, death, independent-film, blood, family-relationships

IMDb - Node labels		
Start	End	Labels
1902	1907	Alf Collins, Peter Elfelt, Lucien Nonguet, Arthur Gilbert, Alice Guy
1909	1915	Siegmund Lubin, Arturo Ambrosio, William Nicholas Selig, Pat Powers, David Horsley
1916	1922	John Randolph Bray, Matsunosuke Onoe, Burton Holmes, Bud Fisher, William Randolph Hearst
1923	1929	Abe Stern, Julius Stern, Jack White, Hal Roach, Paul Terry
1930	1931	Arthur Hurley, Leroy Shield, James Mulhauser, Amadee J. Van Beuren, Albert H. Kelley
1932	1938	Edward LeSaint, Earl Dwire, Dennis O'Keefe, Harry Bowen, Fred Parker
1939	1946	John Tyrrell, Emmett Vogan, Cyril Ring, Jack Gardner, John Dilson
1947	1952	Sam Buchwald, Edward Selzer, Stanley Wilson, Izzy Sparber, Marshall Reed
1953	1963	Milt Franklyn, Ahmet Tarik Teke, Nicholas Balla, Seymour Kneitel, Julian Biggs
1966	1975	Sung-il Shin, David H. DePatie, Luigi Antonio Guerra, Adoor Bhasi, Jeong-geun Jeon
1976	1980	Richard Lemieuvre, Cyril Val, Dominique Aveline, John Seeman, Peter Katadotis
1981	1984	George Payne, Herschel Savage, Ilayaraja, Mona Fong, Paul Thomas
1985	1990	Amrish Puri, Lily Y. Monteverde, Yunus Parvez, Shui-Fan Fung, Tony Fajardo
1991	1996	Brahmanandam, Ilayaraja, Floyd Elliott, Milind Chitragupth, Tony Leung Ka Fai
1997	2003	Brahmanandam, Johnny Lever, Phil Hawn, Yiu-Cheung Lai, Simon Lui
2004	2007	Venu Madhav, Brahmanandam, Himesh Reshammiya, Andreas Schmid, Suneel
2008	2009	Kevin MacLeod, Jose Rosete, Suraaj Venjarammoodu, Brian Jerin, Moby

Table 7.3: Era labels on both IMDb edges and nodes

shorts until 1907[14]) and composers. Exceptions to this rule are the extremely prolific Indian stars like Brahmanandam[15], or Hong Kong superstar Tony Leung Ka Fai, who between 1991 and 1995 appeared in many movies of the most important Hong Kong authors such as Tsui Hark, Gordon Chan and Wong Kar Wai.

Finally, consider the eras emerging in GTD dataset, for which we report the labels in Table 7.4. It is interesting to note that the 1977-1983 edge era was dominated by European countries, particularly Italy and France. This period coincides with the years of activity of the Hyperion School, founded in 1976 in Paris and whose members were arrested in 1983. Hyperion is considered linked with many terroristic cells in all Europe, particularly Italy[16], whose activities culminated in 1978 with the kidnapping and assassination of Italian prime minister Aldo Moro by Red Brigades. Also the node era from 1978 to 1981 witnesses the terror war fought in Italy in this period, by two extremist groups of opposite philosophy: the Marxist-Leninist group Prima Linea and the neofascist group Armed Revolutionary Nuclei (NAR). NAR was responsible, among others, of the 1980 bombing of the Bologna main train station[17]; Prima Linea had carried 18 out of their 23 assassinations from 1978 to 1981[18].

It is interesting to note that, among the sets of labels found to be characteristic for an era, there are only a few of them which were somehow "popular". This might seem a problem of the methodology, but it is essentially due to the frequency-based approach. In the future, we plan to investigate the possibility of comparing several different alternatives, based, perhaps, on PageRank, Hits, and other measures of centrality.

7.2.3 Turning points and link prediction

In our problem we do allow evolution within one specific era, while two subsequent eras are characterized by different paces at which the evolution takes place. Building up a model of network evolution is the task at the basis for link prediction, i.e., the problem of deciding, with a certain score, whether two nodes will link in the future [207]. There are several studies regarding link prediction, and most of them rely on a underlying model of network evolution [2, 29, 30, 138, 53, 174, 188, 142]. However, not all the models fit all the different types of networks, and most predictors perform well on certain networks, but relatively bad on others. To cope with this, recently the authors of [38, 55] introduced a supervised approach based on extracting *graph evolution rules*, i.e., local frequent subgraphs expressing evolution. The model of evolution itself is there learned from the data, by means of the extraction of those rules, that are afterward used to predict the evolution of the network. In contrast with the previous approaches, this approach allows to predict also *when* then new links will form.

However, to the best of our knowledge, all of the current approaches assume that the model of evolution is static, i.e., there is one rule (Jaccard, Common Neighbors, Adamic-Adar, Forest Fire, and so on) or a set of them (the complete set of rules extracted by GERM), governing the creation of new links, that do not change over time.

This is in contrast with our framework, where we detect moments along the evolution of a network in which the underlying evolution rule changes pace. According to this, we could state that the arrive of a new, sudden, turning point, may invalidate future predictions, as the evolution would change pace.

A question then arises: can we somehow forecast the arrive of a new era? The answer would probably change the way we currently see the link prediction problem, for the reasons we stated above.

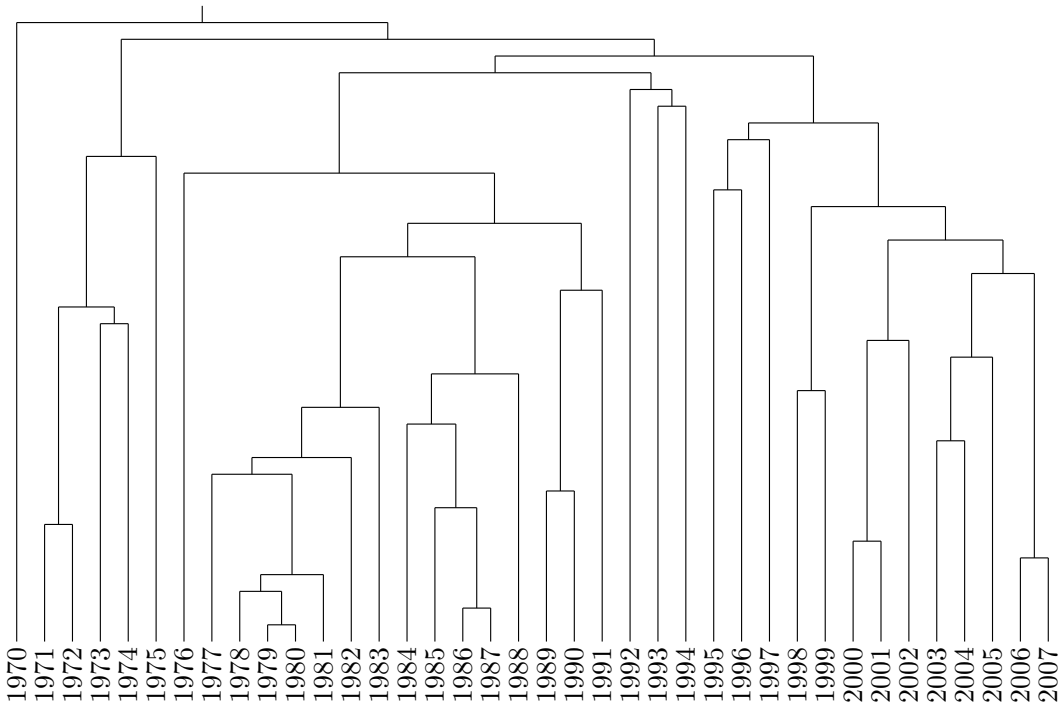
[14] http://www.imdb.com/name/nm0253298/

[15] http://www.imdb.com/name/nm0103977/

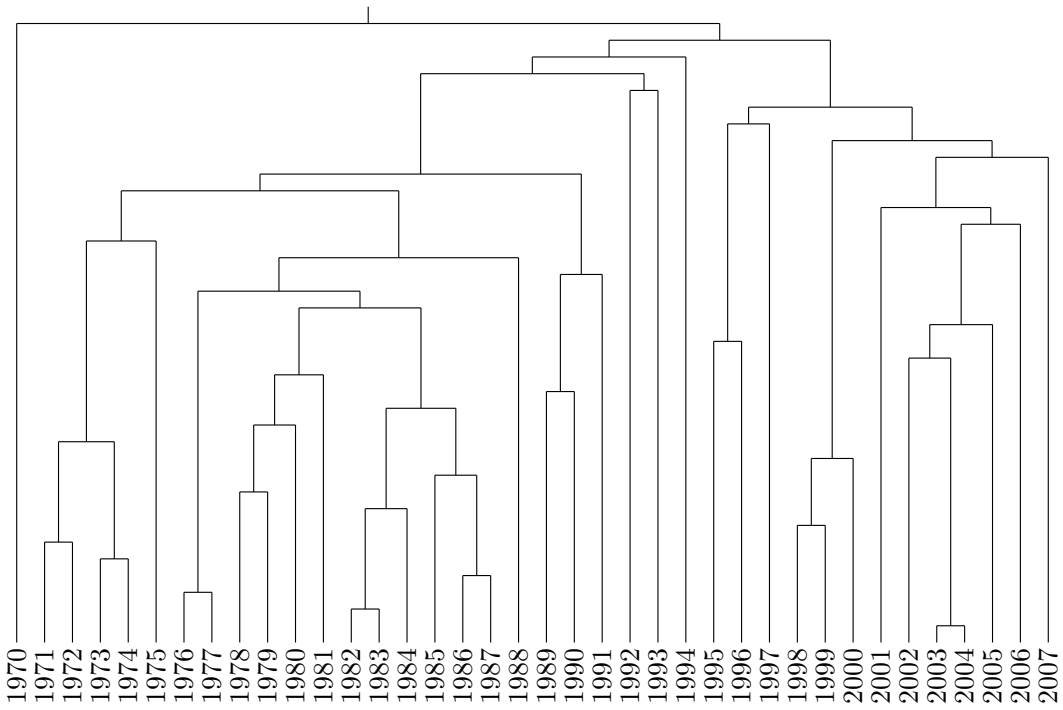
[16] Antonio Ferrari, "In teleselezione dalla Francia gli ordini ai terroristi italiani?", Corriere della Sera 26 aprile 1979

[17] 85 victims, ref. Davies, Peter, Jackson, Paul (2008). "The far right in Europe: an encyclopedia". Greenwood World Press, p. 238

[18] Presidenza della Repubblica, "Per le vittime del terrorismo nellItalia repubblicana: giorno della memoria dedicato alle vittime del terrorismo e delle stragi di tale matrice", 9 maggio 2008 (Rome: Istituto poligrafico e Zecca dello Stato, 2008, ISBN 978-88-240-2868-4)



(a) Edge eras



(b) Node eras

Figure 7.11: Eras on both edge and node evolutions in GTD

GTD - Edge labels		
Start	End	Labels
1971	1975	United States, Northern Ireland, West Germany (FRG), France, Argentina
1977	1983	Italy, France, Spain, El Salvador, Guatemala
1984	1988	Lebanon, Colombia, Sri Lanka, France, Peru
1989	1991	India, Colombia, Israel, Myanmar, Lebanon
1992	1994	India, Bangladesh, Germany, West Bank and Gaza Strip, Venezuela
1995	1997	India, Bangladesh, Pakistan, Indonesia, Colombia
1998	1999	Greece, India, Timor-Leste, Northern Ireland, Kosovo
2000	2002	India, West Bank and Gaza Strip, Israel, Russia, Macedonia
2003	2005	Iraq, India, West Bank and Gaza Strip, Saudi Arabia, Pakistan
2006	2007	Iraq, India, Pakistan, Nigeria, Sudan

GTD - Node labels		
Start	End	Labels
1971	1975	Black September, National Front for the Liberation of Cuba (FLNC), Weatherman, Secret Cuban Government, National Integration Front(FIN)
1976	1977	Communist Combat Unit, Armed Communist Struggle, Baader-Meinhof Group, Black Order, Che Guevara Brigade
1978	1981	Armenian Secret Army for the Liberation of Armenia, Armed Revolutionary Nuclei (NAR), Right-Wing Extremists, Spanish Basque Battalion (BBE), Prima Linea
1982	1987	Armenian Secret Army for the Liberation of Armenia, Abu Nidal Organization (ANO), Anti-terrorist Liberation Group (GAL), M-19 (Movement of April 19), Action Directe
1989	1991	Moslem Janbaz Force, Bhinderanwale Tiger Force of Khalistan (BTHK), Popular Militia (Colombia), Kurdish Dissidents, Death to Bazuqueros
1992	1993	Khasi Students Union, Jharkhand Tribal Forces, Revolutionary Security Apparatus, Allah's Tigers, Ikhwan-ul-Muslimeen
1995	1997	Kuki tribesmen, Jammu and Kashmir Islamic Front, Harkat ul Ansar, Tamil Nadu Liberation Arm, Al Faran
1998	2000	Communist Party of India Marxist-Leninist, Vishwa Hindu Parishad (VHP), Individual, Shiv Sena, Mazdoor Kisan Sangram Samiti (MKSS)
2002	2005	Al-Mansoorian, Kuki Revolutionary Army (KRA), Jaish-e-Mohammad (JeM), Rashtriya Swayamsevak Sangh, Tawhid and Jihad

Table 7.4: Era labels on both GTD edges and nodes

In this section we would like to pose the basis for future work in which we want to solve the link prediction problem taking eras into account. In this section, instead, we try to answer two different questions: do the temporal series formed by our dissimilarities follow any pattern? Is there a way to forecast the subsequent values of the dissimilarities? We will not actually predict eras and formally evaluate the preediction, we rather focus on posing the basis for an era prediction framework.

In the rest of this section we address the above questions, by means of statistical analysis of time series, and, in particular, by means of autoregressive models.

Time series analysis by autoregressive models

An autoregressive model (AR) is a type of random process often used to forecast future values of time series representing natural and social phenomena. The notation $AR(p)$ refers to the autoregressive model of order p, defined as

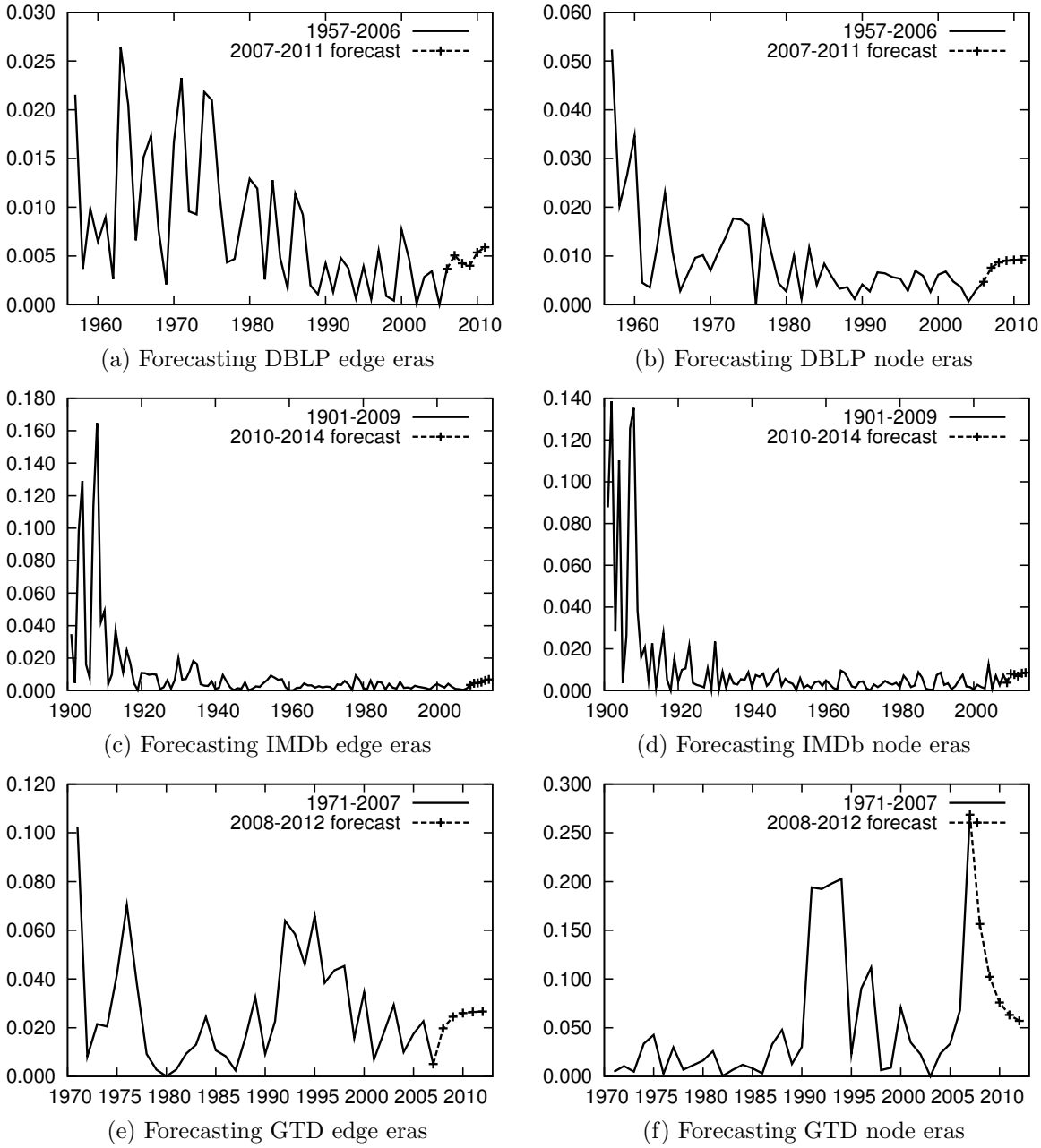
$$X_t = c + \sum_{i=1}^{p} \varphi_i X_{t-i} + \varepsilon_t$$

where $\varphi_1, \ldots, \varphi_p$ are the parameters of the model, c is a constant and ε_t is white noise. Many authors omit the constant for simplicity.

We refer to [57] for a complete introduction to time series analysis, and how to perform prediction on them based on autoregressive models. We used the *tseries* package under the *R* statistical

software[19] to fit autoregressive models on our dissimilarity series, and to perform prediction on them.

Forecasting dissimilarities



(a) Forecasting DBLP edge eras

(b) Forecasting DBLP node eras

(c) Forecasting IMDb edge eras

(d) Forecasting IMDb node eras

(e) Forecasting GTD edge eras

(f) Forecasting GTD node eras

Figure 7.12: Forecasting eras on dissimilarities via autoregressive models

Figure 7.12 reports, for each network, five new values of dissimilarities forecast both on the edges and the nodes. These values were obtained by fitting autoregressive models as explained above, and then using the fits to forecast subsequent values. What we see in the figure is that, while in IMDb the model is forecasting relatively low values of the dissimilarities, this is not true for the other two networks, and in particular for GTD. The intuition behind these plots is that, if

[19] http://www.r-project.org

the forecast values are low, we are not expecting a sudden change of era in the near future, i.e., a link predictor trained on the past evolution of the network, may perform well for the near future. On the other hand, in networks where the forecast values are high, as in GTD - particularly for the nodes - we do expect a new, well distinct, era in the next few years, with this meaning that the results of a link predictor based on the previous history of the network may be not accurate, due to the expected change of evolution pace.

The above might suggest a new way of looking at the link prediction problem, where the basic rules of evolution are supported by a certain confidence in the prediction given also by our temporal analysis of the network evolution. In the future, we plan to investigate the possibility of building such solution for the link prediction problem, based on our clustering framework.

7.3 Dimension Connectivity

We introduce now a small set of measures that analyze the dimension connectivity of the network dimensions, firstly from the point of view of single nodes and then with a higher-level detail.

We start considering two new concepts regarding the nodes of multidimensional networks: *Highest Redundancy Connections (HRC)* and *Lowest Redundancy Connection (LRC)* nodes. They are derived from the combination of the functions Degree and Neighbors. Intuitively, these measures describe the structure around a given node in terms of edge density: if the node is a LRC this structure is sparse, while if the node is HRC it is dense and redundant.

Definition 27 (LRC) *A node $v \in V$ is said to be at* Lowest Redundancy Connection (LRC) *if each of its neighbors is reachable via only one dimension, i.e.,*

$$Degree(v, D) = Neighbors(v, D).$$

□

Definition 28 (HRC) *A node $v \in V$ is called* Highest Redundancy Connections (HRC) *if each of its neighbors is reachable via all the dimensions in the network, i.e.,*

$\forall u \in NeighborSet(v, D) : \ \forall d \in D \ (u, v, d) \in E.$ □

Note that if a node v is HRC we have

$$Degree(v, D) = Neighbors(v, D) \times |D|.$$

Example 7 *In Figure 6.1 we have several LRC nodes: 1, 2, 3, 7, 8 and 9. Some of them appear in both dimensions (2 and 7), while other nodes appear in only one dimension (1, 3, 8 and 9). On the other hand we have only one HRC node: node number 5 is connected via both the dimensions with each of its neighbors.* □

Another interesting quantitative property of multidimensional networks to study is the percentage of nodes or edges contained in a specific dimension or that belong *only* to that dimension. To this end we also introduce: the *Dimension Connectivity* and the *Exclusive Dimension Connectivity* on both the sets of nodes and edges.

Definition 29 (Node Dimension Connectivity) *Let $d \in D$ be a dimension of a network $G = (V, E, D)$. The function $NDC : D \to [0, 1]$ defined as*

$$NDC(d) = \frac{|\{u \in V | \exists v \in V : (u, v, d) \in E\}|}{|V|}$$

computes the ratio of nodes of the network that belong to the dimension d. □

Definition 30 (Edge Dimension Connectivity) *Let $d \in D$ be a dimension of a network $G = (V, E, D)$. The function $EDC : D \to [0, 1]$ defined as*

$$EDC(d) = \frac{|\{(u,v,d)\in E | u,v\in V\}|}{|E|}$$

computes the ratio of edges of the network labeled with the dimension d. □

Definition 31 (Node Exclusive Dimension Connectivity) *Let $d \in D$ be a dimension of a network $G = (V, E, D)$. The function $NEDC : D \to [0, 1]$ defined as*

$$NEDC(d) = \frac{|\{u\in V | \exists v\in V : (u,v,d)\in E \ \wedge \ \forall j\in D, j\neq d : (u,v,j)\notin E\}|}{|\{u\in V | \exists v\in V : \ (u,v,d)\in E\}|}$$

computes the ratio of nodes belonging only to the dimension d. □

Definition 32 (Edge Exclusive Dimension Connectivity) *Let $d \in D$ be a dimension of a network $G = (V, E, D)$. The function $EEDC : D \to [0, 1]$ defined as*

$$EEDC(d) = \frac{|\{(u,v,d)\in E | u,v\in V \ \wedge \ \forall j\in D, j\neq d : \ (u,v,j)\notin E\}|}{|\{(u,v,d)\in E | u,v\in V\}|}$$

computes the ratio of edges between any pair of nodes u and v labeled with the dimension d such that there are no other edges between the same two nodes belonging to other dimensions $j \neq d$. □

Example 8 *In Figure 6.1 the EDC of dimension d_1 is* 0.61 *since it has 8 edges out of the 13 total edges of the network. Its EEDC is equal to* $5/8 = 0.625$. *The NDC for the same dimension d_1 is* 0.88 *(8 nodes out of 9) and its NEDC is* 0.375 *(3 unique nodes out of 8).* □

Table 7.5 presents the values of these measures computed on our real-world networks (for this section, we chose to use Querylog, DBLP-C and DBLP-Y networks).

We now present some results obtained by computing this last set of measures on some of our real world networks, previously introduced. To better understand the meaning of our measures, we also created a random network to be used as null models for our experiments. The network was created at random, while preserving the basic characteristics (number of nodes and number of edges) of each single dimension of the QueryLog network. Thus, we call each of its dimensions with the name of the corresponding dimension in QueryLog, while we refer to the network as Random, or "null model".

What can be seen by looking at the Dimension Connectivity values (especially the EEDC), reported in Table 7.5, is that the measure seems to be correlated with the general trend of Dimension Relevances for the same dataset plotted in Figure 7.3a, b and c. We note, in fact, that the DRs tend to be higher in conjunction with higher Edge Exclusive Dimension Connectivity values (e.g. in the DBLP-Y network, Figure 7.3g, h and i, even if in Table 7.5 for this dataset due to space constraints we report only the last dimensions, the trend is clear). This can be read as: distributions similar to those of the DBLP-Y network occur when the dimensions are quite independent from each other. The QueryLog network presents much more separated distributions among the dimensions where the EEDC values present an high variance. Moreover, the descending order (by dimension) of EEDC follows the decreasing trend (by dimension) in the cumulative distribution plots. This is not surprising. By definition, the two measures are two different perspectives, one local (DR), one global (DC), of the same aspect: how much a dimension is important for the connectivity of a network.

This general tendency of an influence between dimensions can be strengthen by taking a look at the values of Node and Edge Dimension Correlation defined in the previous section. In Figure 7.13 we report the values of the two correlations we defined. We recall that, due to the underlying Jaccard correlation, the matrices shown in Figure 7.13 are symmetric. In these matrices, we reported the correlations computed on each possible pair of dimensions. The values computed on the complete set of dimensions, corresponding to the OCN e OCP percentages, are reported in Table 7.5.

In Figure 7.13 we see that the presence of a natural ordering among the dimensions lets a clear phenomenon emerge: closer dimensions are more similar than distant ones, according to the natural order. The phenomenon is highlighted by the fact that the cells close to the diagonal are darker than those distant from it, in Querylog and DBLP-Y networks. In these two networks, in fact, there

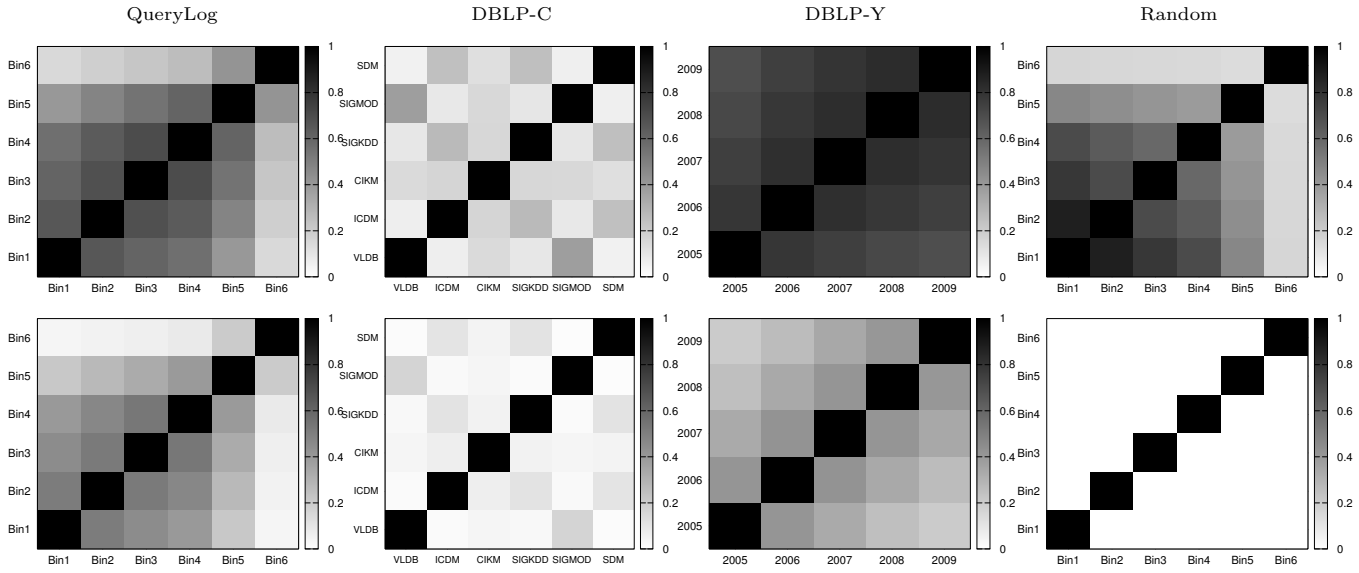


Figure 7.13: The Node and Edge Correlation in our networks.

Network	Dim	NDC	NEDC	EDC	EEDC	HRC	LRC	OCN	OCP
QueryLog	Bin1	75.22%	12.58%	30.98%	38.47%	0.04%	42.47%	3.14%	0.78%
	Bin2	58.69%	4.39%	24.63%	22.39%				
	Bin3	48.39%	2.19%	19.88%	16.30%				
	Bin4	41.05%	1.41%	16.37%	14.05%				
	Bin5	23.24%	0.42%	7.12%	10.72%				
	Bin6	6.62%	0.02%	1.02%	4.45%				
DBLP-C	VLDB	19.28%	0.75%	16.67%	74.75%	0%	79.58%	0.18%	0.01%
	SIGMOD	22.81%	0.97%	21.66%	80.02%				
	CIKM	34.95%	3.86%	22.68%	84.59%				
	SIGKDD	22.58%	1.38%	16.33%	78.68%				
	ICDM	24.38%	2.45%	14.90%	76.24%				
	SDM	13.51%	1.44%	7.76%	68.28%				
DBLP-Y	2005	65.35%	0.50%	16.24%	36.87%	0.35%	9.78%	19.42%	2.83%
	2006	74.69%	0.47%	19.36%	30.90%				
	2007	78.81%	0.47%	21.34%	29.78%				
	2008	78.62%	0.48%	22.10%	33.51%				
	2009	75.01%	0.58%	20.96%	42.33%				
Random	Bin1	75.22%	0%	30.98%	99.97%	0%	99.26%	0.43%	0%
	Bin2	58.69%	0%	24.63%	99.97%				
	Bin3	48.39%	0%	19.88%	99.96%				
	Bin4	41.05%	0%	16.37%	99.96%				
	Bin5	23.24%	0%	7.12%	99.96%				
	Bin6	6.62%	0%	1.02%	99.97%				

Table 7.5: Dimension Connectivity, HRC, LRC, OCN and OCP of our networks.

is a natural order of the years and the bins, used as dimensions. This is not true for DBLP-C: it is not possible to establish a natural ordering among conferences, thus the corresponding matrices in the second column of Figure 7.13 do not have any apparent order like the ones referring to DBLP-Y and Querylog networks.

Consider now the matrices related to the Random network. Due to the random generation, the natural ordering of the dimensions disappears, while, in this case, the size of the dimensions does the difference. Please note that we cannot draw any conclusion from the node correlations for the Random network. They are quite high due to an implementation issue: when generating a dimension for the random network we chose the node set by extracting at random a subset of nodes of the same size of the corresponding Querylog dimension. In this way, the correlation was unfairly increased. Among all the node set generating procedure, we found the one implemented to be the less biased. In any case, the general idea is that more nodes and edges in a dimension imply more correlation with the other ones, by pure chance. The number of possible edges is very large, thus it is difficult to create, using a random generator, the same edges in two different dimensions, dramatically lowering then the Edge Correlation values (which appears almost white in the last column of Figure 7.13) and bringing close to 100% the EEDC values (NEDC values are all equal to zero due to the artifact of choosing the random node ids from the same set).

This is true also considering HRC and OCP values of our networks, reported in Table 7.5. The null model does not present any node with these properties, while instead it has the highest number of LRC nodes. This is again an effect of the above mentioned properties: too many edge combinations lead the edges of a random network to appear only in one dimension. On the other hand, in DBLP-Y we have some authors publishing each year with all their collaborators (HRC column) or at least one time each year (OCN column). These two events are quite rare in the random null model. Some networks may present also situations even more extreme than the random null model: it is the case of DBLP-C in which only 12 authors have published in all the six considered conferences (OCN column), and only two pairs have collaborated at least once in all the conferences (OCP column). But this is natural, since publishing in all of these top conferences is very difficult.

Again, these considerations support the thesis that our multidimensional measures are capturing real, and not random, phenomena, that constitute meaningful knowledge mined in the multidimensional networks analyzed.

Chapter 8

Advanced Analysis

In this chapter we present some classical problem definitions, traditionally tackled in complex network science. We plan to investigate how multidimensionality affects these problem definitions and we propose some solutions, extensions, new methodologies and approaches to take advantage of the additional degree of freedom allowed by this novel setting. The main focus of this chapter will be on community discovery, in Section 2.3. The reason of the choice is due to the incredible amount of literature on this subject, the difficulty of the problem per se and the vast application scenario of community discovery algorithms. Therefore, the community discovery case study is the core of this chapter and of this thesis: firstly we provide a vast analysis of its state of the art and secondly we propose a solution to the problem of finding and characterizing multidimensional communities, solving the multidimensional density ambiguity. In the rest of the chapter, we tackle other classical problems in complex network analysis with a multidimensional perspective: in Section 8.2 we define null models and multidimensional network generators, in Section 8.3 we define the problem of multidimensional link prediction and we analyze how a truly multidimensional algorithm is needed to solve it, finally in Section 8.4 we define the problem of finding the shortest path in a multidimensional network with cost modifiers, we propose a greedy solution and we give the insights for future works in the defined field.

8.1 Multidimensional Community Discovery

As we have seen in Section 2.3, in community discovery the connections among the nodes of a network are posed at the center of investigation, since they play a key role in the study of the network structure, evolution, and behavior. The simplified monodimensional perspective used so far is not suitable to describe the dynamics of communities in the real world, where this perspective is not always enough to model all the available information, especially if the actors are users, with their multiple preferences, their multifaceted behaviors, and their complex interactions. With the aim of better representing these dynamics, in this section we introduce the problem of detecting multidimensional communities of actors in complex networks. As we have stated before, the concept of multidimensional community has to be defined, and we introduce two new measures aimed at analyzing the multidimensional properties of the communities discovered. We then present a framework for finding and characterizing multidimensional communities and we show the results obtained by applying such framework on real-world networks, giving a few examples of interesting multidimensional communities found in different scenarios: co-authorship, movie collaborations and terrorist attacks.

Our main contribution is then: we introduce and formally define the problem of multidimensional community discovery; we introduce two measures for characterizing the communities found; we build up a framework for solving the introduced problem by means of a conjunction of existing techniques and our newly introduced concepts; we perform a case study on real networks, showing a few resulting communities, along with their characterization.

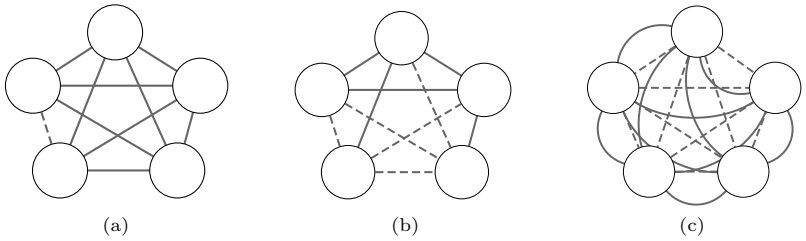


Figure 8.1: Three examples of multidimensional communities

8.1.1 Finding and characterizing multidimensional communities

In this section we define multidimensional communities, two measures aimed at characterizing them, and the problem treated as the main multidimensional case study in this thesis.

Multidimensional Community

Section 2.3 provided a general overview about the literature on community discovery, presenting a large number of diverse definitions of community. Adding multidimensionality to the problem leads to an even more opinable concept of multidimensional community. We start with a high-level possible definition, then we try to add more semantic to it.

Definition 33 (Multidimensional Community) *A* multidimensional community *is a set of nodes densely connected in a multidimensional network.*

As we see, while in a monodimensional network the density of a community refers unambiguously to the ratio between the number of edges among the nodes and the number of all possible edges, the multidimensional setting offers an additional degree of freedom (i.e., the different dimensions). Consider Figure 8.1: in (a) we have a community whose density mostly depends by the connectivity provided by one dimension; in (b) we have a different situation, as both the dimensions are contributing to the density of the community. Also in (c) both dimensions contribute to the density, but clearly in a different way w.r.t (b). Should the three be considered equivalent or can we discern among them? Our aim is to address this question. To do so, we define two measures, γ and ρ, aimed at capturing two different phenomena that can be detected in a community. To compare their values among different networks, we make them take values in $[0, 1]$. Before going to their definition, we introduce some notation used in the rest of this section:

- c is a multidimensional community
- d is a dimension in D
- D_c is the subset of D appearing in c
- P is the set of all pairs (u, v) connected by at least one dimension in the network; $\overline{P} \subseteq P$ is the set of pairs (u, v) connected exclusively by one dimension; $\overline{\overline{P}} = P \setminus \overline{P}$ is the set of pairs connected by at least two dimensions
- P_c is the subset of P appearing in c; $P_{c,d}$ is the set of pairs (u, v) in c connected at least in d and $\overline{P_{c,d}} \subseteq P_{c,d}$ is the set of pairs (u, v) in c connected exclusively in d; $\overline{\overline{P_c}} \subseteq \overline{\overline{P}}$ is the subset of $\overline{\overline{P}}$ containing only pairs in c

Complementarity γ

The first measure, γ, that we call *complementarity*. Intuitively, it should capture the toy example in Figure 8.1(b), where all dimensions are expressed and contribute equally to the density of the community (i.e. the removal of any single dimension will weak the community). Therefore, the

operational definition of a multidimensional community with $\gamma = 1$ is "A community where the removal of any dimension from the network disconnects an equal number of node couples".

We now formally define the concept of complementarity. Following its operational definition, we state that it is composed by the conjunction of three concepts:

- **Variety** $\mathcal{V}_c$: how many different dimensions are detectable among the community c taken as overall;
- **Exclusivity** $\mathcal{E}_c$: how many pairs of nodes are connected exclusively by one dimension among the ones present in c;
- **Homogeneity** $\mathcal{H}_c$: how uniform is the distribution of the number of edges per dimension in c.

We want this measure to be higher when each of the above is high: from the operational definition of γ, variety captures the "removal of *any* dimension from the network" part; exclusivity captures the "disconnects" part; finally homogeneity captures the "equal number of node couples" part. In Figure 8.1(b) if we remove either the solid or the dashed edges we will disconnect an equal number of node couples (namely five), while this does not hold for both Figure 8.1(a) and Figure 8.1(c).

A natural way to achieve this is to aggregate them by their product[1]:

$$\gamma_c = \mathcal{V}_c \times \mathcal{E}_c \times \mathcal{H}_c. \tag{8.1}$$

We now have to define the three concepts. Variety can be computed by

$$\mathcal{V}_c = \frac{|D_c| - 1}{|D| - 1} \tag{8.2}$$

as the number of dimensions expressed with the community c over the total number of dimensions within the network. The two negative terms serve as corrections to make Variety take values in $[0, 1]$. Note that Variety defined as above would be undefined when $|D| = 1$, but this would mean having a monodimensional network, then the use of γ would be meaningless.

Exclusivity can be computed as the ratio between the number of exclusive connections within the community and the total number of connected pairs in c:

$$\mathcal{E}_c = \frac{\sum_{d \in D} |\overline{P_{c,d}}|}{|P_c|}. \tag{8.3}$$

This term is equal to zero when there are no exclusive connections, i.e. every pair of nodes in c is connected by at least two dimensions, while it is equal to one when every pair in c is connected by only one dimension. The formula is not defined for $|P_c| = 0$, which happens only for communities of only one node, for which it has no sense to compute γ.

Finally, we have to define Homogeneity. We want this term to be equal to one when the edges within the community are uniformly distributed among the dimensions represented in c. The simplest way to measure this is to look at the standard deviation of the distribution of the edges in c on the dimensions. We define:

$$\sigma_c = \sqrt{\frac{\sum_{d \in D} \left(|P_{c,d}| - avg_c\right)^2}{|D|}} \tag{8.4}$$

where avg_c is the mean of the distribution, as the standard deviation of the number of edges per dimension in c, and:

$$\sigma_c^{max} = \sqrt{\frac{\left(max(|P_{c,d}|) - 1\right)^2}{2}} \tag{8.5}$$

[1] Although this is not the only possible choice, it is the simplest one.

where $max(|P_{c,d}|)$ is the number of edges belonging to the dimension more represented in c (basically we want to achieve the maximum possible standard deviation). Then, we can define Homogeneity as:

$$\mathcal{H}_c = 1 - \frac{\sigma_c}{\sigma_c^{max}} \tag{8.6}$$

where we subtract the right term from 1, to make $\mathcal{H}_c$ equal to one when the right term is zero, i.e. when the edges are uniformly distributed among the different dimensions.

If we could have the complete set of communities of a network, we could make a more precise estimation of σ_c^{max}:

$$\sigma_c^{max} = \sqrt{\frac{(max(|P_{c,d}|) - min(|P_{c,d}|))^2}{2}} \tag{8.7}$$

where $min(|P_{c,d}|)$ is the number of edges belonging to dimension d in community c where d and c are picked to minimize the number of edges appearing in c labeled with dimension d (i.e. there is no d or c generating a lower $|P_{c,d}|$).

If all the communities presents the same dimensions represented with the same number of edges, i.e. all $|P_{c,d}|$ are equal to the same number, the two normalization coefficients σ_c^{max} would be equal to zero, making the right term of Equation 8.3 undefined. In this case, being the denominator an upper bound, also the numerator would be equal to zero. But this is the ideal topology of a network where the Homogeneity is maximum since all the edges are uniformly distributed, and then we can handle this exceptional case, without lack of generality, by defining $\mathcal{H}_c$ as:

$$\mathcal{H}_c = \begin{cases} 1 & \text{if } \sigma_c = 0 \\ 1 - \dfrac{\sigma_c}{\sigma_c^{max}} & \text{otherwise} \end{cases} \tag{8.8}$$

Example 9 (Multidimensional communities and γ) *Consider Figure 8.1. We see three different multidimensional communities, each of them with different multidimensional structures: in (a), the standard deviation of the number of edges per dimension is the maximum possible, hence $\mathcal{H}_c = 0$, thus $\gamma = 0$; in (b), every term of the complementarity is equal to one, thus $\gamma = 1$; in (c), the exclusivity is zero, as every pair is connected by two dimensions, hence $\gamma = 0$.*

Redundancy ρ

The second measure we define is called *redundancy*, and it captures the phenomenon for which a set of nodes that constitute a community in a dimension tend to constitute a community also in other dimensions. We can see this measure as a simple indicator of the redundancy of the connections: the more dimensions connect each pair of nodes within a community, the higher the redundancy will be.

We can then define the redundancy ρ by counting how many pairs have redundant connections, normalizing by the theoretical maximum:

$$\rho_c = \sum_{(u,v) \in \overline{\overline{P_c}}} \frac{|\{d : (u,v,d) \in E\}|}{|D| \times |P_c|} \tag{8.9}$$

With the help of Figure 8.1 we see how ρ takes values in $[0, 1]$: in 8.1(b), each pair of nodes is connected in only one dimension, then $|\overline{\overline{P_c}}| = 0$ and the numerator is equal to zero; in 8.1(c), all the node pairs are connected in all the dimensions of D, which is equivalent to the number of connected pairs $|P_c|$ multiplied by the number of network dimensions $|D|$ (the denominator), making $\rho = 1$. We see that ρ is undefined for communities formed by one single node, where $|P_c| = 0$ and then the denominator is equal to zero. For this type of communities, however, the redundancy measure is not meaningful, thus we can ignore this case.

Algorithm 1 *MCD_Solver*

Require: $\mathcal{G}, \phi, CD$
Ensure: set of multidimensional communities $\mathcal{C}$ and sets of their charaterization S_γ, S_ρ

1: $G \leftarrow \phi(\mathcal{G})$
2: $C \leftarrow CD(G)$
3: **for all** $c' \in C$ **do**
4: $\quad c \leftarrow \phi'(c')$
5: $\quad \mathcal{C} \leftarrow \mathcal{C} \cup c$
6: $\quad S_\rho \leftarrow S_\rho \cup \rho(c)$
7: $\quad S_\gamma \leftarrow S_\gamma \cup \gamma(c)$
8: **end for**
9: return $\mathcal{C}$, S_γ, S_ρ.

Problem definition

We can now formulate the problem under investigation:

Problem 1 ($\mathcal{MCD}$) *Given a multidimensional network $\mathcal{G}$, find the complete set of multidimensional communities $\mathcal{C}$, and characterize each $c \in \mathcal{C}$ according to γ and ρ.*

As we have seen in the previous section, while finding communities in multidimensional networks is a problem already studied in the literature, to the best of our knowledge, we are the first to introduce formally the problem of finding and characterizing *multidimensional* communities in such networks.

An algorithm for $\mathcal{MCD}$

Given the problem definition above, a complete solution for it would require to design and develop an algorithm for extracting multidimensional communities, driven by the multidimensional density of the connections among nodes. However, according to our vision, it is difficult to define multidimensional density as universal, which is exactly what makes γ and ρ both meaningful. In addition, we believe that trivial design choices may lead to an algorithm producing communities with distributions of γ and ρ possibly unfairly unbalanced by the decisions taken. Moreover, we believe that the main contributions of this section are the problem definition and the characterization of the communities by the introduction of γ and ρ. For all these reasons, we leave for future research the design and implementation of a multidimensional community discoverer able to exploit the additional degree of freedom that multidimensionality provides, and here we propose a different solution based on existing, monodimensional, algorithms.

To apply existing solutions to multidimensional network, and to extract multidimensional communities, we have to introduce a mapping function ϕ able to transform a multidimensional network in a monodimensional one, trying to preserve as much information as possible, and a function ϕ' which recovers multidimensional information from monodimensional communities. Algorithm 1, which is a possible solution for $\mathcal{MCD}$, follows exactly this idea. Given a multidimensional network $\mathcal{G}$, a mapping function ϕ, and a monodimensional community discovery algorithm CD, *MCD_Solver* works as follows: in line 1 it applies ϕ in order to obtain a monodimensional view of $\mathcal{G}$; in line 2 the monodimensional community discovery is applied to G, and its resulting communities are stored in C; in lines 3-7, for each monodimensional community found, we restore its original multidimensional structure via ϕ', then we put the obtained multidimensional community in $\mathcal{C}$, and its characterizations obtained via ρ and γ in the sets S_ρ and S_γ , respectively; we then conclude returning the three sets $\mathcal{C}$, S_ρ, and S_γ.

We next give possible definitions of ϕ, we discuss which algorithm to use as CD, and we see how to implement ϕ'.

Three possible ϕ mappings

There can be several different definitions for ϕ, leading to different monodimensional networks built from $\mathcal{G}$. One possible class of them can be designed by simply *flattening* multidimensional edges to monodimensional ones, possibly weighting the monodimensional edges by some functions of the original multidimensional structure. An observation in support for this strategy is that many community discoverer use edge weights to reflect a more sophisticated definition of *dense* connections. In the following we assume to use a weight-based class of ϕ functions, and, in order to try to preserve as much multidimensional information as possible, we define three different weighting strategies, leading to three different ϕ.

The first weight we define is μ and requires to weight the (u, v) edge in G with 1 if there exists at least one dimension connecting u and v in $\mathcal{G}$, or, in formula:

$$\mu_{u,v} = \begin{cases} 1 & \text{if } \{\exists\ d : (u, v, d) \in E\} \\ 0 & \text{otherwise} \end{cases} \tag{8.10}$$

In the remainder of the paper, we refer to the ϕ designed with this weight as ϕ_μ. This flattening clearly loses most of the multidimensional information residing in $\mathcal{G}$, except the neighborhood: any two nodes connected in $\mathcal{G}$ are also connected in G.

Can we do better? Can we preserve more of the original information? One small improvement would be counting the number of dimensions connecting any two nodes u and v and using this as weight for the monodimensional edge added. We call this weight ν, which can be defined as:

$$\nu_{u,v} = |\{d : \exists u, v \in V : (u, v, d) \in E\}| \tag{8.11}$$

and we refer to the ϕ built upon ν as ϕ_ν.

We now consider a slight modification of ν that, instead of taking into account only the connection between u and v, also looks at their neighborhood, motivated by the intuition that common neighbors will likely be in the same community of u and v. We refer to this weight as η and define it as:

$$\eta_{u,v} = 1 + \sum_{d \in D} \frac{|N_{u,d} \cap N_{v,d}|}{|N_{u,d} \cup N_{v,d}| - 2} \tag{8.12}$$

where $N_{\cdot,d}$ is the set of neighbors in dimension d for a node. This is actually a multidimensional version of the edge clustering coefficient, and, according to the intuition behind it, should be able to better reflect the strength of the ties.

Note that there could be many other possible weighting strategies, as well as other different classes of ϕ relying on different principles. For example, one might consider to use the betweenness centrality instead of the clustering coefficient, or it is possible to consider also even more sophisticated measures. Note, however, that this could also mean additional computational complexity at the pre-processing stage. However, to keep complexity low, and for sake of simplicity, in this section we only examine the results obtained by using the three ϕ defined above. In the future we plan to introduce more sophisticated functions, and to give an extensive comparison of the benefits obtained by varying the definition of ϕ.

The choice for CD

In Algorithm 1, once a multidimensional network is mapped to a monodimensional one, the next step is to extract monodimensional communities. At this stage, any algorithm for community discovery can be used, with one *caveat*: we built a class of weight-based ϕ functions. This has to be taken into account by the algorithm, thus the only limitation we pose is to choose an algorithm able to handle edge weights. In our experiments, we present the results obtained by using an algorithm based on random walk [215], one based on label propagation [220] and one based of the fast greedy optimization of the modularity [75] as choices for possible monodimensional community discoverer. In our analysis we show how the choice among these three does not significantly affect the resulting distribution of γ and ρ.

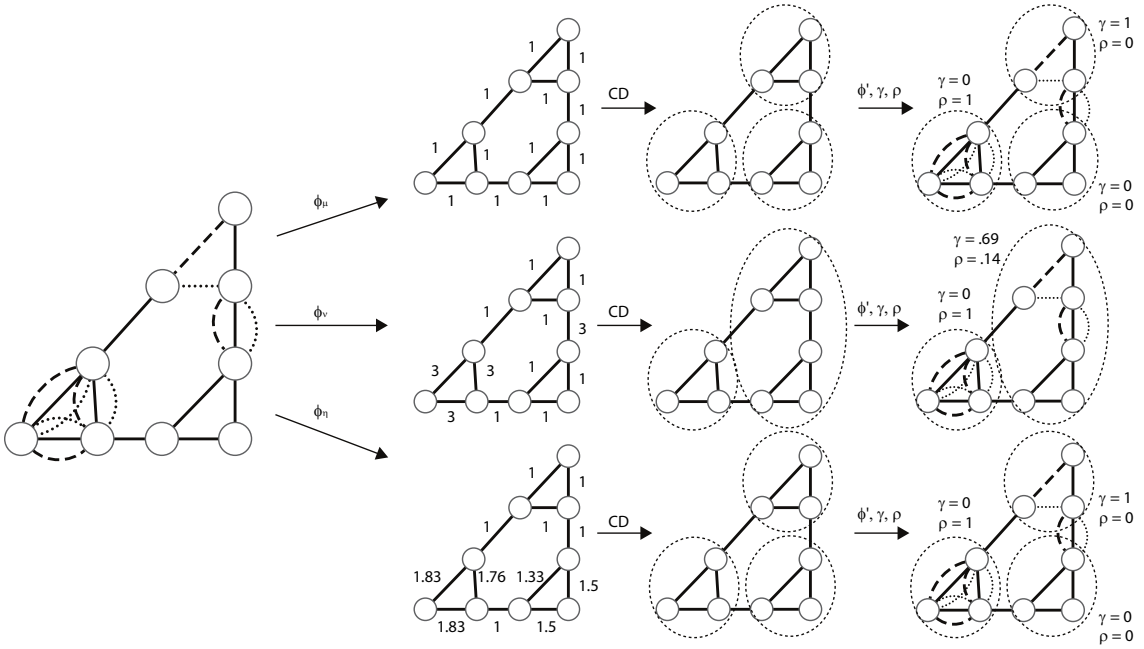


Figure 8.2: Run through example for three instances of MCD_Solver varying the ϕ parameter.

Returning multidimensional communities via ϕ'

We now address a last open point: given the set of monodimensional communities returned by the CD step, how to get back restoring the original multidimensional information. This step turns out to be trivial, as, for every community, we have the set of the IDs of the nodes involved: then, for each connected pair, we only have to restore its original multidimensional connectivity in $\mathcal{G}$.

Complexity

Algorithm 1 is the composition of three main steps: the computation of ϕ (hereafter, STEP1), the monodimensional community discovery (STEP2), and the computation of ϕ', γ and ρ (jointly, STEP3). As it is trivial to infer from their formulations, ϕ, γ and ρ might be implemented by scanning each edge in E only once, therefore their cost is O(E). The computation of ϕ' is slightly different since in the worst case we have to scan once each connected pair, thus requiring O(P), which, in turns, can be approximated by O(E), since generally $|N| >> |L|$. At this point, it is clear how the total complexity of Algorithm 1 might be dominated by the choice of CD. State of the art algorithms for CD, in fact, vary from a complexity of $O(|E|+|N|)$ (where E here is a set of monodimensional edges) for algorithms such as [220, 267], to a complexity of $O(3^{|N|/3})$ for [163]. Hence, it is clear how the choice for a proper CD should be driven by a good tradeoff between running time and quality of the results.

Running example

Figure 8.2 shows a running example of MCD_Solver. In Figure 8.2(a) we have our toy input network. We then run three different instances of our algorithm by varying the ϕ parameter. While we imagined to use three different ϕ in this example, we fixed the choice for CD, as a discussion on it is out of scope here. In Figure 8.2(b-d) we see the three instances (one per line) on the input network. In this simple example, we assumed to easily discover the communities highlighted in Figure 8.2(c-d). As one can see, the effect of the three different ϕ functions is to produce three different input networks for the monodimensional community discovery algorithm

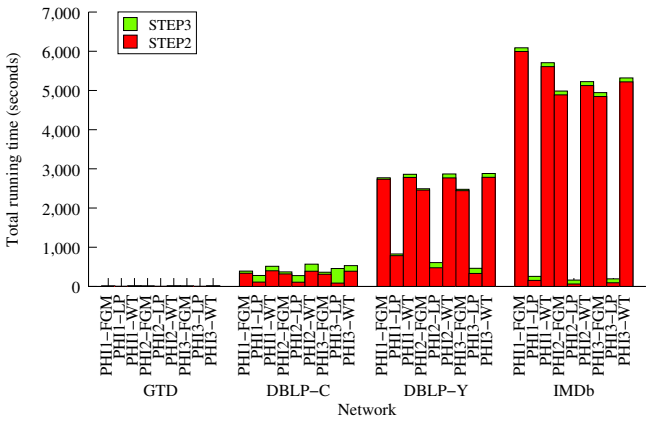


Figure 8.3: The running times of STEP2 and STEP3 on our networks (color image).

in the CD step, which will affect the resulting communities, hence the distributions of γ and ρ.

8.1.2 Experiments

We tested our framework on different real world networks, namely the GTD, DBLP-C, DBLP-Y and IMDb networks introduced in Chapter 5. We ran our experiments on a server with 2 Intel Xeon processors at 3.2GHz, equipped with 8GB of RAM, running GNU/Linux 2.6.27. MCD_Solver was implemented using the software for statistical analysis R, making use of the igraph[2] library.

For the CD step, as stated above, we chose three different algorithms: on based on random walk [215], one based on label propagation [220] and one based of the fast greedy optimization of the modularity [75]. In the rest of the section we refer to them as WT, LP and FGM. Note that, while LP and FGM are parameter-free, WT requires the length of the walk (that we set to 4 after empirical observations). Note also that WT and FGM returns the complete dendrograms of the communities, thus we had to choose a way to cut it. We then decided to take the cut maximizing the modularity as the best cut.

Figure 8.3 reports the running times for STEP2 and STEP3 of all the instances of MCD_Solver ran during our experiments. Since STEP1 may be performed once for all for each network, it is not reported here. As we see, in line with the theoretical complexity, the execution of the CD algorithms is the bottleneck for MCD_Solver.

Quantitative Evaluation

Purpose of this section is to give a quantitative analysis of the results obtained, under two different perspectives driven by the following questions:

Q1. Can we evaluate the performances of the different conjunctions of ϕ and CD, and compare them among the different networks?

Q2. How does the choice of a combination of ϕ and CD affect the distribution of γ and ρ over the communities?

Q3. What is the best choice of ϕ and CD parameters?

In order to answer Q1, we looked at the values of the modularity measure (as defined in [75]), computed on the resulting set of communities C. Note that we could have computed the modularity on $\mathcal{C}$ instead (the modularity allows to be computed also in multidimensional networks), but this would have been inconsistent with the use of ϕ, which would have been disregarded in that way. Instead, the modularity takes into account the weights defined in ϕ.

[2] http://igraph.sourceforge.net/

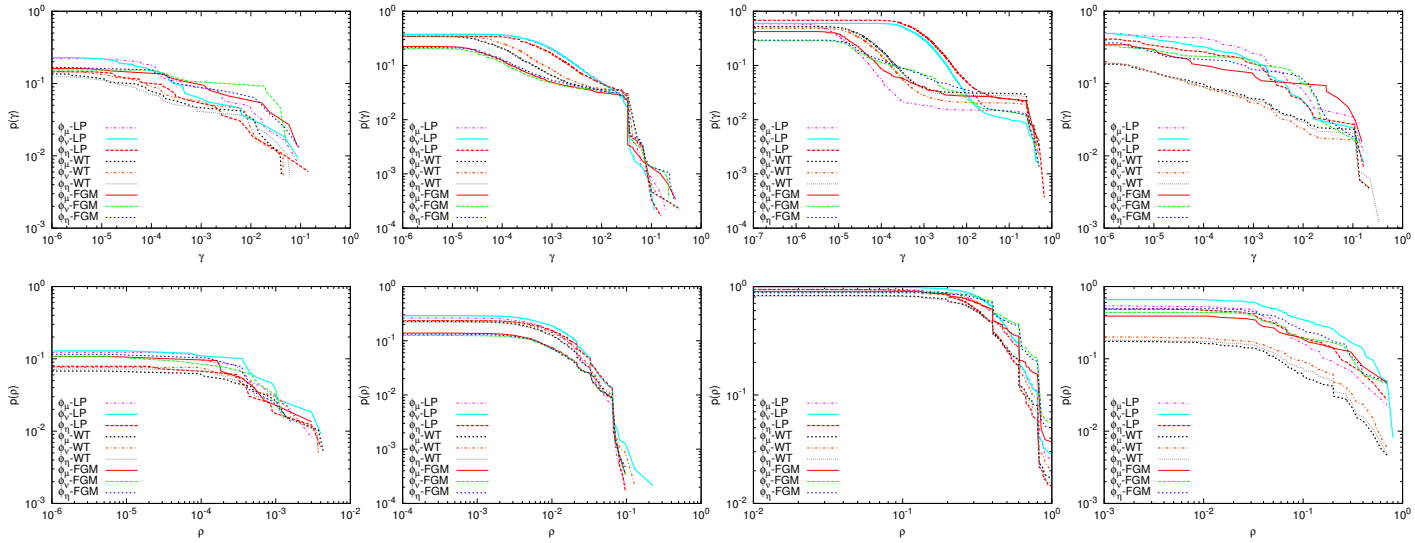


Figure 8.4: The cumulative distributions for γ and ρ in (from left to right): GTD, DBLP-C, DBLP-Y and IMDb datasets.

Network	ϕ	LP		WT		FGM	
		$\lvert\mathcal{C}\rvert$	Q	$\lvert\mathcal{C}\rvert$	Q	$\lvert\mathcal{C}\rvert$	Q
GTD	ϕ_μ	122	**0.622**	192	0.620	74	0.584
	ϕ_ν	109	0.547	197	0.603	65	0.611
	ϕ_η	165	0.500	194	**0.621**	78	**0.616**
DBLP-C	ϕ_μ	4625	**0.793**	4216	0.819	2931	0.860
	ϕ_ν	4685	0.791	4629	0.810	2820	0.881
	ϕ_η	5983	0.783	4345	**0.837**	2869	**0.903**
DBLP-Y	ϕ_μ	1632	0.190	5064	0.561	980	0.593
	ϕ_ν	8088	**0.591**	6397	0.638	754	**0.730**
	ϕ_η	8084	0.584	6131	**0.643**	722	0.723
IMDb	ϕ_μ	87	0.415	860	0.494	64	0.442
	ϕ_ν	124	**0.483**	847	**0.541**	66	**0.536**
	ϕ_η	148	0.460	823	0.507	63	0.530

Table 8.1: Number of communities found ($|\mathcal{C}|$) and modularity (Q) for each combination of network and parameters.

This measure gives a value between minus one and one, indicating how "good" nodes where partitioned into groups. The higher the value of modularity, the higher the partitioning reflects the division in the community of the graph that maximizes intra-community edges and minimizes inter-community edges. Many researchers use the modularity scores as evaluation, or as parameter to be optimized by the community discovery algorithm. However, this is only a partial evaluation of the results, since the well-known problems of modularity [93] (such as the resolution problem, witnessed also by our Table 8.1 in which one can see that modularity-based algorithm FGM retrieve always a smaller number of bigger communities).

Nevertheless, we computed the modularity for each combination of CD algorithm and ϕ preprocessor, for all the networks. In Table 8.1 we report the modularity values, highlighting in bold, for each algorithm, which ϕ produced the highest value. We are interested in seeing whether a specific combination of ϕ and CD tends to produce higher scores. Note that the values are not comparable between different networks since different network topologies may facilitate higher scores. As an example, we can see that in DBLP-C it looks to be easier to obtain higher scores, and this is due to the network statistics (see Chapter 5).

From Table 8.1, we may notice that in only 2 cases out of 12 network-algorithm combinations, ϕ_μ was the best among the three ϕ. This confirms that, in most cases, keeping more information about the dimensions of a network leads to higher modularity, i.e. to a better set of communities. Also, the definition of the dimensions heavily influence the scores of one particular choice of ϕ: both ϕ_η and ϕ_ν produce the highest scores in five cases, but ϕ_ν leads to these scores in DBLP-Y

and IMDb, both networks with a definition of dimensions based on time.

In order to answer Q2, we analyzed the distribution of γ and ρ for the output of each network-ϕ-algorithm combination. These distributions are depicted in Figure 8.4. We can see that in some cases there is a particular algorithm which outputs generally higher values: LP in DBLP-Y for γ, or FGM in GTD for γ. However, there is not a universally dominant combination. This suggests that if the analyst is particularly interested in higher values of γ or ρ, he/she can tune both ϕ and the community discovery algorithm to facilitate the discovery of those communities.

In addition, the information in Figure 8.4 may be used in conjunction with modularity in order to achieve richer knowledge about the results. Modularity, in fact, indicates how well the network is partitioned, and γ and/or ρ distributions characterize the multidimensional structure of the partitioning.

Finally, from these plots we can also see what is the highest value for both the measures, in each network, which provides a guide for the multidimensional analysis and interpretation of the communities achieved, that can be selected by using thresholds on their γ and ρ values.

Before answering Q3, one last consideration can be done looking at the values in Table 8.1: there is no strong prevalence of one choice of parameters over the others. The same happens also for the distributions of γ and ρ. This suggest that the best answer for Q3 really depends on the final analysis of the network: the application scenario, the semantic of the dimensions and the time budget for running the experiments should drive the analyst towards the proper choice of the two parameters for MCD_Solver. We leave for future research the definition of a parameter-free framework able to automatically tune the parameters driven by the optimization of user-defined objective functions.

Analysis of Interesting Communities

We extracted two examples of communities for each of the GTD, DBLP-C, DBLP-Y, and IMDb networks, for which we assume the reader to be the most familiar with. For each network, we extracted one community with a relatively high (among the top 10%) score of γ and one with a relatively high ρ. In each of these examples depicted in Figure 8.5, edges are represented by a different visual style according to the different dimensions they belong to. Figure 8.5(g) differs from the others, as described below.

For the GTD dataset, a community with relatively high complementarity is shown in Figure 8.5(a). In this case γ is able to unveil a very interesting structure for a possible counter-terrorism analysis. In fact we have identified two monodimensional communities joined together by a bridge node. In this case the "Tuareg" group is acting both in Niger and Nigeria, probably unifying two different communities. By intensifying action against particular bridges groups identified with the complementarity in the community discovery, one agency may be able to break alliances and coordination among a vast community of terrorist cells.

We depicted in Figure 8.5(b) a community with high redundancy score. While a complementarity in GTD spots bridge terrorist cells, as we have seen before, in this case the redundancy is able to identify those groups acting in a complex and multiple location scenario. The example depicted refers to the complex situation of the former Yugoslavia, but we have also example from another multi ethnic conflict: the Hutu vs Tutsi fight. Finding high ρ values in this network seems an effective strategy to spot these very complicated scenarios, where the social and political identity itself of different populations is degenerating in a violent conflict.

For the DBLP-C dataset, a community with high complementarity is shown in Figure 8.5(c). It is possible to see one of the main phenomena that we are interested in capturing with our definition of γ: this community presents a one-dimensional main body, to which other nodes connect via other dimensions. These nodes would not have been considered part of this community without the multidimensional approach. The peripheral researchers, such as Cao Dongwei, are included in the community because they collaborated with some members of the main body of it, but in different conferences.

For DBLP-C, Figure 8.5(d) reports an example of an interesting community according to the redundancy. In this case it is easy to highlight a very cohesive group. We are also able to infer

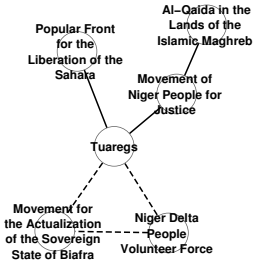
additional important knowledge: this quasi-clique represents a group of authors not only connected by many publications, but also by different, multidisciplinary, venues.

In Figure 8.5(e) we report an example of a community with high complementarity for DBLP-Y. Again, a crucial aspect of the γ measure is highlighted: we are dealing with a triangle composed by edges belonging to different dimensions (i.e. years). There is only one edge per dimension in the triangle, showing the true power of multidimensional community discovery.

A community with high ρ score for the DBLP-Y dataset is shown in Figure 8.5(d). As in Figure 8.5(f), again we can identify a very cohesive group of researchers who worked together. If in the DBLP-C community this type of structure was a sign of multidisciplinary of its members, in this case it is an example of temporal continuity: the authors in this community have published together in each of the years of the network.

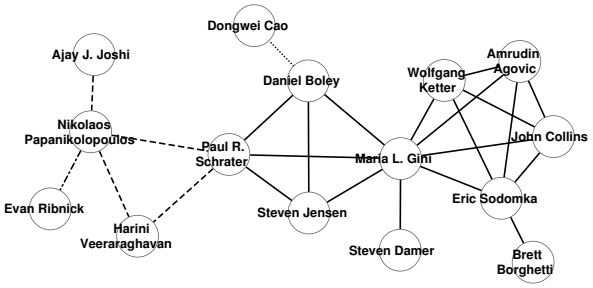
In the IMDb dataset we found one community with high complementarity, which was too large to be easily represented (more than 150 nodes). In Figure 8.5(g) we give a possible representation of it: each node (note the different node style) is, in turns, a subgroup of nodes highly connected within the community. Each group, with a backward analysis of the network meta-data, was found to represent a different documentary (titles provided as node labels). We there represented, then, a series of documentaries dedicated to a few important persons related to the cinema, producted in the last decade. The community has high complementarity because the personalities in a single documentary are connected only by the year of release of the documentary itself. Typically, this happens for directors, actors and other personalities no longer in activity since decades, which are then bound together not by their own works, but thanks only to these kind of documentaries. These personalities are persons such as Alfred Hitchcock, Jean-Luc Godard, François Truffaut, Satyajit Ray, Groucho Marx, Luis Buñuel, Salvador Dalí, Federico Garcia Lorca and more. The connections between the documentaries are due to some stars present in both films, like John Wayne linking "I'm King Kong" to "Go West, young boy!". Redundancy in IMDb (for which we have an example of a community with high ρ in Figure 8.5(h)) is able to identify large teams with continuous collaborations along many years. Generally, the very popular stars who work for more movies together are rather small groups (two or three persons, for instance the collaboration between Johnny Depp and Tim Burton). Bigger teams are usually groups of amateurs producing B-movies. The exception to this rule is the interesting case of some masters of the Iranian cinema, which we have represented in the above mentioned Figure 8.5(h). In this community some famous names pop out, like Mohsen Makhmalbaf and Jafar Panahi, authors of contemporary masterpieces such as "Kandahar" (original title "Safar e Ghandehar", nominated in 2001 for the *Palme d'Or* at the Cannes Film Festival) and "The Circle" (original title "Dayereh", Golden Lion winner at the Venice Festival in 2000).

We have addressed the problem of community discovery applied to the scenario of multidimensional networks. We have given a possible definition of multidimensional community as densely connected nodes in a multidimensional network. We have then provided two different measures aimed at quantify and disambiguate the *density* of the community in the multidimensional scenario. On this basis, we have devised a framework for finding and characterizing multidimensional communities, which is based on a mapping from multidimensional to monodimensional network, on the application of existing monodimensional community discovery algorithms to it, on the restoring of the originally residing multidimensional structure of the communities, and on the characterization of them via the γ and ρ measures. Our results obtained on real world networks are encouraging, and provided a basis for future research on this direction. In particular, we plan to investigate the following possibilities: the creation of a multidimensional community discovery algorithm driven by γ and ρ scores, possibly based on existing multidimensional methods such as the one in [192]; an extended quantitative evaluation of the results by means of additional measures to be used in conjunction with γ and ρ, such as the partition density [12] and/or conductance [176]; the definition of an ad-hoc multidimensional evaluation measure, which should be, according to our vision, independent from the network topology and statistics.

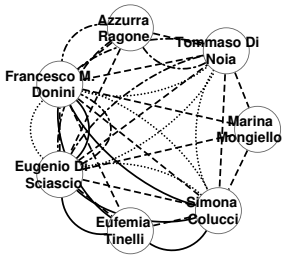


(a) high γ (0.0081) in GTD

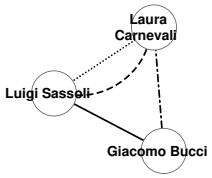
(b) high ρ (0.0035) in GTD



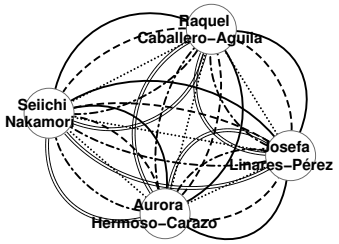
(c) high γ (0.00064) in DBLP-C



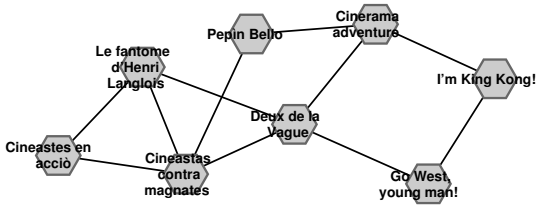
(d) high ρ (0.06048) in DBLP-C



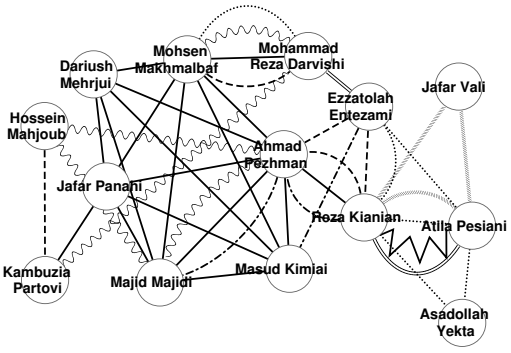
(e) high γ (0.5) in DBLP-Y



(f) high ρ (1) in DBLP-Y



(g) high γ (0.005) in IMDb



(h) high ρ (0.05) in IMDb

Figure 8.5: A few interesting communities found, with their γ or ρ.

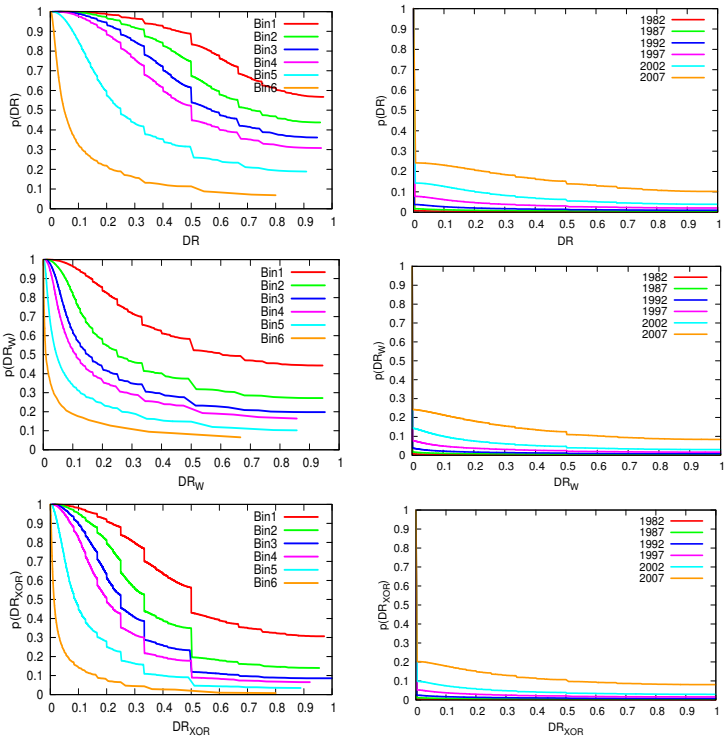


Figure 8.6: QueryLog (left column) and DBLP-Y (right) original Dimension Relevance distributions.

8.2 Multidimensional Network Models

As briefly introduced in Section 7.1, one very important aspect in complex network theory is represented by the study of network null models. To understand the dynamics that regulates the creation and the evolution of links in complex networks is a crucial aspect, because it gives us the power to better understand and predict future behavior. Many network models have been developed for classical monodimensional networks. They date back to the random graph [239, 87], until more recent approaches, as reviewed in Section 2.4. As we have pointed out in many sections of this thesis, even in this case there is little or no study about multidimensional network generators. The aim of this section is to provide some basic network generators and to verify if they are able to grasp some of the crucial characteristics of real world multidimensional networks.

We built four different multidimensional network generators, each with different characteristics, starting from a simple random generator, towards a generator that tries to preserve a global property of the original network that we might see as correlated with our measures, namely the Edge Dimension Correlation 7.2, being intuitively the most important measure to represent dimension interplay. For each model, we present its characteristics and the evaluation of the Dimension Relevances (Section 7.1) on the QueryLog and the DBLP-Y networks, because we choose to verify how much of the local dimension interplay, i.e. the distribution of the DRs, is preserved from the original network. We recall that the Dimension Relevance class of measures quantifies the importance of each single dimension in the economy of the connections of the network, using a node centric perspective. For each node, it quantifies how much each dimension is important in the economy of its connections. The distribution of the DRs gives a picture about how the importance of the dimensions is distributed on the network.

For clarity, we report the original Dimension Relevance distributions for Querylog and DBLP-Y networks in Figure 8.6, originally plotted in Figure 7.3(a-c) and 7.3(g-i) respectively, allowing an easier comparison.

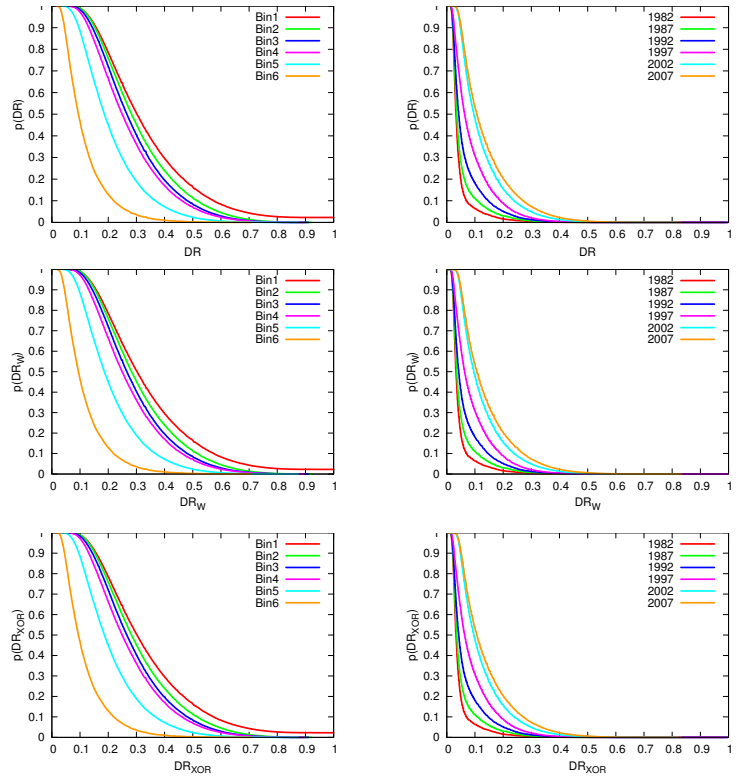


Figure 8.7: Random: QueryLog (left column) and DBLP-Y (right)

Random

We created a generator of random multidimensional networks, which takes in input the number of dimensions to generate, and the number of nodes and edges to put into each dimension. We fed the generator with these statistics computed on the real QueryLog and DBLP-Y networks.

Figure 8.7 shows the cumulative distribution of the DR (top row), DR_W (central row), and DR_{XOR} (bottom row), computed on the QueryLog-like (left column) and DBLP-Y-like (right column) networks. As we expected, the distributions of the DRs looks much different with respect to the original ones, and the relationships among the dimensions residing within the original networks look destroyed when compared to the original distributions (see Figure 7.3(a-c) for QueryLog and Figure 7.3(g-i) for DBLP-Y). Note that the distributions per dimension do not overlap, as we might expect for a random graph, given that we are preserving the number of nodes and edges per dimension, and this causes the DRs computed for each dimension to take different values.

The distributions provide evidences that the knowledge extracted by the DRs on random networks is much different with respect to the one deriving from real data, thus making the knowledge extractable with this analysis on real data non random, supporting then the meaningfulness of the measures. On the other hand, multidimensional random generator, as expected, is a failure in grasping real world network properties. Under this aspect, this is a perfect parallelism with the random network generator also in the monodimensional scenario. We then wanted to see in the next generators what we can add to the null model in order to make the DR distribution look closer.

Preferential Attachment

For the second generator, we took in input the same parameter as the previous one, but we built every dimension by evolving it following the preferential attachment model [30], i.e., after

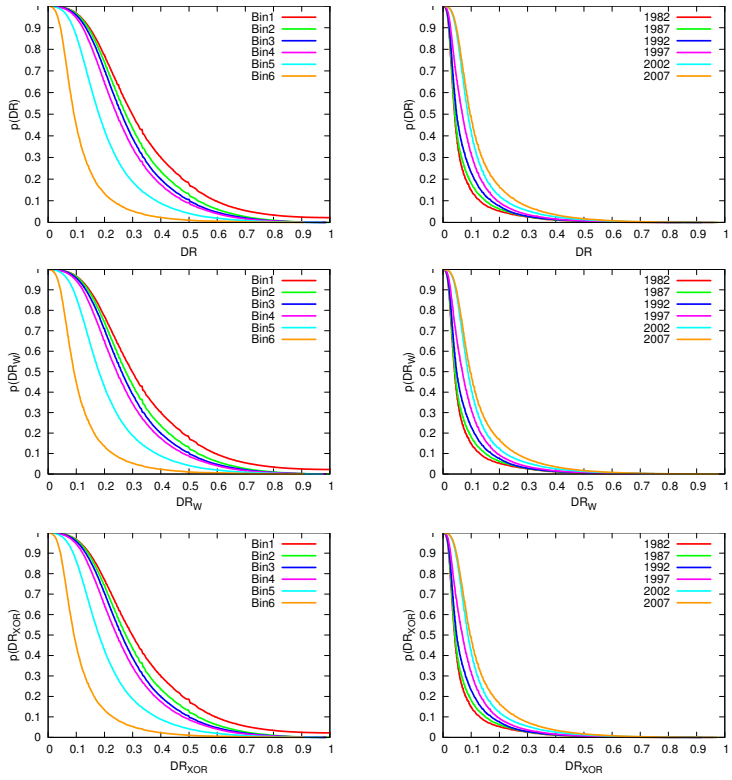


Figure 8.8: Preferential attachment: QueryLog (left column) and DBLP-Y (right column)

a bootstrap consisting of a clique of three nodes, we iteratively added a node attaching it to a random node with a probability directly proportional to its degree. Figure 8.8 reports the distributions of the DRs computed on the two networks. As we can see, we are not adding any significant information to the model compared to the random graph. This is very interesting, since PA-based models in the monodimensional scenario are indeed more useful than random graphs. Multidimensionality proves to be a completely different setting, in which the useful considerations about the degree distribution that unveil important knowledge in the monodimensional case are not enough.

Shuffle

The main explanation about the failure of both Random and PA models is linked to the fact that the two generators are producing random combinations of links, which is, obviously, destroying most of the original information. In the Shuffle generator, instead, we keep, dimension by dimension, all the characteristics of the original graph, except the relations among the dimensions. More clearly, we split the graph by dimensions, and we re-merge them in a random way, shuffling then all the node id correspondences among different dimensions. In this way, except destroying the interplay among dimensions, we are keeping most of the characteristics of the original networks.

As one can see in Figure 8.9 we obtain different results from Random and PA models, but still the DR distributions are very far from the original ones. At this point we might think that there is a strong relationship between the global correlation among dimensions, and the values of the DRs, that are, however, local measures.

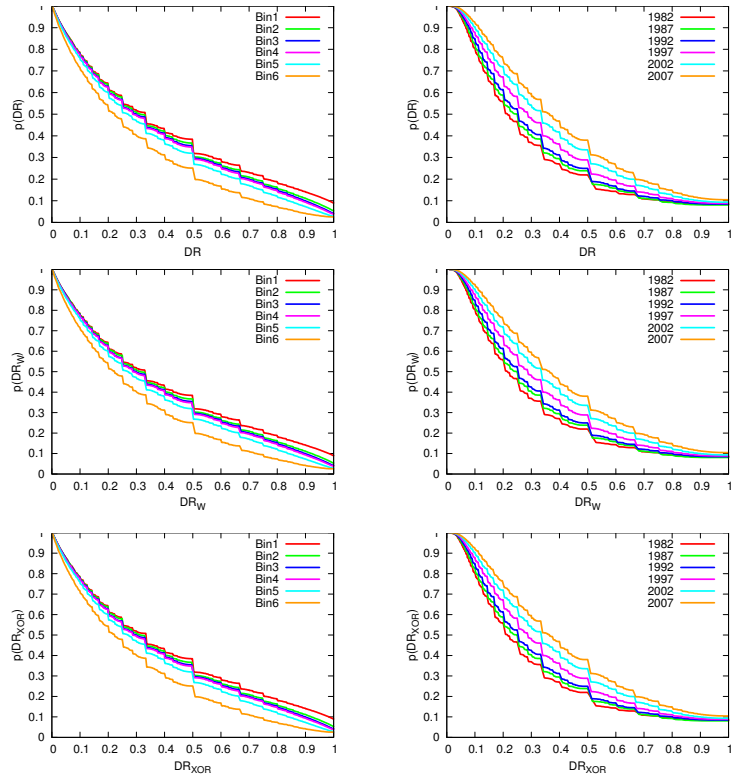


Figure 8.9: Shuffle: QueryLog (left column) and DBLP-Y (right column)

Edge Dimension Correlation

In order to validate the above hypothesis, we built a generator that preserves only the Edge Dimension Correlation. To be more clear, for each pair of dimensions x and y in the original network, we computed $\frac{|E_x \cap E_y|}{|E_x \cup E_y|}$, where E_x and E_y are the sets of edges belonging to dimensions x and y respectively, and generated a network preserving all these values. This was achieved by storing the set of multiedges connecting two nodes, and by using them to build the synthetic graph. The aim of this generator is to preserve the global interplay residing among the dimensions.

As Figure 8.10 shows, for QueryLog we are now a little closer to the original distribution of the pure DR, while this does not hold for the other two measures, nor for DBLP-Y. This is not surprising, as, by its definition, the capability of the pure DR to capture the interplay among the dimensions is weaker with respect to the other two. In particular, the exclusivity of the DR_{XOR} is a stronger concept, which is harder to preserve by this generator.

A different consideration must be done to explain the results in DBLP-Y. Looking at figures 7.3(g-i), we see how the distributions of the three measures are changed in these synthetic networks, in contrast to what happens to QueryLog. However, even though for sake of simplicity we plot only six of them, DBLP-Y has a total of 65 dimensions, thus making it more difficult to preserve the interplay among all of them, even with a little perturbation of the real network. This effect is weaker in QueryLog, that has a total of six dimensions.

As a last note, we must conclude this section with a final remark. What we learned from the null models are two things. Firstly, DR measures are capturing a local phenomenon that is not representable with global models. This consideration makes stronger the analytical need of the DR measures, since they allow analysis that are impossible with different techniques. Secondly, multidimensional network generators have to consider the interplay of dimensions to better represent real world networks. From Random to the Edge Dimension Correlation-based model, going through PA and Shuffle, we are constantly adding more and more information about this interplay

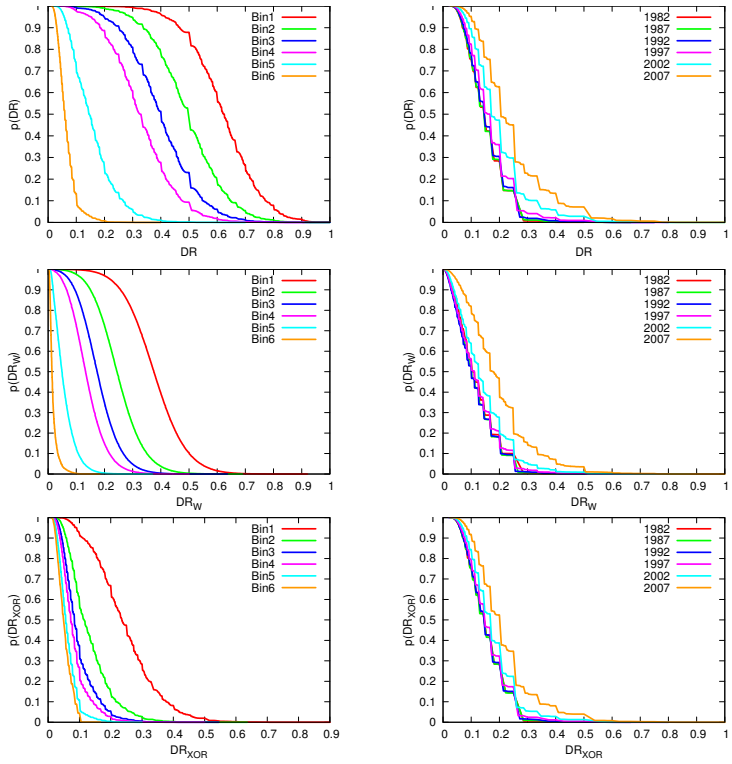


Figure 8.10: Jaccard: QueryLog (left column) and DBLP-Y (column)

and getting progressively better results. The theory about the non-random interactions between different dimensions gains strength from these evidences, and probably to obtain a reliable multidimensional network generator this path is worth exploring, maybe combining the global approach of Dimension Correlation with the local contribution of the Dimension Relevance for each single node (an approach with interesting results also in the monodimensional setting [172]).

8.3 Multidimensional Link Prediction

We already encountered multiple times in this thesis the link prediction problem. We recall that it basically constitutes in ranking unobserved edges to predict the probability they will appear in the network in the near future. We also already discussed about the additional degree of freedom represented by multidimensionality: it is no more only matter of finding the most probable couple of nodes that will be connected in the future, but also in which dimension, i.e. not only *who*, but also *how*.

As we presented, classical link prediction usually relies on simple connectivity measures such as the degree. Degree-based link predictors are the common neighbors, Adamic-Adar and preferential attachment models [207]. The intuition suggests us to use some of the presented multidimensional degree-based measures to extend known link predictors and make them able to "understand" multidimensionality. We are yet to define such multidimensional link predictor. However, in literature, this intuition has been explored in a preliminary work [226], that use the measures defined in this thesis for the link prediction task. Here we briefly describe the results.

The sole difference between our multidimensional model and the one used in [226] is the temporal information on the edges. In practice, each edge is represented by the quadruple (u, v, d, τ), where u and v are nodes, d is the dimension, as we know, and τ is the temporal snapshot in which the edge appears. Then, the authors consider as base predictor some straightforward extensions of the Adamic-Adar and Common Neighbors predictors:

$$MultidimensionalAdamicAdar(u, v, d) = \sum_{z \in N_{u,d} \cap N_{v,d}} \frac{1}{\log |N_{z,d}|}$$

$$MultidimensionalCommonNeighbors(u, v, d) = |N_{u,d} \cap N_{v,d}|$$

where $N_{u,d}$ is the number of neighbors of node u in dimension d. For brevity, the link predictor is called $M - AA$ and $M - CN$ for Adamic-Adar and Common Neighbors respectively. Each of the four variants is then combined with both multidimensional measures, temporal measures and a combination of both multidimensional and temporal measures.

The multidimensional measures considered are the Node Dimension Connectivity and the Edge Dimension Connectivity (NDC and EDC respectively), introduced in Section 7.3. The authors of the paper defines also two aggregations of the Node and Edge Jaccard measures introduced in Section 7.2, namely the average of the Node/Edge Jaccard over all dimensions (ANC and AEC respectively). Finally, they introduce a collection of temporal similarity measures: frequency, that simply counts the number of temporal snapshots in which an edge is present in a dimension (referred as Freq); over all frequency, a frequency aggregate by dimensions, counting the number of snapshots in which a pair of nodes is connected (referred as OAFreq); weighted presence, which gives more (or less) importance to more recent interactions (referred as WPres); and over all weighted presence, that is again the aggregation of the previous measure by dimensions (referred as OAWPres).

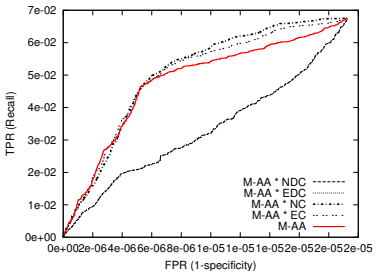
In Figure 8.11 the performances of all the combinations of link predictors are reported, using the ROC curves. The first two rows report the performances on DBLP-Y network for the Adamic-Adar and Common Neighbors respectively, the latter two rows the same performances on the IMDb network. In the first column only the multidimensional measures have been added to the chosen link predictor, in the second column authors added only the temporal measures, in the third column we have a combination of multidimensional and temporal measures.

As we can see, and as discussed in [226], pure multidimensional measures are able to provide only a small improvement over Adamic-Adar predictor, while Common Neighbors is almost not affected a lot. However, the combination of multidimensional and temporal measures provides some interesting insights. The conclusion is that the future research should be driven towards pure multidimensional predictors, as simply extending the classical monodimensional techniques cannot provide significant improvements.

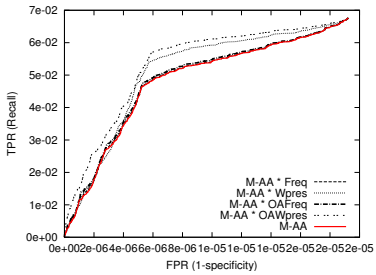
8.4 Multidimensional Shortest Path

In Section 6.2 we introduced a collection of measures for dealing with the shortest path problem in multidimensional network. This collection, however, is flat: it just describes how to extend the problem by filtering some dimensions or counting the number of times there is a dimension change. This is not sufficient for a practical use in real world problems.

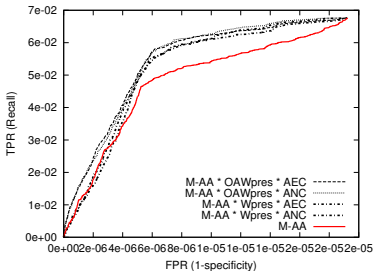
We choose as our main example the classical problem of a transportation network. In this network, nodes are the places we want to reach and/or we need to cross, the edges represent a connection between two places and the dimensions are the different modes of transportation. A dimension may be a car, a bus, tram or metro, bicycle, train or even plane, depending on the granularity of the data. It is clear that if we want to go from node A to node B it may be impossible, or inefficient, to use exclusively one dimension. To go from Rome to New York a plane (or a ship) is needed, but to reach the airport (or the seaport) different modes of transportation can be chosen. Moreover, to switch from one mode to another is not costless, as any commuter waiting for his/her train at the station can confirm. Finally, it is surely possible to change mode of transportation several times to increase the theoretical efficiency, but it may not be practically feasible for many reasons. For example, we may want to jump off the train at a station and take a car, but there is no car available waiting for us; or jumping on and off a train to do only the efficient railways sections on the train and then commute from one train station to the other with



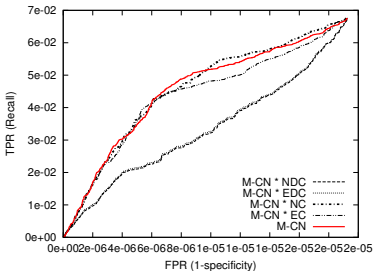
(a) DBLP M-AA Multi



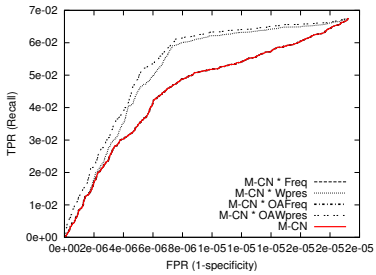
(b) DBLP M-AA Temp



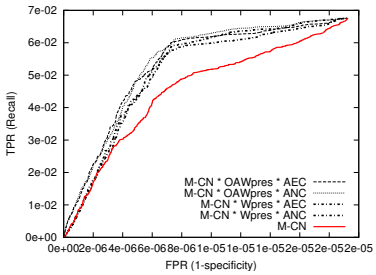
(c) DBLP M-AA Multi + Temp



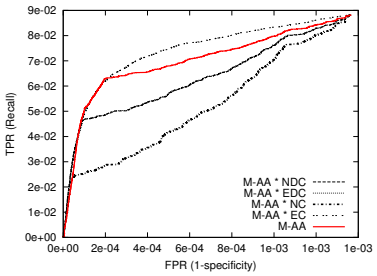
(a) DBLP M-CN Multi



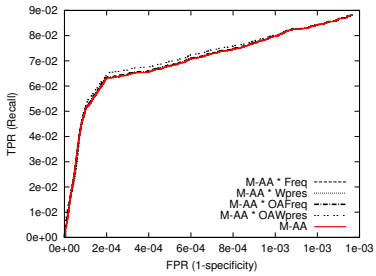
(b) DBLP M-CN Temp



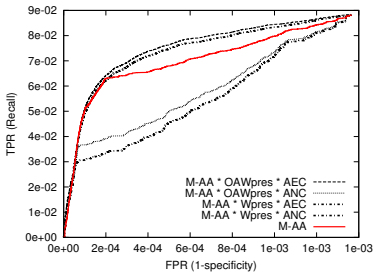
(c) DBLP M-CN Multi + Temp



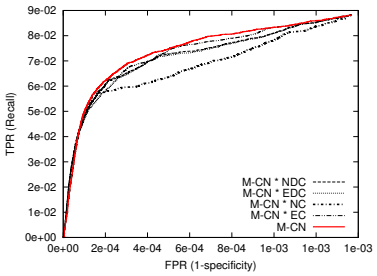
(a) IMDb M-AA Multi



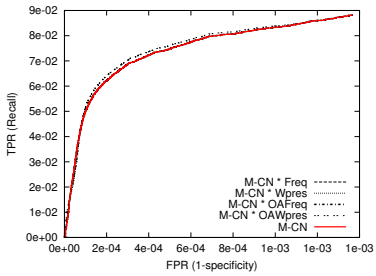
(b) IMDb M-AA Temp



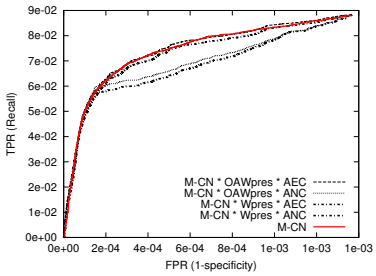
(c) IMDb M-AA Multi + Temp



(a) IMDb M-CN Multi



(b) IMDb M-CN Temp



(c) IMDb M-CN Multi + Temp

Figure 8.11: Performances of multidimensional link predictors.

more efficient modes of transportation is probably not efficient, as staying quiet in the same train for the entire trip is less costly in practice.

All these scenarios lead to more complex tools for a practical computation of shortest paths in multidimensional network. More precisely, here we define the problem of *finding the shortest path in a multidimensional network with cost modifiers*. To tackle this problem we need to introduce some modifications to our model. Of course, our multidimensional network has to be weighted, because otherwise the model loses importance for real world scenarios. Then we will use the classical definition of a simple path in a network, i.e. a sequence $P = (e_0, e_1, ..., e_{n-1}) = ((v_0, v_1, d_0, w_0), (v_1, v_2, d_1, w_1), ..., (v_{n-1}, v_n, d_{n-1}, w_{n-1}))$; where $v_0 \in V$ is the source node, $v_n \in V$ is the destination node, $e_k \in E$ and n is the path length. We recall that in a simple path there is no vertex v_i and no edge e_k crossed more than once.

Usually, the cost of a path is given by the sum of all the weights of the edges crossed, or $C(P) = \sum_{i=0}^{n-1} w_i$. We introduce the concept of a *path constraint*: a boolean check that, given a path, can return true or false, i.e. $A(P) \rightarrow \{T|F\}$. Finally, a cost modifier m is a couple (A, ϕ), where A is a path constraint and $\phi : \mathbb{R} \rightarrow \mathbb{R}$ is a function such that if $A(P) = T$, then $C(P) = \phi(C(P))$. M is defined as a collection of m. It is important to note that M has an internal order, that can be freely defined. Without an order, it is impossible to calculate the final cost of a path P. Let us assume we have two cost modifiers m' and m'' and path P satisfies both A' and A''. Without an order we cannot say if the final cost is $\phi'(\phi''(C(P)))$ or $\phi''(\phi'(C(P)))$, and given that ϕ' may be $+$ and ϕ'' may be $\times$, this will lead to two different results.

We are now able to define the shortest path in a multidimensional network with cost modifiers problem as follows:

Definition 34 (Multidimensional Shortest Path with Cost Modifiers) *Given two nodes v_1 and v_n in a multidimensional network and a set M of cost modifiers, find the path $P : v_1 \rightarrow v_n$ from the source node v_1 and the destination node v_n such that $\Phi(C(P))$ is minimal, where Φ is the ordered application of all $\phi_i \in M$ for which $A_i(P) = T$.*

We do not have a complete solution for this problem, as the research in this direction is in its early stages. We do have, however, a preliminary framework and a proposed implemented algorithm, MSPCM (Multidimensional Shortest Path with Cost Modifiers), that still needs a more principled formulation and rigorous experimental test. We record here what are the foundations and the intuitions constituting the basis of MSPCM.

There is a collection of shortest path algorithms in literature. The most popular in literature are Dijkstra, Bellman-Ford and Floyd-Warshall algorithms [13]. However, these algorithms are not suitable for our scenario, since they cannot handle cost modifiers. A naive solution would be to use a particular algorithm, for example Dijkstra algorithm and for each new edge to check $\Phi(C(P))$. This is not possible, since shortest path algorithms need to satisfy the Bellman equation. In terms of our problem, the Bellman equation is reduced to the following lemma: if P_{st} is a shortest path from s to t and it contains node j, then the sub-path from s to j is the shortest path between s and j. But an hypothetical cost modifier involving an additional node j' may be less costly for going from s to t, even if j' is not required to go from s to j because that particular cost modifier is not triggered.

Consider the graph in Figure 8.12. Obviously, without cost modifiers, the shortest path from node 0 to node 2 passes through node 3. But suppose that our set M contains two cost modifiers. The first modifier subtracts 3 from $C(P)$ if we cross two times in a row an edge in the red dimension. The second modifier adds 4 to $C(P)$ if we cross two times in a row an edge in the gray dimension. What happens now is that the shortest path from node 0 to node 2 crosses node 1 before node 3. But node 1 is not necessary for the shortest path from node 0 to node 3. Therefore the fundamental lemma presented before is not satisfied and no classical algorithm can be used to obtain an exact solution to Problem 34.

Currently, the only algorithm able to solve our problem with an optimal solution is the brute force algorithm, i.e. to calculate all shortest paths and then choose the path with the lowest cost after applying all cost modifiers. Of course, this is not what we are interested in, since the brute

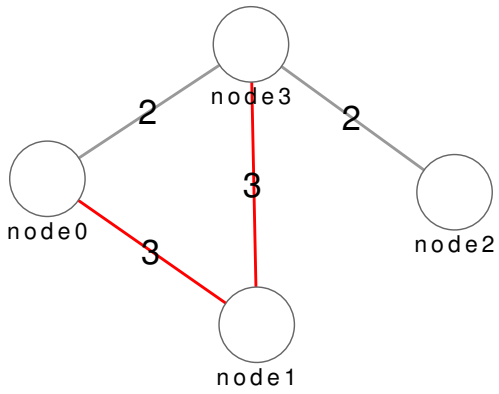


Figure 8.12: A toy example for the multidimensional shortest path with cost modifiers problem.

Algorithm 2 *Flattening*(G, M)

Require: multidimensional network $G = (V, E, L)$, list of cost modifiers M
Ensure: simple weighted graph G'

1: $E' \leftarrow \emptyset$
2: $G' \leftarrow (V, E')$
3: **for all** $\{(u, v) \mid \exists (u, v, d) \in E \wedge d \in D\}$. **do**
4: $\quad c \leftarrow H(F_{uv}, M)$
5: $\quad e' \leftarrow (u, v, c)$
6: $\quad E' \leftarrow E' \cup e'$
7: **end for**
8: return G'

force algorithm will scale as $\mathcal{O}(2^n)$, where n is the number of nodes of the network. The problem is reduced to find an efficient way to find a reasonably short path in a reasonable computing time.

The solution here proposed is a multi-step process. In practice, our algorithm will be composed by three phases:

1. Flattening preprocessing, in which we reduce the multidimensional network and the cost modifier set M into a simple flat structure G';
2. Shortest path detection, that uses a classical algorithm and G' as input to get an intermediate result P'_{st} for each source s and destination d;
3. Path reconstruction, in which we transform each optimal path P'_{st} for G' into a greedy shortest path P_{st} for the original network.

Flattening preprocessing. This is the most important step of our algorithm. The general idea is to create some function $F(G, M) = G'$ able to reduce the original problem into a simplified version that can be handled by the current state-of-the-art algorithms. Of course, the most accurate is G' in describing (G, M), the closest to optimal the results will be. The algorithm we chose is formalized in Algorithm 2. Some additional notation is needed.

The key concept here is the $H(F_{uv}, M)$ function. This functions takes as input a set of nodes and edges F_{uv}, a set of cost modifies M and returns a single edge connecting u and v with an estimation of its average cost. F_{uv} may be simply an arbitrary set of nodes and edges containing nodes u and v, however it is useful to define it as the neighborhood of u and v. The idea is to define a range and then collect into F_{uv} all nodes and edges distant from u or v at most the size of the range. The intuition of breaking down the complex shortest path problem into smaller sections of the network is also at the basis of the fastest implementation of the Newman-Girvan community discovery algorithm based on the concept of "split betweenness" [106] (see also Section

2.3). Into this small range, a smaller portion of the network is analyzed. The lowest cost among all the shortest paths crossing entirely this network portion and containing the edge (u, v) is then returned by the H function.

In practice, Algorithm 2 cycles over all the connected couples of G, regardless the dimensions involved, estimates the cost for passing through this connection applying $H(F_{uv}, M)$ on a given set F_{uv} and adds to a result graph G' an edge e' weighted accordingly to the estimate.

Shortest path detection. This step is trivial. Given a simple weighted graph G' we simply apply a chosen shortest path algorithm and we get P'_{st} for each source s and target t we are interested in. Any shortest path algorithm present in literature can be applied, since in G' the lemma derived from the Bellman equation holds.

Path reconstruction. The last step in conceptually simple. Given a path P'_{st}, optimal in G', we want to reconstruct the actual path P_{st} in our original multidimensional network. Technically, also this step may pose several algorithmic challenges. However, we make use of a classical empirical result of complex network analysis that holds generally for a fairly big amount of different complex networks. This result states that usually the shortest path between any two vertices in a complex network does not include an high number of edges. To be precise, this is the small-world property, derived from an intuition of Stanley Milgram [186] later quantified: the average length in terms of edges of the shortest path in a complex network is proportional to, and usually lower than, the logarithm of the number of nodes of the network itself [195]. Since the number of edges to be considered is small, at this step we apply a brute force solution for the path P'_{st} by checking all the possible combinations of edges in all possible dimensions from s to t and then we return the one that minimizes $\Phi(C(P_{st}))$.

We do not provide an extensive complexity evaluation, but we find that, using Floyd-Washall algorithm for the second step of our algorithm, the time complexity can be approximated to $\mathcal{O}(mn|M|\log n)$ if we want to find the entire collection of shortest path in the network. The Floyd-Washall algorithm forces us to compute all shortest paths instead only one separately, but for this property it needs to run the preprocessing stage only once. Dijkstra algorithm is more efficient for computing all shortest paths from a single root node, but for each root node the preprocessing stage needs to be run again. Also, the range for the approximation should be taken into account in the complexity. However, since for heuristic reasons we rarely will use a range higher than 6, its weight in the complexity is negligible.

We now want to test how efficient our heuristic is on real world data. We calculated all the shortest paths on a multidimensional network using our algorithm MSPCM and the brute force algorithm, that always retrieve the optimal solution. Both algorithms are implemented in Java. We will not use the networks presented in Chapter 5 since they are too big for the brute force algorithm: even the toy network from Facebook with less than 300 nodes takes too long. We instead used a simplified version of the MBTA transportation in Boston, using only the dimension of the subway and reducing the nodes only to the stations where it is possible to change the line (ending up with 27 nodes). On the other hand, when tested on DBLP-C and DBLP-Y networks, MSPCM was able to end its computation in orders of minutes.

One last caveat is needed. It has been observed that the best results in general are obtained with a range value between 4 and 6. However, in these cases the path shorter than the range values were far from optimal. This is a natural consequence of the example depicted in Figure 8.12 and described in the text: using a range of 4 the gain from the cost modifiers related to the red dimension is triggered, making those two edges very convenient in G', but then when moved to G the paths are not optimal. We found convenient, then, to make two consecutive runs of MSPCM, the first one with the optimal range and the second one with range equal to 1. For each path we then choose the solution with the lower cost. With this strategy, we firstly identify long paths and then we obtain the short paths with the second run.

The brute force algorithm found the average cost of all the paths in our MBTA representation to be equal to 17.73. MSPCM was able to find an average value of 19.57. Among the paths returned by MSPCM, 65.16% were also present in the brute force solution, i.e. almost two paths out of three were actually the optimal path. The biggest gain comes from confronting the running times. The brute force algorithm took 155 seconds to calculate all the paths on the 27 nodes

network, while MSPCM took 7.24 seconds. This means that with MSPCM we can expect to find 65% optimal paths with running time 20 times lower, and this gain grows for bigger networks, given the exponential nature of the brute force solution.

Part III

Novel Insights for Network Analysis

Chapter 9

The Product Space

In this chapter we present the creation and the analysis of the Product Space. The Product Space is a tool created with complex network analysis techniques that has been used to obtain an improved knowledge about the mechanics of international trade. We see how this is achieved using monodimensional network analysis, being only one kind of interaction included in the main structure of the Product Space. However, from the creation of the dataset itself to each analytical step, we are able to detect where multidimensionality could be useful to better grasp the real world complexity of international trade, leading to novel results.

We firstly introduce the concept of Economic Complexity, that is the fundamental building block of the Product Space. Then, we show why and how Economic Complexity is a very useful tool able to better explain the country-level trade dynamics of the real world. We then explain how to use Economic Complexity indicators to create the Product Space. Finally, we present one possible analysis done with the Product Space, namely a new product classification based not on a top-down interpretation of what the world should look like according to an analyst or a particular philosophy, but based on how product arrange themselves according to the capabilities needed to produce them. For each of these steps, our aim is to make evident where multidimensional network analysis can be applied.

The work of this chapter is mainly based on [128]. The original work introducing the Product Space concept is [131].

9.1 Economic Complexity

One way of describing the economic world is to say that products are made with machines, raw materials and labor. Another perspective is that products are made with knowledge. The true value of a tube of toothpaste is that it manifests knowledge about the chemicals that facilitate brushing, and that kill the germs that cause bad breath, cavities and gum disease.

Markets allow us to access the vast amounts of knowledge that are scattered among the people of the world, that is concentrated in all the different products. Products are vehicles for knowledge, but embedding knowledge in products requires people who possess a working understanding of that knowledge. Most of us can be ignorant about how to synthesize sodium fluoride because we can rely on the few people who know how to create this atomic cocktail, and who together with their colleagues at the toothpaste factory, can deposit it into a product that we can use. The division of labor is what allows us to access a quantity of knowledge that none of us would be able to hold individually.

We can distinguish between two kinds of knowledge: explicit and tacit. Explicit knowledge can be transferred easily by reading a text or listening to a conversation. If all knowledge had this characteristic, the world would be very different. Countries would catch up very quickly to frontier technologies, and the income differences across the world would be much smaller than what we see today. The problem is that crucial parts of knowledge are tacit and therefore hard to

embed in people: they require a costly and time-consuming effort. Because it is hard to transfer, tacit knowledge is what constrains the process of growth and development. Ultimately, differences in prosperity are related to the amount of tacit knowledge that societies hold. In allocating productive knowledge to individuals, it is important that the chunks each person gets be internally coherent so that he or she can perform a certain function. We refer to these modularized chunks of embedded knowledge as capabilities. Some of these capabilities have been modularized at the level of individuals, while others have been grouped into organizations and even into networks of organizations. Most products require more knowledge than can be mastered by any individual. Hence, those products require that individuals with different capabilities interact. Larger amounts of knowhow are modularized in organizations, and networks of organizations, as organizational or collective capabilities.

Ultimately, the complexity of an economy is related to the multiplicity of useful knowledge embedded in it. Economic complexity, therefore, is expressed in the composition of a country's productive output and reflects the structures that emerge to hold and combine knowledge. Increased economic complexity is necessary for a society to be able to hold and use a larger amount of productive knowledge, and we can measure it from the mix of products that countries are able to make.

9.2 How and Why Economic Complexity?

Now that we have the intuition and informal description of what Economic Complexity is, we define it formally. The primary input of the Product Space is represented by the international world trade. It can be formalized with a four dimensional tensor M_{ccpy}: for each country c_1 we have the exported quantity of each product p to the country c_2 in year y. To obtain the formal definition of the Economic Complexity Index, we reduce the number of dimension by projecting over destination countries and years. What we are interested in is the relations between countries and the products they export.

When associating countries to products it is important to take into account the size of the export volume of countries and that of the world trade of products. This is because, even for the same product, we expect the volume of exports of a large country like China, to be larger than the volume of exports of a small country like Uruguay. By the same token, we expect the export volume of products that represent a large fraction of world trade, such as cars or footwear, to represent a larger share of a country's exports than products that account for a small fraction of world trade, like cotton seed oil or potato flour. To make countries and products comparable we use Balassa's definition of Revealed Comparative Advantage or RCA. Balassa's definition says that a country has Revealed Comparative Advantage in a product if it exports more than its "fair" share, that is, a share that is equal to the share of total world trade that the product represents. For example, in 2008, with exports of $42 billion, soybeans represented 0.35% of world trade. Of this total, Brazil exported nearly $11 billion, and since Brazil's total exports for that year were $140 billion, soybeans accounted for 7.8% of Brazil's exports. This represents around 21 times Brazil's "fair share" of soybean exports (7.8% divided by 0.35%), so we can say that Brazil has revealed comparative advantage in soybeans.

Formally, if X_{cp} represents the exports of country c in product p, we can express the Revealed Comparative Advantage that country c has in product p as:

$$RCA_{cp} = \frac{\frac{X_{cp}}{\sum_c X_{cp}}}{\frac{\sum_p X_{cp}}{\sum_{c,p} X_{cp}}}.$$

We use this measure to construct a matrix that connects each country to the products that it makes. The entries in the matrix are 1 if country exports product with Revealed Comparative Advantage larger than 1, and 0 otherwise. Formally we define this as the matrix M_{cp}, where:

$$M_{cp} = \begin{cases} 1 & \text{if } RCA_{cp} \geq 1; \\ 0 & \text{otherwise} \end{cases}$$

We also smooth changes in export volumes induced by the price fluctuation of commodities by using a modified definition of RCA in which the denominator is averaged over the previous three years.

M_{cp} is then a matrix that is 1 if country c produces with profit product p, and 0 otherwise. We can measure diversity and ubiquity simply by summing over the rows or columns of that matrix. The Diversity $k_{c,0}$ of a country c is related to the number of products that a country is connected to. This is equal to the number of links that this country has in the country-product bipartite network represented by the matrix M_{cp}. The Ubiquity $k_{p,0}$ of a product p is related to the number of countries that a product is connected to. This is equal to the number of links that this product has in the aforementioned bipartite network.

Formally, we define $k_{c,0} = \sum_p M_{cp}$ and $k_{p,0} = \sum_c M_{cp}$. To generate a more accurate measure of the number of capabilities available in a country, or required by a product, we need to correct the information that diversity and ubiquity carry by using each one to correct the other. For countries, this requires us to calculate the average ubiquity of the products that it exports, the average diversity of the countries that make those products and so forth. For products, this requires us to calculate the average diversity of the countries that make them and the average ubiquity of the other products that these countries make. This can be expressed by the recursion:

$$k_{c,N} = \frac{1}{k_{c,0}} \sum_p M_{cp} k_{p,N-1}$$

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

We the insert $k_{p,N-1}$ into $k_{c,N}$ obtaining:

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

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

and rewrite this as:

$$k_{c,N} = \sum_{c'} \widetilde{M}_{cc'} k_{c',N-2},$$

where:

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

We note in the last formulation $k_{c,N}$ is satisfied when $k_{c,N} = k_{c,N-2} = 1$. This is the eigenvector of which is associated with the largest eigenvalue. Since this eigenvector is a vector of ones, it is not informative. We look, instead, for the eigenvector associated with the second largest eigenvalue. This is the eigenvector that captures the largest amount of variance in the system and is our measure of economic complexity. Hence, we define the Economic Complexity Index (ECI) as:

$$ECI = \frac{\vec{K} - < \vec{K} >}{\sigma(\vec{K})},$$

where $\vec{K}$ is the eigenvector of $\widetilde{M}_{cc'}$ associated to the second largest eigenvalue, $< \vec{K} >$ is its average and $\sigma(\vec{K})$ its standard deviation. Analogously, we define a Product Complexity Index

SITC4 Code	Product Name	PCI
7284	Machines & appliances for specialized particular industries	2.27
8744	Instrument & appliances for physical or chemical analysis	2.21
7742	Appliances based on the use of X-rays or radiation	2.16
3345	Lubricating petrol oils & other heavy petrol oils	2.10
7367	Other machine tools for working metal or metal carbide	2.05
3330	Crude oil	-3.00
2876	Tin ores & concentrates	-2.63
2631	Cotton, not carded or combed	-2.63
3345	Cocoa beans Tropical	-2.61
7367	Sesame seeds	-2.58

Table 9.1: The five most and least complex products according to PCI.

(PCI). Because of the symmetry of the problem, this can be done simply by exchanging the index of countries (c) with that for products (p) in the definitions above. Hence, we define PCI as:

$$PCI = \frac{\vec{Q} - < \vec{Q} >}{\sigma(\vec{Q})},$$

where $\vec{Q}$ is the eigenvector of $\widetilde{M}_{pp'}$ associated to the second largest eigenvalue, $< \vec{Q} >$ is its average and $\sigma(\vec{Q})$ its standard deviation.

The difference between the world's most and less complex products is stark (see Table 9.1). The most complex products are sophisticated chemicals and machinery that tend to emerge from organizations where a large number of high skilled individuals participate. The worlds least complex products, on the other hand, are raw minerals or simple agricultural products.

The economic complexity of a country is connected intimately to the complexity of the products that it exports. Ultimately, countries can only increase their score in the Economic Complexity Index by becoming competitive in an increasing number of complex industries.

In [128, 131], authors proved that ECI is related to a country's level of prosperity (except the cases in which a country can be relatively rich by only using natural resources, therefore being not complex). But this is not the end of the story. Countries whose a economic complexity is greater than what we would expect, given their level of income, tend to grow faster than those that are "too rich" for their current level of economic complexity. In this sense, economic complexity is not just a symptom or an expression of prosperity: it is a driver. In [128] (Section 4) it has been shown that ECI is able to describe and predict future growth better than indicators used in the state-of-the-art economics research. We provide here one example. An increase of one standard deviation in ECI, which is something that Thailand achieved between 1970 and 1985, is associated with a subsequent acceleration of a country's long-term growth rate of 1.6 percent per year. This is over and above the growth that would have been expected from mineral wealth and global trends.

Up until now, we defined the traditional monodimensional version of ECI and we saw that it makes sense and it is a valuable tool to predict future growth. Now we are interested in answering the question: is multidimensional network analysis useful to improve ECI? We can provide an intuition of this. The usefulness of multidimensional network analysis lies in the early stages of the creation of the derived data structure M_{cp}. When we calculate RCA_{cp}, we are projecting over importing countries and years. These are two dimensions that cannot be handled with a monodimensional analysis, being the final aim to handle a monodimensional bipartite network connecting countries to products. With multidimensional network analysis, this bipartite network can be multidimensional: we can take into account the countries where the products are going, or to have a temporal analysis by including also the years as dimensions (as seen in Section 7.2).

9.3 Product Space Creation

In the previous section we saw that ECI is a good describing tool for the world trade economic dynamics. What is important is if we are able also to understand how ECI evolves, that will enable us to have a deeper knowledge about these dynamics. Specifically, questions like the followings can be answered: how do societies increase the amount of productive knowledge embedded in them? What limits the speed of this process? And why does it happen in some places but not in others?

The complexity of a country's economy reflects the amount of productive knowledge it contains. This knowledge is costly to acquire and transfer, and is modularized into chunks we call capabilities. Capabilities are difficult to accumulate because doing so creates a complicated chicken and egg problem. On the one hand, countries cannot create products that require capabilities they do not have. On the other hand, there are scant incentives to accumulate capabilities in places where the industries that demand them do not exist. New capabilities will be more easily accumulated if they can be combined with others that already exist. Countries are more likely to move into products that make use of the capabilities that are already available. We are then interested in measuring the similarity in the capability requirements of different products.

Our basic assumption is that the probability that a pair of products is co-exported carries information about how similar these products are. Our measure is based on the conditional probability that a country that exports product p will also export product p'. Since conditional probabilities are not symmetric we take the minimum of the probability of exporting product p, given p' and the reverse, to make the measure symmetric and more stringent. Formally, for a pair of goods p and p' we define proximity as:

$$\phi_{pp'} = \frac{\sum_c M_{cp} M_{cp'}}{\max\{k_{p,0}, k_{p',0}\}}.$$

As results, we obtain a squared matrix of product proximities. Obviously, this one-mode projection is very dense and may contain links that are not significant (the co-export probability may be non zero, but very small). To create the final Product Space we employ the following further strategies.

First, we want the visualization of the Product Space to be a connected network. By this, we mean avoiding islands of isolated products. The second criteria is that we want the Product Space to be relatively sparse. This is achieved by fixing the average number of links per node as not larger than 5 and results in a representation that can summarize the structure of the Product Space using the strongest 1% of the links.

To make sure the visualization of the product space is connected, we calculate the maximum spanning tree (MST) of the proximity matrix. MST is the set of links that connects all the nodes in the network using a minimum number of connections and the maximum possible sum of proximities. We calculated the MST using Kruskal's algorithm [156]. Basically the algorithm sorts the values of the proximity matrix in descending order and then includes links in the MST if and only if they connect an isolated product or two trees without creating a cycle. By definition, the MST includes all products, but the number of links is the minimum requested to have a single connected component.

The second step is to add the strongest connections that were not selected for the MST. In this visualization we included the top 1,006 connections satisfying our criterion. By definition a spanning tree for 774 nodes contains 773 edges. With the additional 1,006 connections we end up with 1,779 edges and an average degree of nearly 4.6.

After selecting the links using the above mentioned criteria we build a visualization using a Force-Directed layout algorithm in Figure 9.1.

We care about the structure of the product space because it affects the ability of countries to move into new products. Products that are tightly connected share most of the requisite capabilities. If this is the case, then countries that already have what it takes to make one product will find it relatively easy to move to the next ones. A highly connected product space, therefore, makes the problem of growing the complexity of an economy easier. Conversely, a sparsely connected product space makes it harder.

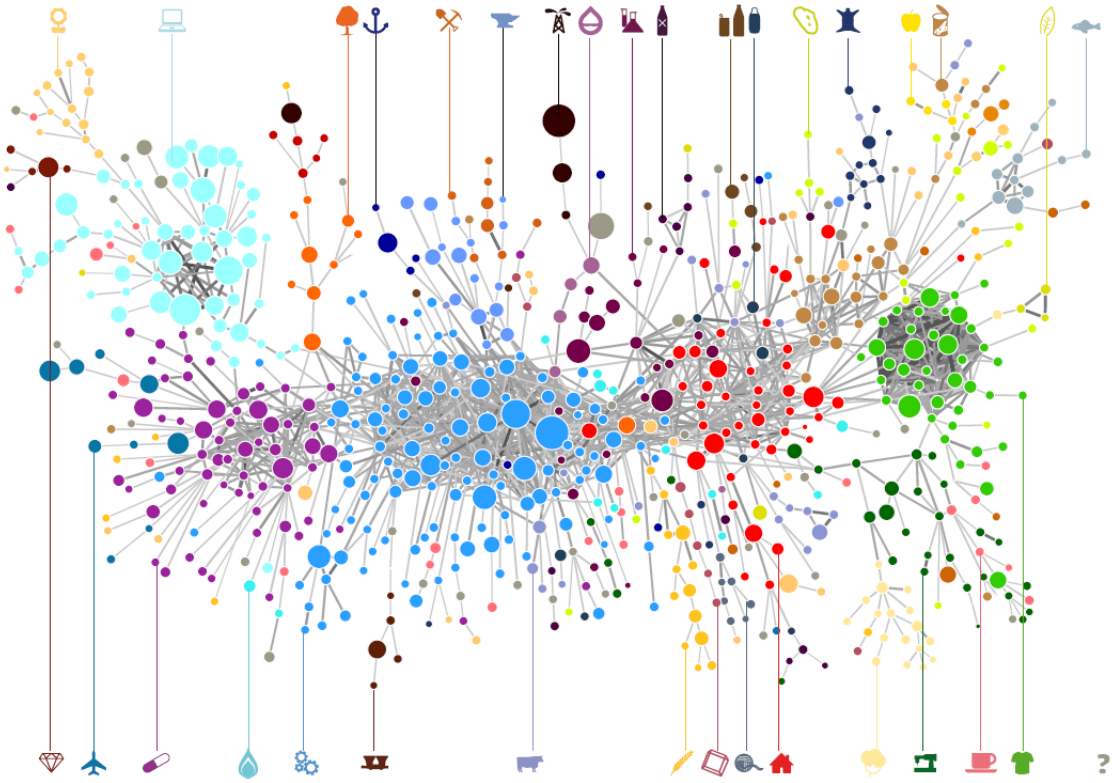


Figure 9.1: The Product Space.

Figure 9.1 reveals that the product space is highly heterogeneous. Some sections of it are composed of densely connected groups of products whereas others tend to be more peripheral and sparse. The product space gives us a glimpse of the embedded knowledge of countries by highlighting the productive capabilities they possess and the opportunities these imply. We can evaluate a country's overall position in the product space by calculating how far it is to alternative products and how complex these products are. We call this measure opportunity value and it can be thought of as the value of the option to move into more and more complex products.

Empirically, we find that countries move through the product space by developing goods close to those they currently produce [128]. This consideration makes the product space an effective tool to understand the dynamics and the evolution of ECI.

Even in this case, the improvements that may lie in a multidimensional analysis of the product space are evident. The product space itself is a complex network, but it is a simple monodimensional network. We can think about a multidimensional formulation of the product space, in this case using the exporting countries as dimensions. Employing this strategy, we do not need to project over countries to formulate $\phi_{pp'}$. The Shortest Path with Cost Modifiers approach described in Section 8.4 is useful to solve the problem of creating a multidimensional maximum spanning tree. Then, in this new multidimensional structure we may apply the statistical multidimensional tools developed in Section 7.1.

9.4 A Novel Product Categorization

The product space shows that many goods group naturally into highly connected communities. This suggests that products in these communities use a similar set of capabilities. We can identify communities because the products that belong to them are more closely connected to each other than to products outside of the community. The usefulness of communities is evident: they are an

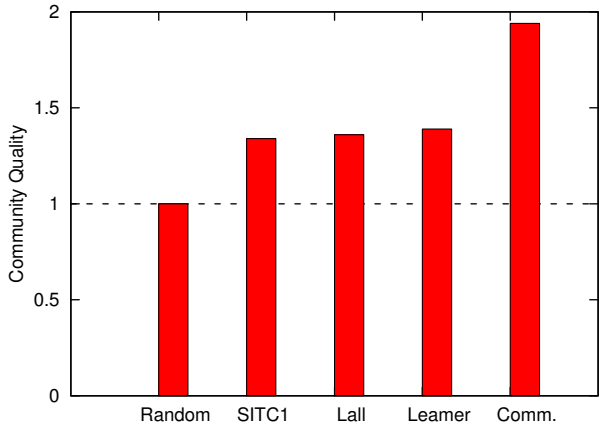


Figure 9.2: Community quality for five different ways of grouping products.

expression of the shared capabilities and they are a more efficient way to create product categories, because they are closer to what exactly happens in reality when producing a good.

We assign products to communities using the Infomap algorithm [227]. We described the operations performed by this algorithm in Section 2.3. The communities determined through this algorithm were manually named and merged into 34 communities. The nodes in Figure 9.1 are colored according to the community they belong.

We compare the ability of these communities to summarize the structure of the product space by introducing a measure of community quality. This is the ratio between the average proximity of the links within a community, and those connecting products from that community to products in other communities. To get a sense of the community quality we compare our assignment of products into communities with a baseline null model and three popular categorizations. The baseline null model is given by an ensemble of communities of the same size, where nodes have been assigned to each community at random. In this case, the average strength of the links within communities is equal to the average strength of links between communities, and the community quality is 1.

The three categorizations we use as comparators are: the first digit of the Standard International Trade Classification, the categories introduced in [167], based on factor intensities, and the technology categories introduced in [160]. All three classifications produce values of the community quality between 1.3 and 1.4, indicating that links within communities tend to be, on average, 30% to 40% stronger than those between communities. The communities we propose here have a community quality value of 1.94, indicating that the links between nodes in the same community are, on average, 94% stronger than those connecting nodes between communities (Figure 9.2). The difference in community quality of our proposed community system and that of the three alternative categorizations is highly statistically significant with a p-value $< 10^{-30}$.

Assuming a multidimensional product space, as suggested at the end of the previous section, also in this case multidimensional network analysis can lead to further advancements. In particular, the community discovery algorithm to use cannot be monodimensional, because this will ignore part of the knowledge embedded in the multidimensional product space itself. An approach as the one described in Section 8.1 is needed, or even the development of a novel truly multidimensional algorithm.

9.5 Applications

We now provide some brief evidences, suggesting the usefulness of Economic complexity in describing the dynamics of country growths, i.e. how much countries can grow. We present the evidence

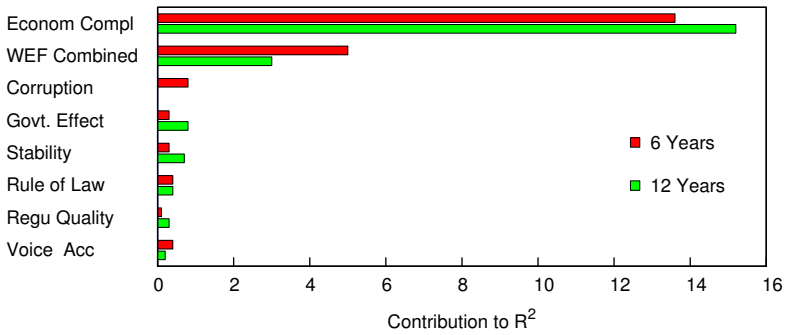


Figure 9.3: Contributions to the R square regression over the GDP growth of several different indicators.

collected so far, but better results may be achieved by employing a new multidimensional approach as described in the previous sections. To predict a country's future growth is a problem already addressed in the economics literature. Some of the most respected measures of institutional quality are the six Worldwide Governance Indicators from the World Economic Forum (WEF), published biennially since 1996.

The Economic Complexity Index aims to capture the same information. Which indicator better captures information that is more relevant for growth is an empirical question. Here we compare the contribution to economic growth of the Worldwide Governance Indicators and economic complexity by estimating a growth regression where all of the indicators and the Economic Complexity Index are used as explanatory variables. As controls we include the logarithm of per capita income, the increase in natural resource exports during the period and the initial share of GDP represented by natural resource exports. The contribution of each variable is estimated by taking the difference between the R^2 obtained for the regression using all variables and that obtained for the regression where the variable was removed.

Since the data from the World Economic Forum are available only since 1996, we perform this exercise using the 1996-2008 period as a whole and as two consecutive 6-year periods. We also compare with each individual WGI and with the six of them together. Figure 9.3 shows that the ECI accounts for 13.6 percent of the variance in economic growth during the 1996-2008 period, while the six WEF indicators combined account only for 5 percent. For the estimation using the two six year periods, we find that ECI accounts for 15.2% of the variance in growth, whereas the six WEF indicators combined account for 3%.

We conclude that as far as future economic growth is concerned, the Economic Complexity Index captures significantly more growth-relevant information than the six World Governance Indicators, either individually or combined. This suggests that the aspects of governance important for growth are weakly reflected in the WGIs and appear to be more strongly reflected in the economic activities that thrive in each country. These may be more effectively captured by the Economic Complexity Index.

Chapter 10

Study of Subject Themes in Classical Archaeology

In this section we tackle a general analytical problem to demonstrate the usefulness of multidimensional network analysis in a real world scenario that does not necessarily involves the use of complex network analysis to be solved. If multidimensional networks are able to help in this case study, and are used to unveil novel analytical insights, we can reasonably conclude that the practical application of the framework created in the second part of this thesis is indeed useful. The scenario in which we want to operate is the study of subject themes in classical archaeology.

In classical archaeology, or the arts and humanities in general, citation indices are of limited use and literature is still not fully available in digital form. Researchers still rely more on traditional subject classification as other fields do. A major pain point in exploring the respective classified literature is that scholars are usually limited to relatively simple user interfaces, where they can search or query for simple lists of literature associated to sets of classifications, or hop back and forth between classifications and publications while browsing the results. In the meantime the complex ecology of classification criteria related to each other remains opaque. Combining complex network analysis and data mining techniques in this paper, we offer a solution to this problem, enabling the exploration of a subject classification system, both on a meso as well as on a global level. Beyond standard user interface functionality, we are able to create a browsable set of visualizations, with which the interested scholar can explore neighboring sub-fields as well as the structure of the discipline as a whole, in a way that is more up to date and contextually superior to any written text book, as the big picture emerges algorithmically from an abundance of data that is accumulated by many actors. The tools we are developing make great use of complex networks and, in particular, of multidimensional networks. As our example we use Archäologische Bibliographie, i.e. a bibliographic database that collects and classifies literature in classical archaeology since 1956 [234]. Analyzing the state of 2007, our source data includes about 370.000 classified publications by circa 88.000 authors that are connected to about 45.000 classification criteria, via 670.000 classification links. Figure 10.1 shows a data model sketch of the database, including two additional link types which we construct within our analysis. First we generate and analyze a classification co-occurrence network from the classification link between publications and classification criteria. Second we abstract further by shortcutting from classifications to persons, resulting in an alternative perspective on classification cooccurrence in authors. Both derived structures are multidimensional networks, in which the dimensions represent different temporal snapshots.

That our problem is not trivial becomes evident by looking at the density of the classification co-occurrence network across publications. Its giant connected component includes about 29.000 classification criteria and over 200,000 co-occurrence links, with an average diameter of 2.7. Simple node-link diagrams of cooccurrence therefore are of limited use on a meso-level, resulting in a totally useless hairball on the global level [232]. This chapter is organized as follows: Section 10.1 indicates previous work. Section 10.2 details our analytical framework. Sections 10.3 and 10.4

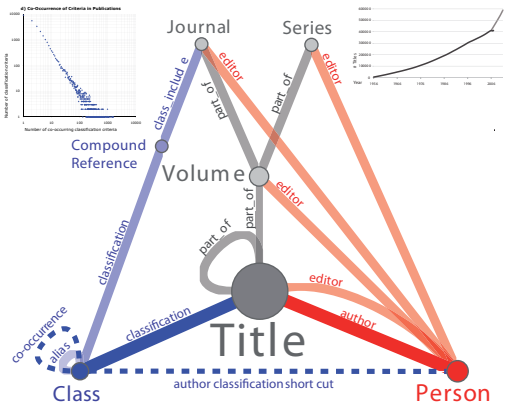


Figure 10.1: Data model sketch for Archäologische Bibliographie, including the fat-tail distribtion for classification co-occurrence in publications (upper left, see [232] for detail), and an indication of dataset growth from 1956 to 2011 (upper right).

present exemplary global as well as meso-level results respectively. Section 10.5 concludes this chapter.

10.1 Previous Work

This chapter builds on previous work [232], in which Schich et al. focus on both the system of classification criteria and the bipartite network of publication-classification in Archäologische Bibliographie. Already discussing thematic subdivisions in the so-called tree of subject headings, classification occurrence frequency, co-occurrence, and persistence in literature, they bring evidence for abundant heterogeneity in the system resulting in fat-tail distributions spanning five to six orders of magnitude (see Figure 10.1 in the upper left) - in fact legitimizing our perspective using approaches taken from the science of complex networks. In particular our method makes use of several different techniques borrowed from complex network analysis and data mining. Firstly, we use multidimensional networks, that were not used in the previous work. Then, we use an algorithm [12] taken from the area of network community finding that we explored in Section 2.3, combining it with an criterion for filtering dense networks in an intelligent way [236]. Regarding the area of data mining and learning, our paper furthermore makes use of an established technique extracting association rules [11, 132] in order to produce a sense-making lift-significance weight in addition to regular co-occurrence. As an alternative to association rules one could also apply a weighting scheme such as TF-IDF [229, 225], which we have avoided as larger background corpuses would have been hard to apply in our case, with classification criteria not being single ngrams, but branches of in part multilingual phrases within the strong tree of subject headings, where the very same term, such as a country name, can appear in multiple places within the hierarchy. For visualization we made use of Cytoscape [1].

10.2 Method

In terms of method this paper centers on the pipline depicted in Figure 10.2. Starting from a given source dataset, that is a bipartite classification network, it includes (a) a multidimensional one-mode projection from object-classification to classification co-occurrence, using the temporal information as dimensions; (b) the creation and visualization of rule-mined directed lift-significance

[1] http://www.cytoscape.org/

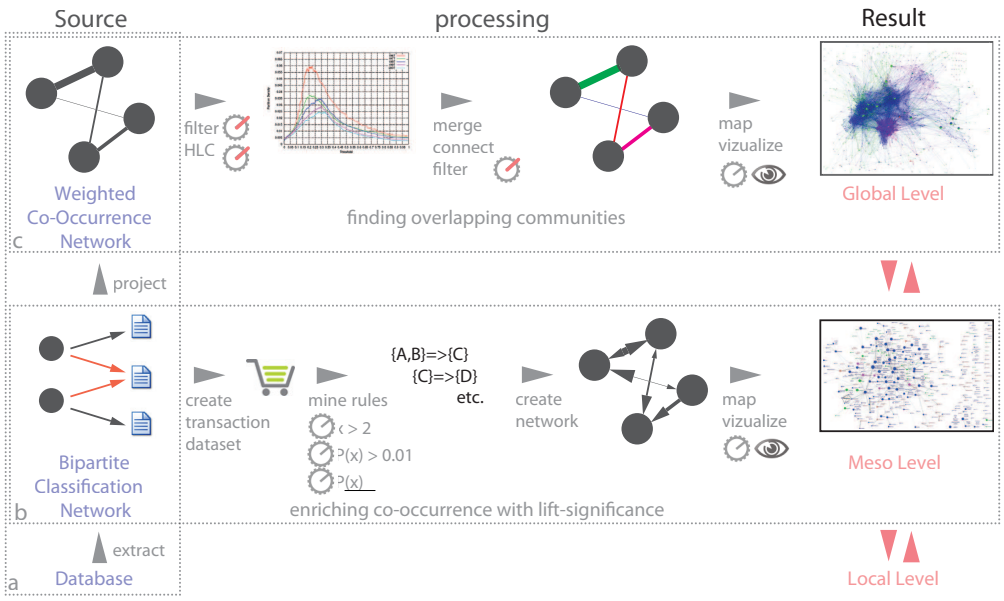


Figure 10.2: Data preparation, analysis, and visualization pipeline as described in Sections 10.2.1 to 10.2.3, including (a) the one-mode multidimensional projection from publication-classification or author-classification to classification co-occurrence, (b) the creation and visualization of rule-mined directed lift significance link weights in addition to regular co-occurrence weights, and (c) the creation and visualization of the multidimensional link community network, using Vespignani-filtering and Hierarchical Link Clustering.

link weights in addition to regular co-occurrence weights; and (c) the creation and visualization of a multidimensional link community network, using Vespignani-filtering, Hierarchical Link Clustering HLC and the node-types as dimensions (more detail in the following sections).

In our work we use the full pipeline in Figure 10.2 on five source dataset snapshots as derived from Archäologische Bibliographie, cumulating from 1956 to each full decade until 2007. Each snapshot is logically considered as a separate dimension for the network, as in the case of GTD, DBLP-Y and IMDb networks. We do this for both, classification co-occurrence in publications as well as classification co-occurrence in authors summing up to ten source dataset snapshots in total. In addition to the main pipeline in Figure 10.2, we also perform the era-discovery procedure on the full publication dataset from 1956-2007, as described in Section 7.2, verifying our arbitrary decision to cumulate decade by decade. Finally we also connect communities resulting from the pipeline in Figure 10.2c across decades. In a more formal way the problem we solve with this pipeline can be defined as follows:

Definitions

Given a bipartite classification network of objects and classification criteria, (1) while aiming for meso-level exploration, construct a weighted network of classification cooccurrence, enriching it with a useful significance measure, which is mined using information inherent in the source network itself, and (2) while aiming for global-level exploration, algorithmically extract sense making communities from the constructed classification co-occurrence network, taking into account that classifications can belong to multiple communities, resulting in a community overlap network. Finally, given multiple snapshots of the co-occurrence network in time, (3) connect their respective

community overlap network, enabling the exploration of their evolution in time.

In formal terms, our analysis starts with a set of objects O - i.e. in our case a set of publications or authors - and a set of associated classification criteria C. Elements $c \in C$ are related to objects $o \in O$ in a many-to-many fashion, meaning each classification can refer to many objects, while each object is potentially connected to many classifications. Both sets of classifications and objects grow over time. Therefore, we model our problem in the form of an evolving unweighted bipartite graph $G = \{O, C, S, E, T\}$, where (a) each classification c may belong to a particular classification superclass $s \in S$, representing the axiomatically discrete dimensions of Location, Person, Event, Period, or more general Subject Themes (we will see in the following sections as these superclasses will be translated into the network dimensions of the link community multidimensional network); (b) E is a set of triples (o, c, y), with y signifying a point in time at which the relationship between c and o has been created, logically modeled into the dimensions of G; (c) T is the set of pairs (c, s) which maps each classification c to its one and only one corresponding supertype s.

It is worthwhile noting that we apply our method to a single data source, while the problem definition given above is general, meaning it can also be applied to any other system that can be interpreted as a bipartite network of objects and classification criteria. Furthermore, losing only one degree of freedom, it is not mandatory that the system grows over time or supertypes are assigned to classifications.

Below the method is explained in more detail. Following data preparation (Section 10.2.1) we split our main analysis pipeline in two: part one finds overlapping communities of classifications (Section 10.2.2), resulting in a global level abstraction of our system; part two enriches co-occurrence with a directed lift-significance weight (Section 10.2.3), refining meso-level exploration. We conclude with the optional era-finding and snapshot connection (Section 10.2.4 & 10.2.5).

10.2.1 Data Preparation

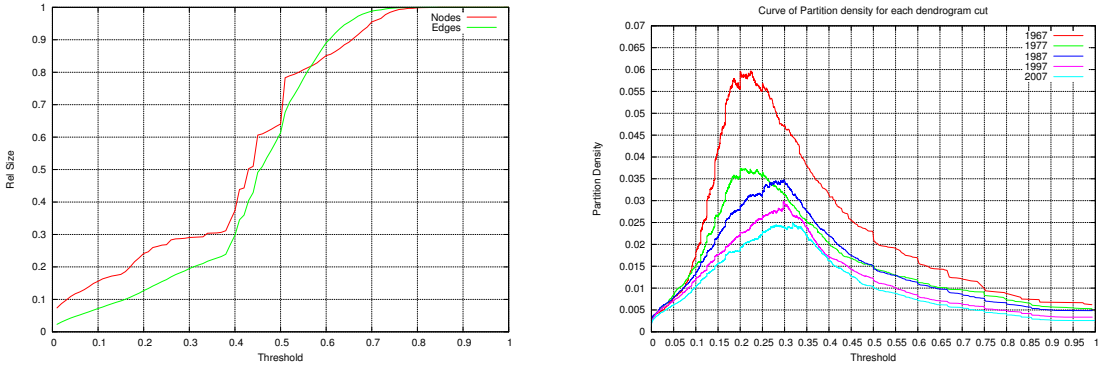
Regarding data preparation we follow the pipeline in Figure 10.2a, starting from a bipartite classification network extracted from a source database, as formalized above. For the meso-level pipeline (Section 10.2.3) we transform the edgeset E into a transaction dataset where each line takes the form $(o, y, c_1, c_2, ..., cn)$. In other words, each object o is handled as a transaction in a transactional dataset containing the list of its classifications as items, resulting in a list of adjecency lists for all $o \in O$.

For the final visualization in the meso-level pipeline and the global level pipeline (Sections 10.2.2 & 10.2.3) we project our bipartite or two-mode classification network to a one-mode multidimensional network of classification co-occurrence. Projecting to the set of classifications C, here results in a weighted undirected graph $G' = (C, E', D)$, where E' is a set of triples (c_1, c_2, w, y) and w is the number of objects attached to both c_1 and c_2 in the original bipartite graph G, and y is the point in time in which the edge appear.

As we are interested in co-occurrence evolution, but our implemented pipelines are not defined for evolving data, we filter our source data, producing a number of discrete temporal snapshots, that cumulate from the beginning of source dataset to a selected point time. More formally, for each snapshot $d \in D$ a dimension will be created, in which all $(c, o, y) \in d$ will respect the condition $y \leq d$ and all edges respecting this condition are included in d. For d we arbitrarily choose cumulating to each full decade of our example dataset, while we also address finding optimal set of d (in Section 10.2.4), and connecting multiple d (in Section 10.2.5).

10.2.2 Finding Overlapping Communities

For global level exploration we follow the pipeline in Figure 10.2c, where we aim to provide a big picture that exposes overlapping community structure as expected to be inherent in the network of classification co-occurrence. Not enforcing classifications to belong to a single community we eventually want to build and visualize a multidimensional community network, where links signify at least one shared classification in a particular superclass.



(a) (b)

Figure 10.3: (a) Relative number of nodes and edges size for different filtering thresholds. (b) Partition density values for each dendrogram cut threshold for each decade. Higher values means denser partition, i.e. a better community division.

Starting from the weighted one-mode projection of our bi-partite classification graph (Section 10.2.1) we want to apply an overlapping community discovery technique. Before we do so however, we have to deal with the extreme density of our one-mode projection, which is expected especially for bipartite classification graphs, caused by hubby objects and authoritative classification criteria. In order to get around this problem, we apply a statistical filter. Instead of a simple threshold on the edge weights, we apply a sophisticated network backbone extraction technique [236], that takes into account that in weighted networks many nodes have only low-weight connections, causing them to disappear in a naive threshold filtering. Instead of deleting all edges with a weight less than a particular value and consequently many nodes, network backbone extraction in ideal cases preserves 90% of the nodes while reducing the number of edges to 50% or lower (see Figure 10.3a). To do so, for each edge (i, j) the weight is recomputed - two times for both nodes it is attached to - according to the following formula:

$$\alpha_{ij} = 1 - (k-1) \int_0^{p_{ij}} (1-x)^{k-2} \, \mathrm{d}x$$

where k is the degree of i (or j), and p_{ij} is the normalized weight of the edge, according to the total weight of node i (or j). Those edges for which $a_{ij} \leq a$, i.e. which pass the significance test according to the threshold, are preserved in the network. This technique is not multidimensional, for this reason we apply this statistical filter on the network dimensions take separately.

From the filtered co-occurrence network we can now extract communities. A recent approach to obtain an overlapping graph partition is to perform the community discovery on the edges instead of the nodes themselves [12, 89]. From the given options we chose to apply Hierarchical Link Clustering HLC [12] as this method turned out to produce the most useful results. HLC first uncovers the hierarchical structure of the link communities in a complex network, where communities composed of a single link are recursively merged until the network itself composes one giant community. Meaningful communities are then extracted, by cutting the community dendrogram. Please note that also HLC algorithm is not defined for multidimensional networks. Again, we apply HLC on each dimension of the network taken separately. Deciding for a meaningful cut, modularity [198] is widely used to evaluate the quality of a partition. However as this is not well defined when including overlap, plus some other drawbacks (such as the resolution limit [93]), we follow [12] evaluating the quality of each partition using the partition density D score, which is (given a partition p returning a set of link communities LC):

$$D(p) = \frac{2}{|E'|} \sum_{lc \in LC} |E'|_{lc} \frac{|E'|_{lc} - (|C(lc)| - 1)}{(|C(lc)| - 2)(|C(lc)| - 1)}$$

where $|E'|$ is the total number of edges in the dimension, and $|C(lc)|$ and $|E'|_{lc}$ are the numbers of nodes and edges in lc in the dimension respectively. The higher $D(p)$, the better the partition p identifies well divided clusters in the network.

Figure 10.3b reports the evolution of the partition density for all possible dendrogram cuts in our co-occurrence network in publications for each decade (i.e. our dimensions $d \in D$). For each dimension, choosing the given optimal partition p, we now obtain a set of overlapping communities LC, allowing us to produce the desired global level picture of our classification network. In order to do so, we collapse each $lc \in LC$ into a single node, connecting the nodes of this network with links, whose weight is proportional to the number of nodes shared by the two communities. As each node and edge has a complex internal structure derived from the weight of the classification supertypes of all $c \in lc$, we can further enrich both nodes and edges in the representation of the resulting community overlap network, by representing the nodes with pie diagrams and splitting the edges in different dimensions according to the inherent superclass frequency. As the community overlap network is again very dense, and we aim for a text-book-style global picture we apply the backbone filter again.

10.2.3 Lift Significance

For meso-level exploration we follow the pipeline in figure 10.2b. Here we aim to visualize our simple weighted co-occurrence network of classifications $c \in \mathcal{C}$ with a more sophisticated directed significance measure. In order to do so, we perform association rule mining [11] over our transaction dataset (as introduced in Section 10.2.1), mining for frequent rules of co-classifications. Minimum support and confidence thresholds may be tuned depending on the phenomenon one is interested to highlight. As a result we obtain a set R of rules in the form $P(\mathcal{C}) \Rightarrow c$, where $P(\mathcal{C})$ is the set of all subsets of $\mathcal{C}$, excluding $\emptyset$. Using this result, we are able to build our significance network in which the nodes are the classifications $\mathcal{C}$, and the edges are triples $(c_1, c_2, w(c_1, c_2))$, where $w(c_1, c_2)$, i.e. the significance of the relationship between c_1 and c_2 is defined as follows

$$w(c_1, c_2) = \sum_{\forall r \in R. c_1 \in P(\mathcal{C}) \wedge c = c_2} \frac{supp(P(\mathcal{C}) \cup c)}{supp(P(\mathcal{C})) \times supp(c)}$$

where $P(\mathcal{C})$ is the set of classifications in the left side of rule r, c is the classification in the right side of the rule r, and $supp(x)$ is the support of the set x of classifications inside the transactional dataset. In other words, w is the sum of the lift of all rules involving c_1 as one of the antecedents of the rule, and c_2 is the consequence. The lift measure as such is not directed, but since we are filtering rules according to their confidence, which is directed, it follows that $w(c_1, c_2) \neq w(c_2, c_1)$, resulting in a directed network. This means a situation may (and does) occur, in which c_1 is very significant in pointing to c_2, while c_2 is not so significantly pointing to c_1 (see Section 10.4.1).

10.2.4 Era Discovery

Neither the global- nor meso-level pipelines above take into account time. To study evolution therefore, it is necessary to discretize the evolving source network into temporal dimensions on which the pipelines can be applied separately - raising the question: how to choose the right snapshot size?

Looking for eras, i.e. periods of regular and predictable network evolution, we apply a method [40] (see Section 7.2) that calculates the Node and Edge Dimension Correlation between all consecutive observations of the network, resulting in the ability to define a distance measure between groups of observations. We refer mainly to Section 7.2 of this thesis for more details about the procedure.

Using this distance measure, computed on all adjacent observations, a dendrogram is built, grouping together consecutive observations, presenting regular evolution separated from abrupt changes in trend. Figure 10.4 depicts the respective dendrogram for classification co-occurrence in

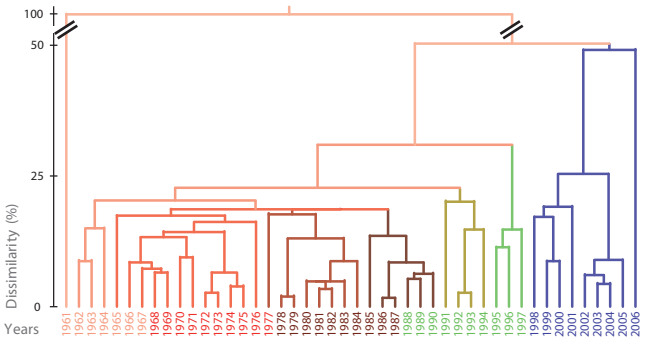


Figure 10.4: Era structure dendrogram of classification cooccurrence in publications of Archäologische Bibliographie according to [40] and Section 7.2. Eras are colored in the tree, while our arbitrary decades are highlighted in the x-axis labels.

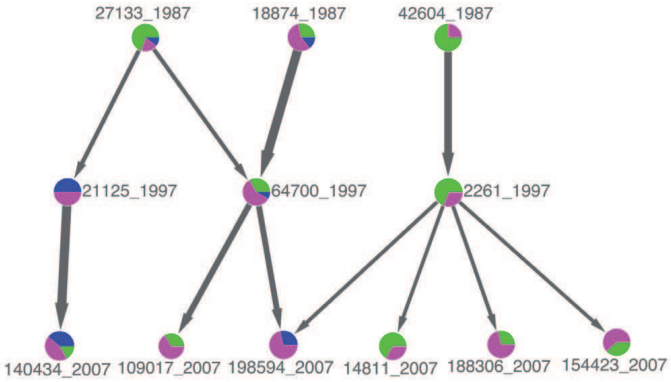


Figure 10.5: Communities belonging to various temporal snapshots are connected using a dedicated algorithm, revealing interesting merges and splits over time.

publications of Archäologische Bibliographie from 1956 to 2007, with our arbitrary decades fitting surprisingly nice to the found era structure.

10.2.5 Snapshot Connections

Finally, given the fact that our analysis is performed in separate pipeline for each decade or dimension, how can the dimension results be connected? In the meso-level case the solution is trivial: All classifications are uniquely identified and can therefore be connected across snapshots. For the global level this is not true since communities are calculated for each snapshot separately. So, given community A in dimension d and community B in dimension $d+1$, can we decide if A and B are related or not - i.e. if they are equivalent, if B forked from A, or B is a merge of A and C? In [90] the authors solve this problem with the concept of minimum description length, i.e. by using a data description language to produce the shortest data description possible (see Timefall algorithm from Section 2.3). In our case all communities are lists of classifications, where we can calculate the relative entropy between any community pair from one snapshot to another. The relative entropy takes values from 0 (where two communities share all classifications) to +1

Figure 10.6: Links in the community overlap network corresponding to subject themes, locations, and periods are distributed in a very different way.

(where the community overlap becomes zero). Calculating the relative entropy across dimensions, we can put weighted links from a community in dimension d to one or more communities in the subsequent dimension $d + 1$. The weight is inversely proportional to the relative entropy. Figure 10.5 shows an example result.

10.3 Global Exploration

As a result of processing our source data according to the pipeline in Figure 10.2c, we can explore the ecology of classifications in Archäologische Bibliographie on a global level, i.e. in form of an overlapping community network. Nodes in this network, as shown in Figure 10.6, represent a number of classifications belonging to the respective communities, with the amount of classifications indicated by node size. Links between the communities stand for the number of classifications that are shared between them. Every classification in our system, can therefore potentially be part of multiple nodes and links in the community network. That the found configuration of communities makes sense, becomes clear while zooming into the meso-level structure of our system further below in Section 10.4. First however, we take a look at some obvious features of the global community network.

One of the most striking features of the community overlap network in Figure 10.6 is that it is not a hairball, but a collection of tightly connected clusters that are interconnected in a semitight fashion and surrounded by a sparsely connected periphery. The usage of a multidimensional network for distinct edge types according to the superclass enriches the visualization, and it makes clear where the observed structure is rooted: every node in our community network is depicted as a pie chart indicating the presence of classification superclasses in the respective community blue for subject themes, green for locations, pink for periods, red for persons and black for objects and monuments. Even without knowing the detailed content of our communities it becomes immediately clear to the eye, that the superclasses are not distributed in a random way, but grouped into

genres defining the tightly connected clusters.

A community link containing three locations and four subject themes creates an edge in two dimensions, i.e. a green line of width three, and a blue line of width four. Figure 10.6 shows all the location, period, and subject theme dimensions in isolation for both co-occurrence of classifications in authors as well as publications according to the state of Archäologische Bibliographie in 2007. We can clearly see that subject classification dimension permeate throughout the whole community network, while period and location dimensions co-govern certain clusters. In other words, according to Archäologische Bibliographie, publications and - as clusters appear to be tighter and even better defined - even more so authors in classical archaeology seem to specialize roughly on certain genres, governed by an either spatial, temporal, or a more generic conceptual perspective.

Focusing on the evolution of the community overlap network, we have applied the data processing pipeline in Figure 10.2c five times each, cumulating classification data from 1956 for every decade from 1967 to 2007, both for classification co-occurrence in publications as well as authors. Keeping our variable threshold settings over the decades and using the same simple edge-weighted spring-embedded layout, we can see in Figure 10.7 that the colored cluster structure identified comes into existence in the form of a bare skeleton of a few connected communities very early on, fleshing out to massive more differentiated proportions over the decades. The smooth development seems to legitimate our arbitrary decision to split our dataset into five decades. The fairly accurate fit of the decades to the algorithmically extracted era structure of our data in Figure 10.4 further supports our choice. In summary we can say that the picture of community network evolution, or in other words classical archaeology according to Archäologische Bibliographie as a whole, does not feature large surprises - for e.g. in the form of significant phase transitions in node connectivity - but seems to grow in a smooth manner. If the smooth development reflects the evolution of classical archaeology as a discipline or is rooted in the attention towards literature on behalf of the curators of Archäologische Bibliographie, remains a subject of further investigation.

Zooming into the evolution of communities themselves, according to [90], reveals a more differentiated situation in detail. Looking at Figure 10.5 for e.g. we can see two communities 27133 and 18874, which over the decade from 1987 to 1997 merge into a single community 64700, approximately averaging the fraction of associated locations, subject themes and periods, only to split up into two separate communities 109017 and 198594 again by 2007, now concentrating periods and locations vs. periods and subject themes respectively curiously reflecting the often idiosyncratically perceived spat between excavation archaeologists and more art historically focused scholars spending most of their time in the library.

10.4 Meso Level Exploration

10.4.1 Co-Occurrence plus Lift-Significance

As a result of the pipeline in Figure 10.2b, we can explore the ecology of classifications in Archäologische Bibliographie on a meso-level, i.e. in form of a significance weighted co-occurrence network. Nodes in this network, as shown in Figure 10.8, are the classifications themselves, with node color signifying the classification superclass - i.e. subject themes, locations, periods, persons, or objects. Node size indicates the amount of literature or number of authors associated with the classification. Links connect co-occurring classifications. Line width is proportional to a simple co-occurrence weight, i.e. the amount of literature or number of authors shared by the two connected classifications. The line color depth reflects the lift significance measure introduced in Section 10.2.3, with light grey links carrying low significance vs. darker links being highly significant. While line color depth is only a simple sum of lift significance in both directions, the respective arrow heads at both ends of the line contain information about link symmetry. This is interesting, as co-occurrence usually turns out to be symmetrical, but sometimes is remarkably directed by nature.

Figure 10.8 presents a striking example showing all the properties mentioned above. It depicts co-occurrence in the branch Plastic Art and Sculpture i.e. a subset of classifications within the tree of subject headings in Archäologische Bibliographie. As in previous work [232] we threshold

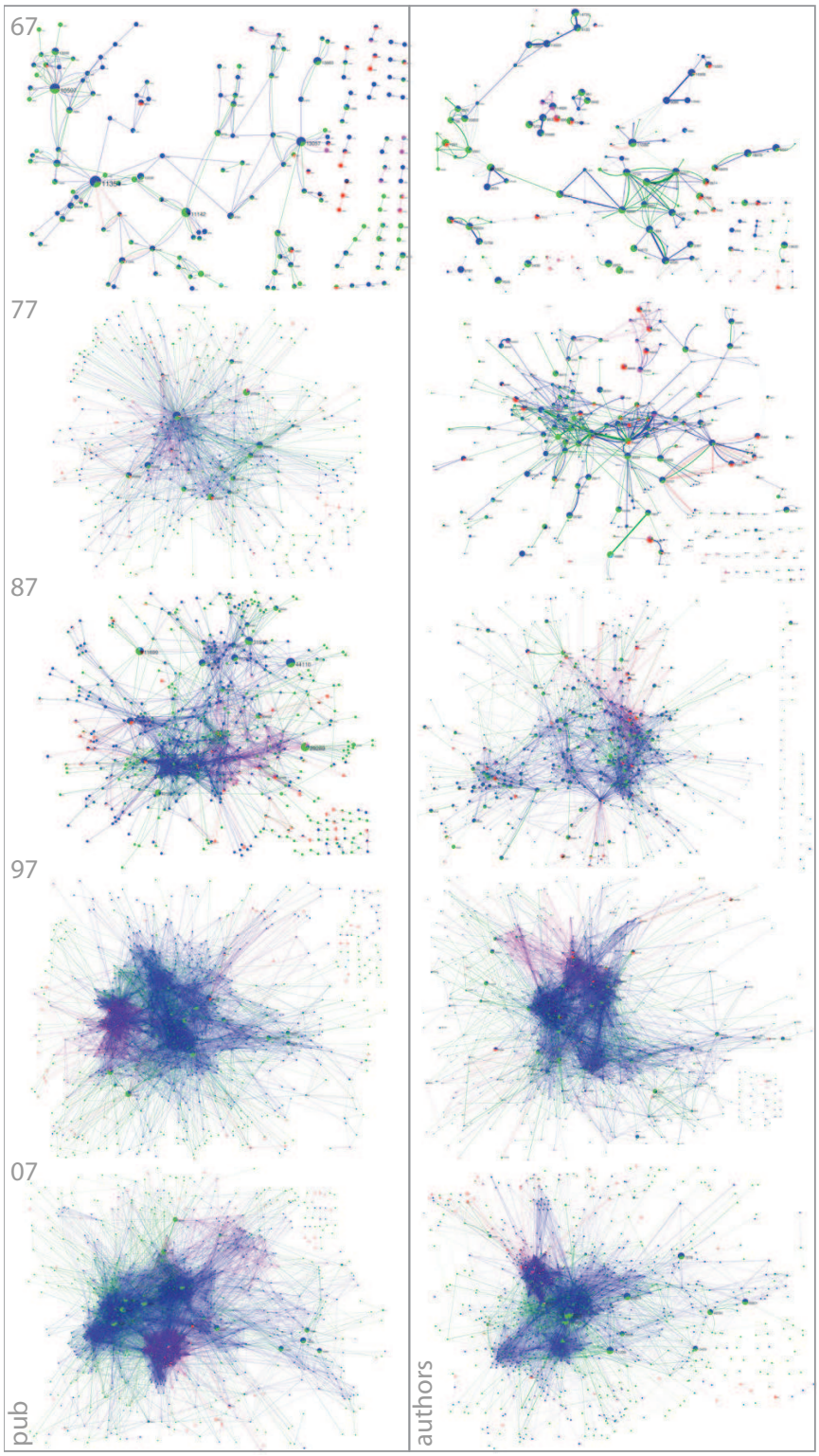


Figure 10.7: Both classification co-occurrence in publications as well as authors evolve over time, fleshing out structure that emerges early on in the process.

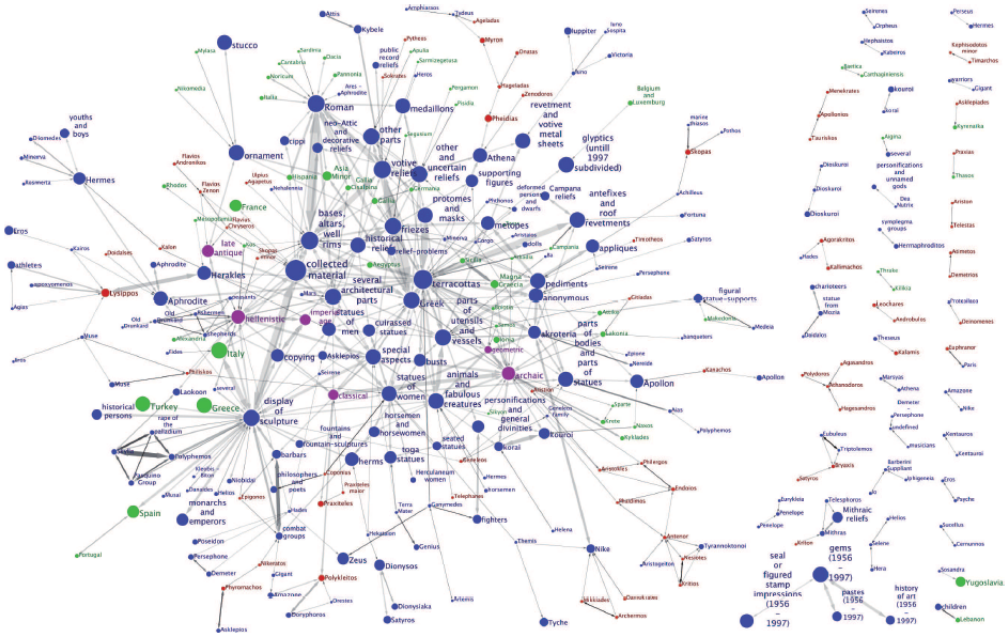


Figure 10.8: Classification co-occurrence (≥ 4) in publications with lift-significance (≥ 0.056) for the branch Plastic Art and Sculpture.

the subset, taking only links into account that contain at least four publications. Improving over the previous version however, we also add highly significant links containing as few as a single publication. As a threshold for lift significance we use a rule of thumb, taking into account as many significant links as highly co-occurrence ones, merging the two resulting thresholded networks to achieve the final figure.

It is interesting that the networks thresholded by heavy co-occurrence or high lift significance do not overlap much. In fact, when merged as in Figure 10.8 they turn out to complement each other: Greek and votive reliefs for example have a very strong connection in terms of co-occurrence without high significance, which in a sense is trivial, as any archaeologist would know that both classifications are highly related. Zeus and Ganymed on the other hand share less literature, but nevertheless their connection is highly significant and should therefore be part of the picture. In fact their relation is also asymmetrical, which makes sense as Zeus, the father of god and men, makes us think of many aspects, while Ganymed in sculpture is mostly depicted with Zeus in the form of an eagle. Taken together the networks of heavy occurrence and high lift significance result in a kind of cheat sheet for Plastic Art and Sculpture, where we can easily see what is often related to each other or rare and significant. Similar pictures as in Figure 10.8 can be produced for any given branch of classifications in the tree of subject headings, and also, as we will see below, for more sophisticated selections of classification criteria. Before we go into detail however, lets also take a look at network evolution.

As on the global level, looking at network evolution also makes sense on the meso scale. Besides the obvious growth regarding the number of classifications, and as a consequence their respective cooccurrence links, there is one particular phenomenon striking the eye in Figure 10.9, which shows a detail of the network in Figure 10.8 evolving from 1967 to 2007. As becomes clear over the decades, significant links tend to accumulate literature, while losing significance. In other words as the association starts to be taken for granted the link line widens and becomes more light in color, as we can see for the links between Nike and akroteria, or kouroi and korai in Figure 10.9. Of course, as with link symmetry, the effect shows interesting exceptions such as the highly significant clique of Polyphemos, Skylla, Pasquino Group, and rape of the palladium that we can spot on the left side periphery in Figure 10.8. Given the spectacular uniqueness of the sculptures in question

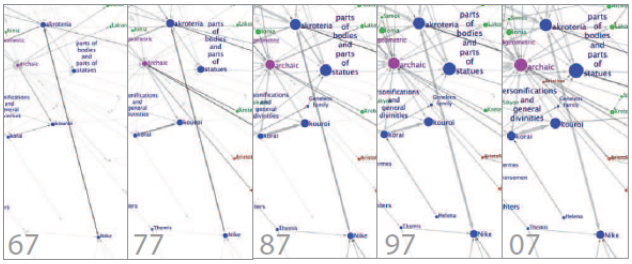


Figure 10.9: Classification co-occurrence evolution clearly shows that initially highly significant, i.e. dark links become less significant and wider as they accumulate literature.

Figure 10.10: Mutual self-definition of Names Portraits.

and the related controversial discussion in the literature, it is not a surprise that the associated links stayed significant over four decades while accumulating more and more literature.

Another interesting phenomenon on the meso level is the mutual self-definition of classifications across co-occurrence links. In previous work [232] we have already mentioned some striking examples for Plastic Art and Sculpture regarding this effect. Here we present another example that highlights the inherent potential: For Figure 10.10 we chose all classifications in the branch Named Portraits (across publications in 2007), thresholding both co-occurrence ≥ 2 and lift-significance ≥ 0.06 in a minimal way. Again the figure, which only shows the largest connected component of the result, can be used as a cheat sheet, indicating the relations of portraits from Augustus, to Phillippus Arabs at the end of the Roman empire, with lift significance highlighting relations between strongly connected types such as Caracalla, Septimius Severus and Geta. In general terms this means our approach provides easy access to highly specialized fields that are hard to explore using a regular user interface that browses bibliographic classifications on a local level. As similar insights can easily be produced for all areas covered by Archäologische Bibliographie, the respective visualizations call for being used to complement classic textbook introductions to classical archaeology.

10.4.2 Ego-Networks vs. Communities

An alternative starting point in exploring the ecology of classifications in our system - beyond picking predefined branches of the tree of subject headings - is to begin with a single classification of interest. Here, a seemingly obvious approach would be to draw the ego-network, meaning the

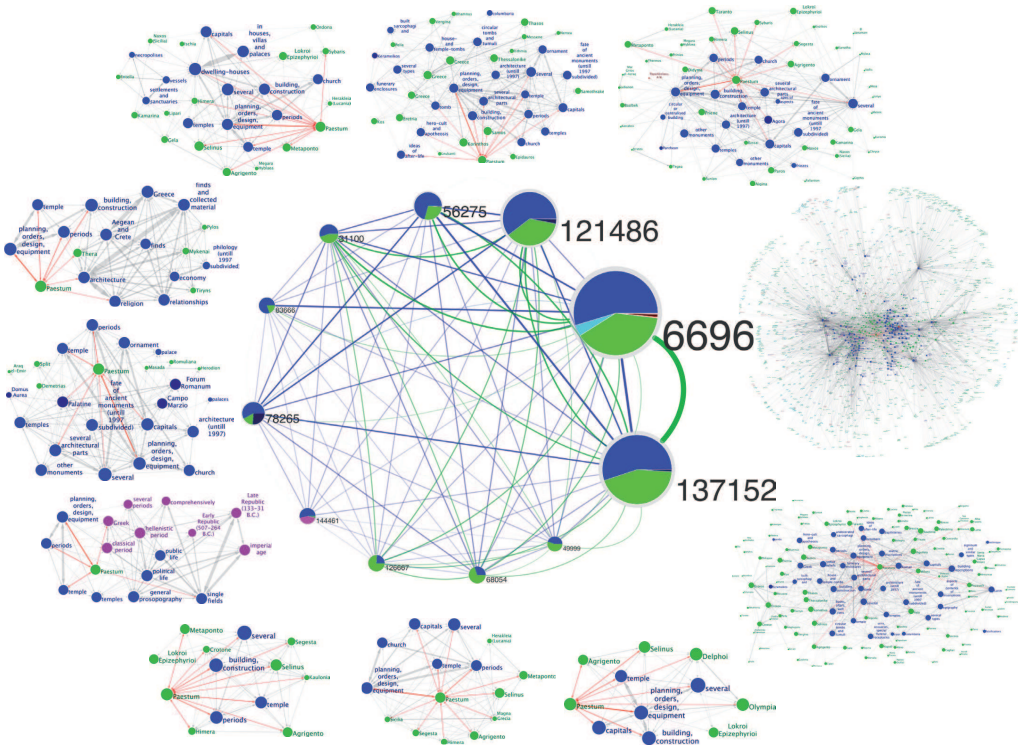


Figure 10.11: Combining global and meso-level exploration by zooming into overlapping communities containing a given classification - here Paestum - reveals its meaning even to the uneducated eye, improving significantly over simple ego-networks (see 10.4.2).

network of all links between classifications, the classification of interest is related to - in equivalence to the widespread basic diagrams of friendships between our own friends in popular social network platforms. Unfortunately the ego-network strategy does not work for our co-occurrence network, as the average network path length is only 2.7, making it very likely that the result contains an almost fully connected clique. An excellent example is the ego-network of Paestum - an important and popular archaeological site in Italy. Even worse than raw, thresholding has almost no effect on this structure: in fact if we threshold heavily for co-occurrence ≥ 25 - while lift significance is virtually irrelevant - the picture starts to get clearer, but we only isolate what could be called the generic Italian core of classical archaeology, where Paestum, even though popular, only appears in the very periphery of a large cluster, connected to a few even more peripheral events, and the obvious fact that it is known for temples.

The solution to the problem of dense ego-networks is to harness our global level community overlap network, from which we can pick all communities in which Paestum appears. Looking into those communities on a meso level it turns out, we can learn in a very straight forward way what Paestum is really about. Figure 10.11 shows the relevant section of the global community overlap network, surrounded with the meso level co-occurrence networks for the respective communities. We can see that the community size distribution is heterogeneous. Let us look into some of them: Community 6696 already improves over the basic ego-network, as it embeds Paestum into the core of classic archaeology including relevant classifications that are more than one hop away. The smallest communities such as 68054 tell us that Paestum is about temple, capitals, planning orders, building construction, similar to a couple of strikingly related sites. Community 144461 dates Paestum to the Greek period, again as a striking example for temples. Community 78265

provides a hint that architectural parts from Paestum were reused later in Roman buildings such as the Palatine palaces. Community 137152 finally provides a wider context of Paestum, also including tombs, pointing to literature of the famous tomb of the diver among others in sum a pretty accurate description of what Paestum is about, accessible in an easy way, even to the non specialist.

10.5 Conclusion

Summing up, we have presented a way to explore a complex system of subject classification co-occurrence, by combining network filtering, community finding, association rule mining and multi-dimensional networks both in the representation of the data (the global level community network) and in the analysis (the discovery of the eras of our evolving network). As a result we can now explore Archäologische Bibliographie on three levels. To the standard local level user interface we have added a meso-level network of significance-weighted co-occurrence that allows us to explore the regional neighborhood of individual (groups of) classifications. Furthermore we also provide a global level community overlap multidimensional network, that allows us to grasp the big picture of classical archaeology in an intuitive way.

Chapter 11

Conclusion and Future Works

In this thesis, we have introduced multidimensional network analysis, a novel framework for the analysis of complex networks where the interactions between entities can be labelled with different types, or observed from different dimensions. We organized the presentation of multidimensional network analysis in three parts.

Firstly, we defined what a multidimensional network is; what are the alternative representations and the works that different authors already presented in the literature; and where it is possible to find multidimensional networks in the real world. Secondly, we presented our contribution to multidimensional network analysis, expressed by extending the network measures defined for simple graphs; proposing novel measures meaningful only in the multidimensional setting; and developing more complex analytical solutions to advanced network problems, with particular attention to our main case study: the community discovery. Finally, we defined two different real world problems not necessarily related to network theory: the analysis of international trade and the problem of exploring literature in classical archaeology. In the first case we show how it is possible to introduce multidimensional network analysis to enrich a problem tackled with monodimensional networks; in the second scenario we show how we can develop a solution to a general problem by using multidimensional network analysis.

The future research directions of this thesis can be mainly divided in two tracks. The first track has been proposed in the related work section. There are several different models trying to grasp the complex setting of multiple different relations and interactions in a complex network. To study the relationships between layered interdependent networks, tripartite hypergraphs and multidimensional networks is an important future development. A second track of research is open from the second part of this thesis. For example, in the multidimensional community discovery section, a method for characterizing multidimensional communities has been proposed. But the algorithm described to extract them is an adaptation of a classical monodimensional community discoverer. To develop a truly multidimensional community discovery algorithm, able to take advantage of the proposed characterization of communities is still an open problem. And this statement holds true for the other analysis, both for the multidimensional measures and the other problems, proposed in this thesis.

Multidimensional link prediction has been proved to be a more complex problem that cannot be tackled with a simple combination of known techniques and multidimensional score modifiers. A technique that combines multidimensional network analysis and the real world properties of the different dimensions creating the multidimensional structure is needed. For example, a mobility network with different modes of transportation provides different working mechanism than the ones of a multirelational social network.

Also the contribution for the multidimensional shortest path problem is here represented more in the formalization of the problem posed by the introduction of cost modifiers than in its solution. More research on heuristics, optimization strategies and experiments is needed.

Of course, this big and novel analytical section has one main end: to be able to better represent, and to provide better tools for the analysis of, real world interacting phenomena. The

underlying meta-objective is to grasp complex interactions from real world and to translate them into knowledge. An example is provided in the third part of the thesis and it is our mission to make clear that this is just one of the many instances of what can be done with multidimensional networks. The final end is to populate the scientific world with better tools, and we believe that multidimensional networks should be part of the workbench of any complex system scientist. Many other case studies of multidimensional networks can be identified.

Bibliography

[1] Evrim Acar, Daniel M. Dunlavy, and Tamara G. Kolda. Link prediction on evolving data using matrix and tensor factorizations. In *LDMTA'09: Proceeding of the ICDM'09 Workshop on Large Scale Data Mining Theory and Applications*, pages 262–269. IEEE Computer Society Press, December 2009.

[2] Lada Adamic and Eytan Adar. Friends and neighbors on the web. *Social Networks*, (25):230, 2001.

[3] Lada A. Adamic and Bernardo A. Huberman. Power-law distribution of the world wide web. *Science*, 287(5461):2115a–, 2000.

[4] Lada A. Adamic, Rajan M. Lukose, and Bernardo A. Huberman. Local search in unstructured networks. *Handbook of Graphs and Networks*, pages 295–317, 2003.

[5] Lada A. Adamic, Rajan M. Lukose, Amit R. Puniyani, and Bernardo A. Huberman. Search in power-law networks. *CoRR*, cs.NI/0103016, 2001.

[6] Nitin Agarwal, Huan Liu, Lei Tang, and Philip S. Yu. Identifying the influential bloggers in a community. In *WSDM '08: Proceedings of the international conference on Web search and web data mining*, pages 207–218, New York, NY, USA, 2008. ACM.

[7] Sumeet Agarwal, Charlotte M. Deane, Mason A. Porter, and Nick S. Jones. Revisiting date and party hubs: Novel approaches to role assignment in protein interaction networks. *PLoS Comput Biol*, 6(6):e1000817, 06 2010.

[8] Agarwal, G. and Kempe, D. Modularity-maximizing graph communities via mathematical programming. *Eur. Phys. J. B*, 66(3):409–418, 2008.

[9] Charu C. Aggarwal, Jiawei Han, Jianyong Wang, and Philip S. Yu. A framework for clustering evolving data streams. In *VLDB '2003: Proceedings of the 29th international conference on Very large data bases*, pages 81–92. VLDB Endowment, 2003.

[10] Charu C. Aggarwal, Jiawei Han, Jianyong Wang, Philip S. Yu, T. J. Watson, and Resch Ctr. A framework for projected clustering of high dimensional data streams. In *Proc. of VLDB*, pages 852–863, 2004.

[11] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. In *SIGMOD '93: Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, pages 207–216, New York, NY, USA, 1993. ACM.

[12] Yong-Yeol Ahn, James P. Bagrow, and Sune Lehmann. Link communities reveal multi-scale complexity in networks. *Nature*, (466):761–764, 2010.

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

[14] Edoardo M. Airoldi, David M. Blei, Stephen E. Fienberg, and Eric P. Xing. Mixed membership stochastic blockmodels. *JOURNAL OF MACHINE LEARNING RESEARCH*, 9:1981, 2007.

[15] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Rev. Mod. Phys.*, 74(1):47–97, January 2002.

[16] Andrés Gago Alonso, José E. Medina-Pagola, Jesús Ariel Carrasco-Ochoa, and José Francisco Martínez Trinidad. Mining frequent connected subgraphs reducing the number of candidates. In *ECML/PKDD (1)*, pages 365–376, 2008.

[17] Nelson Augusto Alves. Unveiling community structures in weighted networks. *Physical Review E*, 76:036101, 2007.

[18] A. Arenas, J. Duch, A. Fernandez, and S. Gomez. Size reduction of complex networks preserving modularity. *New Journal of Physics*, 9:176, 2007.

[19] Lars Backstrom, Cynthia Dwork, and Jon Kleinberg. Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 181–190, New York, NY, USA, 2007. ACM.

[20] Lars Backstrom, Dan Huttenlocher, Jon Kleinberg, and Xiangyang Lan. Group formation in large social networks: membership, growth, and evolution. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 44–54, New York, NY, USA, 2006. ACM.

[21] B.W. Bader, R.A. Harshman, and T.G. Kolda. Temporal analysis of semantic graphs using asalsan. *ICDM*, pages 33 –42, oct. 2007.

[22] Jim Bagrow and Erik Bollt. A local method for detecting communities. *Physical Review E*, 72:046108, 2005.

[23] Per Bak and Kim Sneppen. Punctuated equilibrium and criticality in a simple model of evolution. *Phys. Rev. Lett.*, 71(24):4083–4086, Dec 1993.

[24] B. Balasundaram, S. Butenko, I.V. Hicks, and S. Sachdeva. Clique relaxations in social network analysis: The maximum k-plex problem. In *Operations Research*, 2009.

[25] B. Ball, B. Karrer, and M. E. J. Newman. An efficient and principled method for detecting communities in networks. *ArXiv e-prints*, April 2011.

[26] Arindam Banerjee, Sugato Basu, and Srujana Merugu. Multi-way clustering on relation graphs. In *SDM*. SIAM, 2007.

[27] Arindam Banerjee, Inderjit Dhillon, Joydeep Ghosh, Srujana Merugu, and Dharmendra S. Modha. A generalized maximum entropy approach to bregman co-clustering and matrix approximation. In *In KDD*, pages 509–514, 2004.

[28] Arindam Banerjee, Srujana Merugu, Inderjit Dhillon, and Joydeep Ghosh. Clustering with bregman divergences. In *Journal of Machine Learning Research*, pages 234–245, 2004.

[29] A. L. Barabasi, H. Jeong, Z. Neda, E. Ravasz, A. Schubert, and T. Vicsek. Evolution of the social network of scientific collaborations. *PHYSICA A*, 311:3, 2002.

[30] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, October 1999.

[31] M. J. Barber. Modularity and community detection in bipartite networks. *arXiv*, 76(6):066102–+, December 2007.

[32] Jeffrey Baumes, Mark Goldberg, and Malik Magdon-ismail. Efficient identification of overlapping communities. In *In IEEE International Conference on Intelligence and Security Informatics (ISI*, pages 27–36, 2005.

[33] Jeffrey Baumes, Mark K. Goldberg, Mukkai S. Krishnamoorthy, Malik Magdon-Ismail, and Nathan Preston. Finding communities by clustering a graph into overlapping subgraphs. In *IADIS AC*, pages 97–104, 2005.

[34] Jeffrey Baumes, Mark K. Goldberg, Malik Magdon-Ismail, and William A. Wallace. Discovering hidden groups in communication networks. In *ISI*, pages 378–389, 2004.

[35] Tanya Y. Berger-Wolf and Jared Saia. A framework for analysis of dynamic social networks. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 523–528, New York, NY, USA, 2006. ACM.

[36] M. Berlingerio, M. Coscia, and F. Giannotti. Finding and characterizing communities in multidimensional networks. In *ASONAM*. IEEE, 2011.

[37] M. Berlingerio, M. Coscia, F. Giannotti, A. Monreale, and D. Pedreschi. Foundations of multidimensional network analysis. In *ASONAM*. IEEE, 2011.

[38] Michele Berlingerio, Francesco Bonchi, Björn Bringmann, and Aristides Gionis. Mining graph evolution rules. In *ECML/PKDD (1)*, pages 115–130, 2009.

[39] Michele Berlingerio, Michele Coscia, and Fosca Giannotti. Finding redundant and complementary communities in multidimensional networks. In *CIKM*, pages 2181–2184, 2011.

[40] Michele Berlingerio, Michele Coscia, Fosca Giannotti, Anna Monreale, and Dino Pedreschi. As time goes by: Discovering eras in evolving social networks. In *PAKDD (1)*, pages 81–90, 2010.

[41] Michele Berlingerio, Michele Coscia, Fosca Giannotti, Anna Monreale, and Dino Pedreschi. Towards discovery of eras in social networks. In *ICDE - M3SN*, 2010.

[42] Michele Berlingerio, Michele Coscia, Fosca Giannotti, Anna Monreale, and Dino Pedreschi. The pursuit of hubbiness: Analysis of hubs in large multidimensional networks. *Journal of Computational Science*, 2(3):223 – 237, 2011.

[43] Michele Berlingerio, Michele Coscia, Fosca Giannotti, Anna Monreale, and Dino Pedreschi. Evolving networks: Eras and turning points. In *IDA DyNak Journal*, pages 81–90, To appear.

[44] Indrajit Bhattacharya and Lise Getoor. A latent dirichlet model for unsupervised entity resolution. In *SDM*, 2006.

[45] Åke Björck. *Numerical methods for least squares problems*. SIAM, 1996.

[46] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

[47] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *J.STAT.MECH.*, page P10008, 2008.

[48] Stefan Boettcher and Allon G. Percus. Optimization with extremal dynamics. *Phys. Rev. Lett.*, 86(23):5211–5214, Jun 2001.

[49] Paolo Boldi, Massimo Santini, and Sebastiano Vigna. A large time-aware web graph. *SIGIR Forum*, 42(2):33–38, 2008.

[50] Béla Bollobás, Oliver Riordan, Joel Spencer, and Gábor Tusnády. The degree sequence of a scale-free random graph process. *Random Struct. Algorithms*, 18(3):279–290, 2001.

[51] Francesco Bonchi, Fosca Giannotti, Alessio Mazzanti, and Dino Pedreschi. Examiner: Optimized level-wise frequent pattern mining with monotone constraints. *Data Mining, IEEE International Conference on*, 0:11, 2003.

[52] Karsten M. Borgwardt, Hans-Peter Kriegel, and Peter Wackersreuther. Pattern mining in frequent dynamic subgraphs. In *ICDM*, pages 818–822. IEEE Computer Society, 2006.

[53] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the 7th International Conference on World Wide Web*, 1998.

[54] Sergey Brin, Rajeev Motwani, Jeffrey D. Ullman, and Shalom Tsur. Dynamic itemset counting and implication rules for market basket data. In *SIGMOD '97*, pages 255–264, New York, NY, USA, 1997. ACM.

[55] Björn Bringmann, Michele Berlingerio, Francesco Bonchi, and Aristides Gionis. Learning and predicting the evolution of social networks. *IEEE Intelligent Systems*, 25(4):26–35, 2010.

[56] Bjorn Bringmann and Siegfried Nijssen. What is frequent in a single graph? In *PAKDD*, pages 858–863, 2008.

[57] Peter J. Brockwell and Richard A. Davis. *Introduction to Time Series and Forecasting*. Springer, 2nd edition, March 2002.

[58] Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener. Graph structure in the web. *Comput. Netw.*, 33(1-6):309–320, 2000.

[59] Coen Bron and Joep Kerbosch. Algorithm 457: finding all cliques of an undirected graph. *Commun. ACM*, 16(9):575–577, 1973.

[60] Sergey V. Buldyrev, Roni Parshani, Gerald Paul, H. Eugene Stanley, and Shlomo Havlin. Catastrophic cascade of failures in interdependent networks. *Nature*, 464(7291):1025–1028, April 2010.

[61] Deng Cai, Zheng Shao, Xiaofei He, Xifeng Yan, and Jiawei Han. Community mining from multi-relational networks. In *Proceedings of the 2005 European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'05)*, Porto, Portugal, 2005.

[62] J. Carroll and Jih J. Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of 'Eckart-Young' decomposition. *Psychometrika*, 35(3):283–319, September 1970.

[63] D. Cartwright and F. Harary. Structural balance: a generalization of Heider's theory. *Psychological Review*, 63(5):277–93, 1956.

[64] Deepayan Chakrabarti. Autopart: parameter-free graph partitioning and outlier detection. In *PKDD '04: Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 112–124, New York, NY, USA, 2004. Springer-Verlag New York, Inc.

[65] Deepayan Chakrabarti, Yang Wang 0008, Chenxi Wang, Jure Leskovec, and Christos Faloutsos. Epidemic thresholds in real networks. *ACM Trans. Inf. Syst. Secur.*, 10(4), 2008.

[66] Deepayan Chakrabarti and Christos Faloutsos. Graph mining: Laws, generators, and algorithms. *ACM Comput. Surv.*, 38(1):2, 2006.

[67] Deepayan Chakrabarti, Ravi Kumar, and Andrew Tomkins. Evolutionary clustering. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 554–560, New York, NY, USA, 2006. ACM.

[68] Deepayan Chakrabarti, Spiros Papadimitriou, Dharmendra S. Modha, and Christos Faloutsos. Fully automatic cross-associations. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 79–88, New York, NY, USA, 2004. ACM.

[69] Donald D. Chamberlin and Raymond F. Boyce. Sequel: A structured english query language. In Randall Rustin, editor, *SIGMOD Workshop, Vol. 1*, pages 249–264. ACM, 1974.

[70] Wei Chen, Yajun Wang, and Siyu Yang. Efficient influence maximization in social networks. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 199–208, New York, NY, USA, 2009. ACM.

[71] Yun Chi, Shenghuo Zhu, Yihong Gong, and Yi Zhang. Probabilistic polyadic factorization and its application to personalized recommendation. In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, pages 941–950, New York, NY, USA, 2008. ACM.

[72] Hyuk Cho, Inderjit S. Dhillon, Yuqiang Guan, and Suvrit Sra. Minimum sum-squared residue co-clustering of gene expression data. In *SDM*, 2004.

[73] Aaron Clauset, Cristopher Moore, and M. E. J. Newman. Structural inference of hierarchies in networks. *CoRR*, abs/physics/0610051, 2006.

[74] Aaron Clauset, Cristopher Moore, and M EJ Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453(7191):98–101, 2008.

[75] Aaron Clauset, M. E. J. Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical Review E*, 70:066111, 2004.

[76] Alain Colmerauer and Philippe Roussel. The birth of Prolog. In *The second ACM SIGPLAN conference on History of programming languages*, pages 37–52. ACM Press, 1993.

[77] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. The MIT Press, September 2001.

[78] 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.

[79] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, August 1991.

[80] Regino Criado, Julio Flores, Alejandro Garca del Amo, Jess Gmez-Gardees, and Miguel Romance. A mathematical model for networks with structures in the mesoscale. *CoRR*, abs/1012.3252, 2010. informal publication.

[81] Regino Criado, Miguel Romance, and M. Vela-Prez. Hyperstructures, a new approach to complex systems. *I. J. Bifurcation and Chaos*, 20(3):877–883, 2010.

[82] Leon Danon, Albert D. Guilera, Jordi Duch, and Alex Arenas. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(9):P09008–09008, September 2005.

[83] A. Decelle, F. Krzakala, C. Moore, and L. Zdeborová. Phase transition in the detection of modules in sparse networks. *ArXiv e-prints*, February 2011.

[84] Inderjit S. Dhillon, Subramanyam Mallela, and Dharmendra S. Modha. Information-theoretic co-clustering. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 89–98, New York, NY, USA, 2003. ACM.

[85] Jordi Duch and Alex Arenas. Community detection in complex networks using extremal optimization. *Physical Review E*, 72(2):027104+, Aug 2005.

[86] Tina Eliassi-Rad, Keith Henderson, Sprios Papadimitriou, and Christos Faloutsos. A hybrid community discovery framework for complex networks. In *SIAM Conference on Data Mining*, 2010.

[87] P. Erdös and A. Rényi. On random graphs, i. *Publicationes Mathematicae (Debrecen)*, 6:290–297, 1959.

[88] Elena A. Erosheva and Stephen E. Fienberg. Bayesian mixed membership models for soft clustering and classification. In *Classification - The ubiquitous challenge*. Springer Berlin Heidelberg, 2005.

[89] T. S. Evans and R. Lambiotte. Line graphs, link partitions and overlapping communities. *Physical Review E*, 80:016105, 2009.

[90] Jure Ferlez, Christos Faloutsos, Jure Leskovec, Dunja Mladenic, and Marko Grobelnik. Monitoring network evolution using mdl. *Data Engineering, International Conference on*, 0:1328–1330, 2008.

[91] Douglas H. Fisher. Knowledge acquisition via incremental conceptual clustering. *Mach. Learn.*, 2(2):139–172, September 1987.

[92] Gary William Flake, Steve Lawrence, C. Lee Giles, and Frans M. Coetzee. Self-organization and identification of web communities. *IEEE Computer*, 35:66–71, 2002.

[93] S. Fortunato and M. Barthelemy. Resolution limit in community detection. *Proceedings of the National Academy of Science*, 104:36–41, January 2007.

[94] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75 – 174, 2010.

[95] Santo Fortunato and Claudio Castellano. Community structure in graphs. In *Encyclopedia of Complexity and Systems Science*, pages 1141–1163. 2009.

[96] Santo Fortunato, Vito Latora, and Massimo Marchiori. Method to find community structures based on information centrality. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 70(5 Pt 2), November 2004.

[97] Francois Fouss, Alain Pirotte, and Marco Saerens. A novel way of computing similarities between nodes of a graph, with application to collaborative recommendation. In *WI '05: Proceedings of the The 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05)*, pages 550–556, Washington, DC, USA, 2005. IEEE Computer Society.

[98] Ove Frank and David Strauss. Markov graphs. *Journal of the American Statistical Association*, 81(395):832–842, 1986.

[99] D. W. Franks, J. Noble, P. Kaufmann, and S. Stagl. Extremism propagation in social networks with hubs. *Adaptive Behavior - Animals, Animats, Software Agents, Robots, Adaptive Systems*, 16(4):264–274, 2008.

[100] Linton C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40(1):35–41, March 1977.

[101] Lisa Friedland and David Jensen. Finding tribes: identifying close-knit individuals from employment patterns. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 290–299, New York, NY, USA, 2007. ACM.

[102] Bin Gao, Tie-Yan Liu, Xin Zheng, Qian-Sheng Cheng, and Wei-Ying Ma. Consistent bipartite graph co-partitioning for star-structured high-order heterogeneous data co-clustering. In *KDD '05: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 41–50, New York, NY, USA, 2005. ACM.

[103] Lise Getoor and Christopher P. Diehl. Link mining: a survey. *SIGKDD Explor. Newsl.*, 7(2):3–12, 2005.

[104] Gourab Ghoshal, Vinko Zlatić, Guido Caldarelli, and M. Newman. Random hypergraphs and their applications. *Physical Review E*, 79(6):066118+, June 2009.

[105] Fosca Giannotti, Mirco Nanni, and Dino Pedreschi. Efficient mining of temporally annotated sequences. In Joydeep Ghosh, Diane Lambert, David B. Skillicorn, and Jaideep Srivastava, editors, *SDM*. SIAM, 2006.

[106] Michelle Girvan and M. E. J. Newman. Community structure in social and biological networks. *PROC.NATL.ACAD.SCI.USA*, 99:7821, 2002.

[107] K. I. Goh, E. OH, H. Jeong, B. Kahng, and D. Kim. Classification of scale-free networks. *PROC.NATL.ACAD.SCI.USA*, 99:12583, 2002.

[108] Mark Goldberg, Stephen Kelley, Malik Magdon-Ismail, Konstantin Mertsalov, and William A. Wallace. Communication dynamics of blog networks. In *The 2nd SNA-KDD Workshop '08 (SNA-KDD'08)*, August 2008.

[109] G. H. Golub and C. F. Van Loan. Matrix computations. Baltimore, MD, USA, 1989.

[110] S. Gomez, P. Jensen, and A. Arenas. Analysis of community structure in networks of correlated data. *ArXiv e-prints*, December 2008.

[111] Manuel Gomez-Rodriguez, Jure Leskovec, and Andreas Krause. Inferring networks of diffusion and influence. In *KDD*, pages 1019–1028, 2010.

[112] Benjamin H. Good, Yves A. de Montjoye, and Aaron Clauset. Performance of modularity maximization in practical contexts. *Physical Review E*, 81(4):046106+, April 2010.

[113] Amit Goyal, Francesco Bonchi, and Laks V.S. Lakshmanan. Discovering leaders from community actions. In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, pages 499–508, New York, NY, USA, 2008. ACM.

[114] Amit Goyal, Byung-Won On, Francesco Bonchi, and Laks V. S. Lakshmanan. Gurumine: A pattern mining system for discovering leaders and tribes. *Data Engineering, International Conference on*, 0:1471–1474, 2009.

[115] Steve Gregory. An algorithm to find overlapping community structure in networks. In *Proceedings of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2007)*, pages 91–102. Springer-Verlag, September 2007.

[116] Steve Gregory. A fast algorithm to find overlapping communities in networks. In *ECML PKDD '08: Proceedings of the 2008 European Conference on Machine Learning and Knowledge Discovery in Databases - Part I*, pages 408–423, Berlin, Heidelberg, 2008. Springer-Verlag.

[117] Steve Gregory. Finding overlapping communities using disjoint community detection algorithms. In *Complex Networks: CompleNet 2009*, pages 47–61. Springer-Verlag, May 2009.

[118] Steve Gregory. Finding overlapping communities in networks by label propagation. *New Journal of Physics*, 12(10):103018, 2010.

[119] Tom Griffiths. Gibbs sampling in the generative model of latent dirichlet allocation. Technical report, Stanford University, 2002.

[120] Peter Grünwald. A tutorial introduction to the minimum description length principle. *CoRR*, math.ST/0406077, 2004.

[121] Peter D. Grünwald. *The Minimum Description Length Principle*, volume 1 of *MIT Press Books*. The MIT Press, 2007.

[122] Roger Guimera and Luis A. Amaral. Cartography of complex networks: modules and universal roles. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(02), February 2005.

[123] Roger Guimera and Luis A. Nunes Amaral. Functional cartography of complex metabolic networks. *Nature*, 433(7028):895–900, 2005.

[124] R. A. Hanneman and M. Riddle. Introduction to social network methods. 2005.

[125] R. A. Harshman. Foundations of the PARAFAC procedure: Models and conditions for an" explanatory" multi-modal factor analysis. *UCLA Working Papers in Phonetics*, 16(1):84, 1970.

[126] T. Hastie, R. Tibshirani, and J. H. Friedman. The elements of statistical learning. Springer, 2003.

[127] M. B. Hastings. Community detection as an inference problem. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 74(3):035102, 2006.

[128] Ricardo Hausmann, Cesar Hidalgo, Sebastian Bustos, Michele Coscia, Sara Chung, Juan Jimenez, Alexander Simoes, and Muhammed Yildirim. *The Atlas of Economic Complexity*. Harvard / MIT, 2011.

[129] George T. Heineman, Gary Pollice, and Stanley Selkow. Algorithms in a nutshell (chapter 6). In *O'Reilly Media*, 2008.

[130] Keith Henderson and Tina Eliassi-Rad. Applying latent dirichlet allocation to group discovery in large graphs. In *SAC '09: Proceedings of the 2009 ACM symposium on Applied Computing*, pages 1456–1461, New York, NY, USA, 2009. ACM.

[131] C. A. Hidalgo, B. Klinger, A.-L. Barabasi, and R. Hausmann. The Product Space Conditions the Development of Nations. *Science*, 317(5837):482–487, 2007.

[132] Jochen Hipp, Ulrich Güntzer, and Gholamreza Nakhaeizadeh. Algorithms for association rule mining – a general survey and comparison. *SIGKDD Explor. Newsl.*, 2(1):58–64, June 2000.

[133] Jake M. Hofman and Chris H. Wiggins. A bayesian approach to network modularity. *Physical Review Letters*, 100:258701, 2008.

[134] P. W. Holland, K. B. Laskey, and S. Leinhardt. Stochastic blockmodels: Some first steps. *Social Networks*, 5:109–137, 1983.

[135] Hiro Ito and Kazuo Iwama. Enumeration of isolated cliques and pseudo-cliques. In *ACM Transactions on Algorithms*, 2008.

[136] Hiro Ito, Kazuo Iwama, and Tsuyoshi Osumi. Linear-time enumeration of isolated cliques. In *ESA*, pages 119–130, 2005.

[137] Gabor Ivancsy, Renata Ivancsy, and Istvan Vajk. Graph mining-based image indexing. In *Proc. of the 5th International Symposium of Hungarian Researchers on Computational Intelligence*, pages 313–323, Budapest, Hungary, 2004.

[138] G Jeh and J Widom. Simrank: a measure of structural-context similarity. In *in KDD '02: Proceedings of the eighth ACM SIGKDD international*. ACM Press, 2002.

[139] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A. L. Barabasi. The large-scale organization of metabolic networks. *Nature*, 407(6804):651–654, 2000.

[140] Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul. An introduction to variational methods for graphical models. *Mach. Learn.*, 37(2):183–233, November 1999.

[141] Dmitri V. Kalashnikov and Sharad Mehrotra. Domain-independent data cleaning via analysis of entity-relationship graph. *ACM Trans. Database Syst.*, 31(2):716–767, June 2006.

[142] Leo Katz. A new status index derived from sociometric analysis. *Psychometrika*, (18):39–43, 1953.

[143] L. Kaufman and P. J. Rousseeuw. Finding groups in data: An introduction to cluster analysis. John Wiley, 1990.

[144] Charles Kemp, Joshua B. Tenenbaum, Thomas L. Griffiths, Takeshi Yamada, and Naonori Ueda. Learning systems of concepts with an infinite relational model. In *AAAI'06: Proceedings of the 21st national conference on Artificial intelligence*, pages 381–388. AAAI Press, 2006.

[145] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM, 2003.

[146] B. W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *The Bell system technical journal*, 49(1):291–307, 1970.

[147] J.R. Kettenring. Canonical analysis of several sets of variables. *Biometrika*, 58(3):433–451, 1971.

[148] Min-Soo Kim and Jiawei Han. A particle-and-density based evolutionary clustering method for dynamic networks. *Proc. VLDB Endow.*, 2(1):622–633, 2009.

[149] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science, Number 4598*, 220, 4598:671–680, 1983.

[150] J. Kleinberg. An impossibility theorem for clustering. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, pages 446–453. MIT Press, 2002.

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

[152] Hisashi Koga, Tetsuo Ishibashi, and Toshinori Watanabe. Fast agglomerative hierarchical clustering algorithm using locality-sensitive hashing. *Knowl. Inf. Syst.*, 12(1):25–53, 2007.

[153] Tamara G. Kolda and Brett W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, September 2009.

[154] Tamara G. Kolda and Jimeng Sun. Scalable tensor decompositions for multi-aspect data mining. In *ICDM 2008: Proceedings of the 8th IEEE International Conference on Data Mining*, pages 363–372, December 2008.

[155] Christian Komusiewicz, Falk Hüffner, Hannes Moser, and Rolf Niedermeier. Isolation concepts for efficiently enumerating dense subgraphs. *Theor. Comput. Sci.*, 410(38-40):3640–3654, 2009.

[156] Joseph B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 7(1):48–50, February 1956.

[157] Michihiro Kuramochi and George Karypis. Finding frequent patterns in a large sparse graph*. *Data Min. Knowl. Discov.*, 11(3):243–271, 2005.

[158] Maciej Kurant and Patrick Thiran. Layered complex networks. *Physical Review Letters*, 96(13):138701+, April 2006.

[159] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[160] S. Lall. The technological structure and performance of developing country manufactured exports, 1985-1998. *Queen Elizabeth House Working Paper #44*, 2000.

[161] R. Lambiotte, J. -. Delvenne, and M. Barahona. Laplacian Dynamics and Multiscale Modular Structure in Networks. *ArXiv e-prints*, December 2008.

[162] A. Lancichinetti and S. Fortunato. Community detection algorithms: A comparative analysis. *Physical Review E*, 80(5):056117–+, November 2009.

[163] Andrea Lancichinetti, Santo Fortunato, and Janos Kertesz. Detecting the overlapping and hierarchical community structure of complex networks. *New Journal of Physics*, 11:033015, 2009.

[164] Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. Benchmark graphs for testing community detection algorithms. *Physical Review E*, 78:046110, 2008.

[165] Amy Nicole Langville and William J. Stewart. Testing the nearest kronecker product preconditioner on markov chains and stochastic automata networks. *INFORMS Journal on Computing*, 16(3):300–315, 2004.

[166] Lieven De Lathauwer and Alexandre de Baynast. Blind deconvolution of ds-cdma signals by means of decomposition in rank-(1, l, l) terms. *IEEE Transactions on Signal Processing*, 56(4):1562–1571, 2008.

[167] E. E. Leamer. Sources of comparative advantage: Theory and evidence. *The MIT Press*, 1984.

[168] Sune Lehmann, Martin Schwartz, and Lars Kai Hansen. Bi-clique communities. *PHYS.REV.*, 78:016108, 2008.

[169] E. A. Leicht and M. E. J. Newman. Community structure in directed networks. *Physical Review Letters*, 100:118703, 2008.

[170] J. Leskovec, D. Huttenlocher, and J. Kleinberg. Predicting positive and negative links in online social networks. In *WWW*, pages 641–650. ACM, 2010.

[171] Jure Leskovec, Lada A. Adamic, and Bernardo A. Huberman. The dynamics of viral marketing. *ACM Transactions on the Web*, 1:1, 2007.

[172] Jure Leskovec, Lars Backstrom, Ravi Kumar, and Andrew Tomkins. Microscopic evolution of social networks. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 462–470. ACM, 2008.

[173] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Trans. Knowl. Discov. Data*, 1(1):2, 2007.

[174] Jure Leskovec, Jon M. Kleinberg, and Christos Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *KDD*, pages 177–187, 2005.

[175] Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance. Cost-effective outbreak detection in networks. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 420–429. ACM, 2007.

[176] Jure Leskovec, Kevin J. Lang, and Michael Mahoney. Empirical comparison of algorithms for network community detection. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 631–640, New York, NY, USA, 2010. ACM.

[177] Yu-Ru Lin, Yun Chi, Shenghuo Zhu, Hari Sundaram, and Belle L. Tseng. Facetnet: a framework for analyzing communities and their evolutions in dynamic networks. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 685–694, New York, NY, USA, 2008. ACM.

[178] Yu-Ru Lin, Jimeng Sun, Paul Castro, Ravi Konuru, Hari Sundaram, and Aisling Kelliher. Metafac: community discovery via relational hypergraph factorization. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 527–536, New York, NY, USA, 2009. ACM.

[179] Bo Long, Xiaoyun Wu, Zhongfei (Mark) Zhang, and Philip S. Yu. Unsupervised learning on k-partite graphs. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 317–326, New York, NY, USA, 2006. ACM.

[180] Bo Long, Zhongfei (Mark) Zhang, Xiaoyun Wú, and Philip S. Yu. Spectral clustering for multi-type relational data. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 585–592, New York, NY, USA, 2006. ACM.

[181] Qing Lu and Lise Getoor. Link-based classification. In Tom Fawcett, Nina Mishra, Tom Fawcett, and Nina Mishra, editors, *ICML*, pages 496–503. AAAI Press, 2003.

[182] Hao Ma, Haixuan Yang, Michael R. Lyu, and Irwin King. Mining social networks using heat diffusion processes for marketing candidates selection. In *CIKM*, pages 233–242, 2008.

[183] Sara C. Madeira and Arlindo L. Oliveira. Biclustering algorithms for biological data analysis: A survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1(1):24–45, 2004.

[184] Arun S. Maiya and Tanya Y. Berger-Wolf. Online sampling of high centrality individuals in social networks. In Mohammed Javeed Zaki, Jeffrey Xu Yu, B. Ravindran, and Vikram Pudi, editors, *PAKDD (1)*, volume 6118 of *Lecture Notes in Computer Science*, pages 91–98. Springer, 2010.

[185] Miller Mcpherson, Lynn S. Lovin, and James M. Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27(1):415–444, 2001.

[186] Stanley Milgram. The small world problem. *Psychology Today*, 2:60–67, 1967.

[187] Thomas Minka and John Lafferty. Expectation-propagation for the generative aspect model. In *In Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence*, pages 352–359. Morgan Kaufmann, 2002.

[188] Michael Mitzenmacher. A brief history of lognormal and power law distributions. *Internet Mathematics*, (1):2004.

[189] Michael Molloy and Bruce A. Reed. A critical point for random graphs with a given degree sequence. *Random Struct. Algorithms*, 6(2/3):161–180, 1995.

[190] J Moody. Race, school integration, and friendship segregation in america. *Am J Soc*, 107(3):679–716, 2001.

[191] M. Mørup and L. K. Hansen. Learning latent structure in complex networks. *Workshop on Analyzing Networks and Learning with Graphs*, 2009.

[192] P. J. Mucha, T. Richardson, K. Macon, M. A. Porter, and J.-P. Onnela. Community Structure in Time-Dependent, Multiscale, and Multiplex Networks. *Science*, 328:876–878, November 2010.

[193] M. N. L. Narasimhan. Principles of continuum mechanics. John Wiley and Sons, New York, 1993.

[194] Arvind Narayanan and Vitaly Shmatikov. How to break anonymity of the netflix prize dataset. *CoRR*, abs/cs/0610105, 2006.

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

[196] M. E. J. Newman. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69:066133, 2004.

[197] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74:036104, 2006.

[198] M. E. J. Newman. Modularity and community structure in networks. *PROC. NATL. ACAD. SCI. USA*, 103:8577, 2006.

[199] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69:026113, 2004.

[200] M. E. J. Newman and E. A. Leicht. Mixture models and exploratory analysis in networks. *Proceedings of the National Academy of Science*, 104:9564–9569, June 2007.

[201] M.E.J. Newman. Detecting community structure in networks. *Eur. Phys. J. B*, 38(2):321–330, mar 2004.

[202] Phu Chien Nguyen, Takashi Washio, Kouzou Ohara, and Hiroshi Motoda. Using a hash-based method for apriori-based graph mining. In *PKDD '04: Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 349–361, New York, NY, USA, 2004. Springer-Verlag New York, Inc.

[203] V. Nicosia, G. Mangioni, V. Carchiolo, and M. Malgeri. Extending the definition of modularity to directed graphs with overlapping communities. *J.STAT.MECH.*, page P03024, 2009.

[204] Siegfried Nijssen and Joost N. Kok. A quickstart in frequent structure mining can make a difference. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 647–652, New York, NY, USA, 2004. ACM.

[205] Naomi Nishimura, Prabhakar Ragde, and Dimitrios M. Thilikos. Fast fixed-parameter tractable algorithms for nontrivial generalizations of vertex cover. *Discrete Appl. Math.*, 152(1-3):229–245, 2005.

[206] Jasmine Novak, Prabhakar Raghavan, and Andrew Tomkins. Anti-aliasing on the web. In *WWW*, pages 30–39, 2004.

[207] David L. Nowell and Jon Kleinberg. The link-prediction problem for social networks. *Journal of the American Society for Information Science and Technology*, 58(7):1019–1031, 2007.

[208] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, 1998.

[209] Gergely Palla, Imre Derenyi, Illes Farkas, and Tamas Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435:814, 2005.

[210] Spiros Papadimitriou, Aristides Gionis, Panayiotis Tsaparas, Risto A. Visnen, Heikki Mannila, and Christos Faloutsos. Parameter-free spatial data mining using mdl. *Data Mining, IEEE International Conference on*, 0:346–353, 2005.

[211] Spiros Papadimitriou, Jimeng Sun, Christos Faloutsos, and Philip S. Yu. Hierarchical, parameter-free community discovery. In *ECML PKDD '08: Proceedings of the European conference on Machine Learning and Knowledge Discovery in Databases - Part II*, pages 170–187, Berlin, Heidelberg, 2008. Springer-Verlag.

[212] Roni Parshani, Sergey V. Buldyrev, and Shlomo Havlin. Interdependent networks: Reducing the coupling strength leads to a change from a first to second order percolation transition. *Physical Review Letters*, 105(4):048701+, July 2010.

[213] Greg Pass, Abdur Chowdhury, and Cayley Torgeson. A picture of search. In Xiaohua Jia, editor, *Infoscale*, volume 152 of *ACM International Conference Proceeding Series*, page 1. ACM, 2006.

[214] J. Pitman. *Combinatorial Stochastic Processes.* Lecture Notes in Mathematics, Vol. 1875, Picard, Jean (Ed.), 2006.

[215] Pascal Pons and Matthieu Latapy. Computing communities in large networks using random walks. In *ISCIS 2005*, volume 3733, chapter 31, pages 284–293. Springer, Berlin, Heidelberg, 2005.

[216] Mason A. Porter, Jukka-Pekka Onnela, and Peter J. Mucha. Communities in networks. *NOTICES OF THE AMERICAN MATHEMATICAL SOCIETY*, 56:1082, 2009.

[217] Alex Pothen, Horst D. Simon, and Kan-Pu Liou. Partitioning sparse matrices with eigenvectors of graphs. *SIAM J. Matrix Anal. Appl.*, 11(3):430–452, 1990.

[218] D. J. de S. Price. Networks of scientific papers. *Science*, 149(3683):510–515, July 1965.

[219] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi. Defining and identifying communities in networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101(9):2658–2663, March 2004.

[220] Usha Nandini Raghavan, Reka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76:036106, 2007.

[221] W. J. Reed. The pareto, zipf and other power laws. *Economics Letters*, 74(1):15–19, 2001.

[222] J. Reichardt and S. Bornholdt. Statistical mechanics of community detection. 74(1):016110–+, July 2006.

[223] J Rissanen. Modelling by the shortest data description. *Automatica*, 14:465–471, 1978.

[224] Stephen E. Robertson, C. J. van Rijsbergen, and Martin F. Porter. Probabilistic models of indexing and searching. In *SIGIR*, pages 35–56, 1980.

[225] Thomas Roelleke and Jun Wang. TF-IDF uncovered: a study of theories and probabilities. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 435–442, New York, NY, USA, 2008. ACM.

[226] Giulio Rossetti, Michele Berlingerio, and Fosca Giannotti. Scalable link prediction on multidimensional networks via structural analysis. In *ICDM Workshop*, 2011.

[227] M. Rosvall and C. T. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Science*, 105:1118–1123, January 2008.

[228] Kazumi Saito and Takeshi Yamada. Extracting communities from complex networks by the k-dense method. In *ICDMW '06: Proceedings of the Sixth IEEE International Conference on Data Mining - Workshops*, pages 300–304, Washington, DC, USA, 2006. IEEE Computer Society.

[229] Gerard Salton and Chris Buckley. Term weighting approaches in automatic text retrieval. Technical report, Ithaca, NY, USA, 1987.

[230] Satu Elisa Schaeffer. Stochastic local clustering for massive graphs. In *Proceedings of the Ninth Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD-05)*, volume 3518 of *Lecture Notes in Computer Science*, pages 354–360. Springer-Verlag GmbH, 2005.

[231] Satu Elisa Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27 – 64, 2007.

[232] M. Schich, C. Hidalgo, S. Lehmann, and J. Park. The network of subject co-popularity in classical archaeology. *Bolletino di Archeologia On-line*, 2009.

[233] Maximilian Schich and Michele Coscia. Exploring co-occurrence on a meso and global level using network analysis and rule mining. *Proceedings of the ninth workshop on mining and Learning with Graphs MLG 11*, 2011.

[234] Martina Schwarz et al. Archäologische bibliographie. online-database. *Munich: Biering & Brinkmann*, 1956-2011.

[235] John P. Scott. *Social Network Analysis: A Handbook*. SAGE Publications, 2000.

[236] M. Ángeles Serrano, Marián Boguñá, and Alessandro Vespignani. Extracting the multiscale backbone of complex weighted networks. *Proceedings of the National Academy of Sciences*, 106(16):6483–6488, 2009.

[237] Huawei Shen, Xueqi Cheng, Kai Cai, and Mao-Bin Hu. Detect overlapping and hierarchical community structure in networks. *PHYSICA A*, 388:1706, 2009.

[238] X. Shi, B. Tseng, and L.A. Adamic. Looking at the Blogosphere Topology through Different Lenses. In *ICWSM 2007*, volume 1001, page 48109, 2007.

[239] Ray Solomonoff and Anatol Rapoport. Connectivity of random nets. *Bulletin of Mathematical Biology*, 13(2):107–117, June 1951.

[240] Jimeng Sun, Christos Faloutsos, Spiros Papadimitriou, and Philip S. Yu. Graphscope: parameter-free mining of large time-evolving graphs. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 687–696, New York, NY, USA, 2007. ACM.

[241] Jimeng Sun, S. Papadimitriou, and P.S. Yu. Window-based tensor analysis on high-dimensional and multi-aspect streams. pages 1076 –1080, dec. 2006.

[242] Jimeng Sun, Dacheng Tao, and Christos Faloutsos. Beyond streams and graphs: dynamic tensor analysis. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 374–383, New York, NY, USA, 2006. ACM.

[243] Yizhou Sun, Rick Barber, Manish Gupta, Charu C. Aggarwal, and Jiawei Han. Co-author relationship prediction in heterogeneous bibliographic networks. In *ASONAM*, pages 121–128, 2011.

[244] Michael Szell, Renaud Lambiotte, and Stefan Thurner. Multirelational organization of large-scale social networks in an online world. *Proceedings of the National Academy of Sciences*, 107(31):13636–13641, August 2010.

[245] Lei Tang and Huan Liu. Relational learning via latent social dimensions. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 817–826, New York, NY, USA, 2009. ACM.

[246] Lei Tang and Huan Liu. Scalable learning of collective behavior based on sparse social dimensions. In *CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management*, pages 1107–1116, New York, NY, USA, 2009. ACM.

[247] Lei Tang and Huan Liu. Uncovering cross-dimension group structures in multi-dimensional networks. In *SDM workshop on Analysis of Dynamic Networks*, 2009.

[248] Lei Tang, Suju Rajan, and Vijay K. Narayanan. Large scale multi-label classification via metalabeler. In *WWW '09: Proceedings of the 18th international conference on World wide web*, pages 211–220, New York, NY, USA, 2009. ACM.

[249] Lei Tang, Xufei Wang, and Huan Liu. Uncovering groups via heterogeneous interaction analysis. In *ICDM*. IEEE, 2009.

[250] Chayant Tantipathananandh and Tanya Berger-Wolf. Constant-factor approximation algorithms for identifying dynamic communities. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 827–836, New York, NY, USA, 2009. ACM.

[251] Chayant Tantipathananandh, Tanya Berger-Wolf, and David Kempe. A framework for community identification in dynamic social networks. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 717–726, New York, NY, USA, 2007. ACM.

[252] Y. W. Teh, D. Newman, and M. Welling. A collapsed variational Bayesian inference algorithm for latent Dirichlet allocation. In *Advances in Neural Information Processing Systems*, volume 19, 2007.

[253] V. A. Traag and J. Bruggeman. Community detection in networks with positive and negative links. *arXiv*, 80(3):036115–+, September 2009.

[254] Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. Support vector machine learning for interdependent and structured output spaces. In *ICML*, 2004.

[255] Takeaki Uno, Masashi Kiyomi, and Hiroki Arimura. Lcm ver.3: collaboration of array, bitmap and prefix tree for frequent itemset mining. In *OSDM '05: Proceedings of the 1st international workshop on open source data mining*, pages 77–86, New York, NY, USA, 2005. ACM.

[256] S. M. van Dongen. *Graph Clustering by Flow Simulation*. PhD thesis, University of Utrecht, The Netherlands, 2000.

[257] Alexei Vazquez, Marian Boguna, Yamir Moreno, Romualdo Pastor-Satorras, and Alessandro Vespignani. Topology and correlations in structured scale-free networks. *Physical Review E*, 67:046111, 2003.

[258] I. Vragović and E. Louis. Network community structure and loop coefficient method. *Phys. Rev. E*, 74(1):016105, Jul 2006.

[259] Ken Wakita and Toshiyuki Tsurumi. Finding community structure in mega-scale social networks: [extended abstract]. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 1275–1276, New York, NY, USA, 2007. ACM.

[260] Matthew L. Wallace, Yves Gingras, and Russell Duhon. A new approach for detecting scientific specialties from raw cocitation networks. *J. Am. Soc. Inf. Sci. Technol.*, 60(2):240–246, 2009.

[261] Y. J. Wang and G. Y. Wong. Stochastic blockmodels for directed graphs. *Journal of American Statistical Associ- ation*, 82:8–19, 1987.

[262] Duncan J. Watts and Steve H. Strogatz. Collective dynamics of small-world networks. *Nature*, 393(6684):440–442, June 1998.

[263] Fang Wei, Weining Qian, Chen Wang, and Aoying Zhou. Detecting overlapping community structures in networks. *World Wide Web*, 12(2):235–261, 2009.

[264] Fang Wei, Chen Wang, Li Ma, and Aoying Zhou. Detecting overlapping community structures in networks with global partition and local expansion. *Progress in WWW Research and Development*, 2008.

[265] Douglas R White, Frank Harary, Michael Sobel, and Mark Becker. The cohesiveness of blocks in social networks: Node connectivity and conditional density. *Sociological Methodology*, 2001.

[266] Howard D. White, Barry Wellman, and Nancy Nazer. Does citation reflect social structure?: longitudinal evidence from the "globenet" interdisciplinary research group. *J. Am. Soc. Inf. Sci. Technol.*, 55(2):111–126, 2004.

[267] F. Wu and B. A. Huberman. Finding communities in linear time: a physics approach. *The European Physical Journal B - Condensed Matter and Complex Systems*, 38(2):331–338, 2004.

[268] Xifeng Yan and Jiawei Han. gspan: Graph-based substructure pattern mining. ICDM '02, pages 721–. IEEE, 2002.

[269] Ding Zhou, Sergey A. Orshanskiy, Hongyuan Zha, and C. Lee Giles. Co-ranking authors and documents in a heterogeneous network. In *ICDM*, pages 739–744. IEEE Computer Society, 2007.

[270] Haijun Zhou and Reinhard Lipowsky. Network brownian motion: A new method to measure vertex-vertex proximity and to identify communities and subcommunities. In *International Conference on Computational Science*, pages 1062–1069, 2004.

[271] Etay Ziv, Manuel Middendorf, and Chris H. Wiggins. Information-theoretic approach to network modularity. *Phys. Rev. E*, 71(4):046117, Apr 2005.

[272] Vinko Zlatic, Gourab Ghoshal, and Guido Caldarelli. Hypergraph topological quantities for tagged social networks. *CoRR*, abs/0905.0976, 2009.