Link Prediction in Large-scale Complex Networks (Application to bibliographical Networks)

Manisha Pujari

N° attribué par la bibliothèque

| _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ |

UNIVERSITÉ PARIS NORD

DOCTORAL THESIS

Link Prediction in Large-scale Complex Networks

(Application to Bibliographical Networks)

Author:

Manisha PUJARI

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science in the research team

Apprentissage Artificiel et Applications

LIPN CNRS UMR-7030

Jury:

Reviewer:		
Céline ROBARDET	Professor	INSA Lyon
Bénédicte LE GRAND	Professor	Université Paris 1 Panthéon Sorbonne
Examiner:		
Aldo GANGEMI	Professor	SPC, Université Paris 13
Christophe PRIEUR	Associate Professor, HDR	Université Paris Diderot
Director:		
Céline ROUVEIROL	Professor	SPC, Université Paris 13
Supervisor:		
Rushed KANAWATI	Associate Professor	SPC, Université Paris 13

“ The more I learn, the more I realize how much I don’t know. ”

Albert Einstein

Abstract

Link Prediction in Large-scale Complex Networks
(Application to Bibliographical Networks)

In this work, we are interested to tackle the problem of link prediction in complex networks. In particular, we explore topological dyadic approaches for link prediction. Different topological proximity measures have been studied in the scientific literature for finding the probability of appearance of new links in a complex network. Supervised learning methods have also been used to combine the predictions made or information provided by different topological measures. They create predictive models using various topological measures. The problem of supervised learning for link prediction is a difficult problem especially due to the presence of heavy class imbalance.

In this thesis, we search different alternative approaches to improve the performance of different dyadic approaches for link prediction. We propose here, a new approach of link prediction based on supervised rank aggregation that uses concepts from computational social choice theory. Our approach is founded on supervised techniques of aggregating sorted lists (or preference aggregation). We also explore different ways of improving supervised link prediction approaches. One approach is to extend the set of attributes describing an example (pair of nodes) by attributes calculated in a multiplex network that includes the target network. Multiplex networks have a layered structure, each layer having different kinds of links between same sets of nodes. The second way is to use community information for sampling of examples to deal with the problem of class imbalance. Experiments conducted on real networks extracted from well known DBLP bibliographic database.

Keywords: Complex networks, Link prediction, Supervised rank aggregation, Multiplex network analysis.

Résumé

Prévision de Liens dans les Grands Graphes de Terrain
(Application aux réseaux bibliographiques)

Nous nous intéressons dans ce travail au problème de prévision de nouveaux liens dans des grands graphes de terrain. Nous explorons en particulier les approches topologiques dyadiques pour la prévision de liens. Différentes mesures de proximité topologique ont été étudiées dans la littérature pour prédire l'apparition de nouveaux liens. Des techniques d'apprentissage supervisé ont été aussi utilisées afin de combiner ces différentes mesures pour construire des modèles prédictifs. Le problème d'apprentissage supervisé est ici un problème difficile à cause notamment du fort déséquilibre de classes.

Dans cette thèse, nous explorons différentes approches alternatives pour améliorer les performances des approches dyadiques pour la prévision de liens. Nous proposons d'abord, une approche originale de combinaison des prévisions fondée sur des techniques d'agrégation supervisée de listes triées (ou agrégation de préférences). Nous explorons aussi différentes approches pour améliorer les performances des approches supervisées pour la prévision de liens. Une première approche consiste à étendre l'ensemble des attributs décrivant un exemple (paires de noeuds) par des attributs calculés dans un réseau multiplexe qui englobe le réseau cible. Un deuxième axe consiste à évaluer l'apport des techniques de détection de communautés pour l'échantillonnage des exemples. Des expérimentations menées sur des réseaux réels extraits de la base bibliographique DBLP montrent l'intérêt des approaches proposées.

Mots-clès : Réseaux complexes, Prévision de liens, Agrégation supervisée de préférences, Analyse de réseaux multiplexes.

Dedicated to my mother Dr. Kamalini Pujari

Acknowledgements

Acknowledgement is something very special for me where I can show my gratitude towards all who have directly or indirectly affected my life and helped me to sail through the tides.

I must start with my supervisor Dr. Rushed Kanawati who has been a perfect guide during this research work. He has been a great mentor for me starting from the days of my internship in 2010. I still remember the early time when I was not very confident about my ability to do a thesis, and it was Rushed who used to tell me stories from his life to build up my confidence. And indeed, I will be going out to the professional world as a much more confident person than before. His liveliness and enthusiasm towards his work is very inspiring. His advice on both research as well as on my career have been invaluable. I thank him for believing in me and giving me an opportunity to work with him.

I would like to express my thanks to my thesis director, Prof. Céline Rouveirol for being very kind and supportive throughout these years of research. In spite of her busy schedules, she always took time for guiding and helping me whenever I needed. She was always ready to answer my questions on anything regarding teaching or research. Her encouragement and guidance during the final preparations of this thesis and presentation is overwhelming. She is a highly knowledgeable and experienced person. I feel myself very fortunate to have got a chance know her and to work with her.

I also thank all members of the jury for accepting to be a part of final judgement day of my work. In particular, I would like to thank Prof. Céline Robardet and Prof. Bénédicte Le Grand for being the reviewers. Their comments and advices helped me a lot to improve this manuscript.

A good support system is important for surviving and staying sane during Ph.D. For making this journey fun and being there with me, many thanks to all my friends: Yue Ma, Zied Yakoubi, Abdoulaye Guissé, Nasserine Benchettara, Pegah Alizadeh, Hanene Ochi, Ines Chebil, Amine Chaibi, Sarra Ben Abbes, Nouha Omrane. They will always remain in my mind and heart. I also thanks all other friends and colleagues at LIPN. My list will remain incomplete without thanking Brigitte Guéveneux, Nathalie Tavares, Marie Fontanillas and Antonia Wilk who were very kind and helpful for all types of administrative works throughout these years.

In the end I would like to thank my entire family for being with me through thick and thin. My husband Samarendra Tripathy is my greatest strength. He is my best friend, my guide and my critic too. It was his dream that I do a Ph.D. and it was never possible

for me to complete this work without his constant support. I feel so lucky to have him in my life. I think of my parents today. I thank my mother Kamalini Pujari for being the greatest inspiration of my life. May it be work or home, she is just perfect. I thank her for teaching me how to lead a happy life, not just by words but by example. I thank my father Byomokesh Pujari for his unconditional love and trust in me at every step of life. He has been a friend in moments when I badly needed one. I also thank my little brother Manas Ranjan Pujari for being a wonderful sibling in spite of all tantrums I throw. I greatly thank my parent-in-laws, Binod Chandra Tripathy and Snehalata Tripathy for their constant blessings and prayers which helped me to never lose hope and keep going. I thank the entire family of my husband for their fun-filled and sweet encouraging words that helped me discharging all the stress. Thank you.

Manisha Pujari
March, 2015

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Context

Complex networks and their characteristics have gained considerable attention of researchers in various domains. A complex network can be any real world network which has an abstract form without a predefined structure or pattern of evolution. They can be highly dynamic in nature, evolving or changing constantly. Also, starting with a tiny form, in this era of big data, there is a spectacular increase in the size of the network. Analyzing these dynamic large-scale networks is a major challenge for network scientists.

Many real-world systems can be modeled as evolving networks of interacting *actors* (e.g. users, authors, documents or scientific papers, items, proteins etc.). They can be on-line social networks depicting social relationships between people like friendship; collaboration networks showing some kind of professional relationships (such as academic co-authorship/co-publishing networks, product co-purchasing etc.); biological systems (such as protein interaction networks) or computer science networks (e.g. the Internet and peer-to-peer networks) etc. These systems can be represented as graphs with *actors* as nodes and edges representing any kind of interaction, collaboration or influence between actors. Almost all types of complex networks have some common topological properties like sparseness or low density, small diameter or average distance, a degree distribution following power-law, high clustering coefficient, presence of community structures etc. The basic network structures and properties are explained in Chapter 2. The real-world networks can be heterogeneous in nature having different kinds of nodes and links. One such representation of heterogeneity is in the form of *multiplex networks*. These networks have a layered structure with same types of nodes but different types of links in each layer. This concept is discussed in detail in Chapter 5.

Bibliographical networks especially scientific collaboration networks are very rich in a variety of information that can be exploited for network analysis. Since a long time, they have attracted attention of many researchers. They can be used for multiple network analysis tasks like link prediction, community detection, identification of influential nodes etc. Also due to the presence of different kinds of link information, these networks have been used for the study of heterogeneous properties of complex networks.

Link prediction, which is the central topic in this research, refers to the task of predicting the existence or occurrence of associations (edges) in the network at a given point of time

t when provided with the information about the network's history before time t. The problem has a wide variety of applications such as: recommender systems, identification of probable professional or academic associations in scientific collaboration networks, identification of structures of criminal networks and structural analysis in the field of microbiology or biomedicine, etc. A variety of approaches has been proposed in the scientific literature. Recent surveys on the topic can be found in [Al Hasan and Zaki, 2010; Lü and Zhou, 2011]. A detailed state of art of link prediction approaches is provided in Chapter 3.

1.2 Contributions

The major contributions of this research work, that are presented in this report are:

1. **Supervised rank aggregation based link prediction:** Many link prediction approaches use topological characteristics of a pair of unconnected nodes for making prediction of appearance or existence of a probable link between them. We come up with a new approach where the effect of topological attributes is combined by using supervised rank aggregation (stemming from computational social choice theory) and this combined effect is used to predict the possible new links in a co-authorship network. The approach has been compared with the baseline supervised machine learning methods. Rank aggregation methods have been very much used in meta-search engines, but not much seemed to have been explored in the context of link prediction.

    - Manisha Pujari, Rushed Kanawati. *Link prediction in complex networks by supervised rank Aggregation.* ICTAI 2012: 24th IEEE International Conference on Tools with Artificial Intelligence, pages.782-789, 7-9 November, 2012, Athens, Greece.
    - Manisha Pujari, Rushed Kanawati. *Supervised rank aggregation approach for link prediction in complex networks.* In Proceedings of the 21st international conference companion on World Wide Web (WWW '12 Companion). ACM, New York, NY, USA, pages 1189-1196.

2. **Link prediction in multiplex networks:** While working on homogeneous networks our attention went to the widely existing heterogeneity in complex networks. Heterogeneous link information can be very well used to improve the prediction results. We were particularly interested to work on multiplex networks. Multiplex networks are a category of heterogeneous complex networks, which essentially have different kinds of links between same nodes. They can be represented as a set (or layers) of simple networks, each having the same set of nodes but different types of links (dimensions). Our approach includes computations of simple topological scores (attributes) for unconnected node pairs from different dimension graphs. These scores can then be used either directly or in a combined way for the purpose of link prediction. Combinations of scores can be done using any standard functions like min, max or average. We also propose an entropy based version of the score, which gives importance to the presence of a non-zero value in each dimension.

    - Manisha Pujari, Rushed Kanawati. *Link prediction in multiplex networks.* Special Issue of Networks and Heterogeneous Media entitled New trends, models and applications in Complex and Multiplex Networks, 2014.(Accepted)

- Manisha Pujari, Rushed Kanawati. *Link prediction in multiplex bibliographical networks.* International Journal of Complex Systems in Science proceedings of NET-WORKS 2013, El Escorial, 11-13 December, 2013.

3. **Community detection and link prediction:** Another research direction that we are interested in, is the use of communities in link prediction. Our goal is to study how the presence of an unconnected node pair in same or different communities can affect the probability of having a new link between them. We have used communities to filter out irrelevant candidate node pairs to build a better prediction model. Another use of community detection methods can be to produce compressed graphs that can help in dealing with the large sizes of real networks. We are on our way to explore it more and find how to use it for link prediction.

New topological measure: As an additional work in the course of our research, we have developed a new path based topological measure which can be used for unsupervised and supervised methods of link prediction. It tries to quantify the importance of a shortest path between two nodes in a network. We experimented to see its utility in the task of link prediction in a scientific collaboration network. It can also be used for other analysis tasks in complex networks but in this work we could only explore its applicability in the context of link prediction. Details about this measure can be found in appendix B.

- Manisha Pujari. *Path betweenness centrality: A new topological measure for link prediction.* Journée de fouille de grandes graphes (JFGG), MARAMI2013, 16-18 October 2013, à Saint-Etienne, France.

At the same time we have also done some work on the application of link prediction:

Link prediction based tag recommendation: We have also applied link prediction for the purpose of tag recommendation in folksonomy. Folksonomies are websites where users can save and share online resources (documents, bookmarks, images, songs etc.) with other users and they have complete freedom to choose tags to annotate their resources. Such systems are prone to the problem of *tag ambiguity*. We developed a framework called LiPTaR to cope up with the problem of tag recommendation. The original idea of LiPTaR is to mine the dynamic of the tagging activity in order to compute the most suitable tag for a given user and a given resource. The tagging history of each user is modelled by a temporal sequence of bipartite graphs linking tags to resources. Given a target user and a target resource, we first compute a set of similar users. The tagging history of the identified set of users is merged in one temporal sequence of bipartite graphs. The obtained sequence is used to learn a model of link prediction in bipartite graphs. The learned model is then applied to predict tags to be linked to the target resource and a list of most similar resources. (Appendix A)

- Manisha Pujari, Rushed Kanawati. *Tag recommendation by link prediction based on supervised machine learning.* Sixth International AAAI Conference on Weblogs and Social Media (ICWSM 2012), 4-7 June 2012, Dublin. (Poster session) (selection rate 26%)

- Manisha Pujari, Rushed Kanawati. *Supervised machine learning link prediction approach for tag recommandation.* 4th International Conference on Online Communities and Social Computing @ HCI International, pages.336-344, 9-14 July 2011, Hilton Orlando Bonnet Creek, Orlando, Florida, USA, LNCS Springer.

1.3 Outline

This report is organized as follows.

- Chapter 2 presents the context of our research work. It provides a detailed description about complex networks, their structures and properties. It also presents the different tasks involved in complex network analysis.
- Chapter 3, provides basic definition of the link prediction problem. We provide a quick survey on existing approaches, mostly based on network structure. It also includes details about a new type of path based topological feature that can be used for link prediction.
- Chapter 4 describes our proposed approach for link prediction based on supervised rank aggregation and gives a detail of experimentation done so far. We provide information about standard rank aggregation methods, our proposed approximate version of one of the well known methods *Kemeny optimal aggregation*, and its application to a supervised link prediction task.
- Chapter 5, presents expansion our sphere of research by including heterogeneity in the network in the form of *multiplex networks*. We present our versions of multiplex topological attributes, extending a few of know features to fit into the scenario of multiplex networks. We then apply them for predicting co-authorship links.
- Chapter 6 provides details about community detection and different existing algorithms. It also present our proposed method of filtering of potential candidates using community information. This is mainly done to reduce the large size of a dataset to a manageable level and deal with the problem of class imbalance which is very common in especially supervised link prediction approaches.

Last but not the least, the report is concluded in Chapter 7 where we summarize the whole work, ending with the perspectives.

Some additional information is provided in the appendices. Appendix A presents our work on application of link prediction for *tag recommendation* in folksonomies. Appendix B provides details about our work on *path betweenness centrality* . Appendix C provides information about the predictive performances of some of the topological metrics in co-authorship link prediction in authors networks created from DBLP database. Appendix D presents a list of our publications done during this research work. Lastly, appendix E provides visualizations of some other co-authorship networks and multiplex networks generated from DBLP data.

Chapter 2

Complex Networks Analysis

2.1 Introduction

Analysis of complex networks, traditionally known as *Social Network Analysis (SNA)*, has its theoretical roots in the work of early sociologists such as George Simmel and Émile Durkheim, who wrote about the importance of studying patterns of relationships that connect social actors. A complex network consists of several independent units interacting in a non-linear way. The brain is a network of nerve cells connected by axons; cells themselves are networks of molecules connected by biochemical reactions. Societies are networks of people, connected by friendship, familial and professional relationships. On large scale, food webs and ecosystems can be represented as networks of predator-prey relations. Networks are present also in technology: the Internet as web pages connected by hyperlinks, the routers network, power grids, and transportation systems are some of the examples. Graph theory emerged as a key tool for analyzing complex networks. Having its roots basically in sociology and mathematics, it gained rapid importance in network analysis in the field of biology, physics, telecommunication, computer science and others. It has various divisions like structural analysis of network, temporal analysis that studies the evolution of networks, content-based analysis etc.

In this chapter we throw light on some basic concepts related to representation and analysis of complex networks using graph theory. Section 2.2 gives a general description about complex networks, different types of graphs that can be used to represent complex networks and formal definition of a complex network. Section 2.3 summarizes the basic characteristics of networks which are common to almost all types of complex networks. Section 2.4 present different classical ways of modeling networks and their relevance in real network modeling. Section 2.5 lists different tasks in complex networks analysis describing a few of them. Section 2.6 presents specific description of Bibliographical networks focusing mainly on scientific collaboration networks.

2.2 Complex networks

In [Estrada, 2011], author defines a network as a diagrammatic representation of a system consisting of nodes and links or edges. Nodes represent the units/entities of the system. Nodes are joined by links or edges, which represent a particular type of interconnection

or relationship between those entities. We use the term network or graph to represent the graphical representation of a real complex network. Network or graphs are used in various disciplines and each have their own terms for entities and relationships. Table 2.1 lists some graph terminologies used in different domains of research. In this report, we shall use *nodes* to represent entities and *edges* or *links* to represent relationships or interactions in a graph.

There are a number of real world systems that can be represented in the form of graphs, the most known in computer science being the Internet and the World Wide Web. Then there are many biological networks like food webs, protein interactions, connections of nervous systems or blood vessels. Infrastructural systems such as networks of transportation connecting roads or places [Barthélemy, 2011; Roth et al., 2012] and also those of power-grids can also be studied using graphs. Most popular networks are that of social systems showing relationships such as friendship, trust, academic, professional or commercial collaboration. There are also graphs showing review or rating or opinion of people on movies, actors, on various purchased products etc. Figure 2.1 shows two examples of complex networks: one is protein interaction network where the nodes are proteins and the other one shows a graph from social interaction website Twitter[1] where the nodes represent people or users and links represents who follows who or who is followed by whom.

Discipline	**Units/Entities**	**Relationships/Interactions**
Physics	sites	bonds
Sociology	actors	ties, relations
Mathematics	vertices	edges, arcs
Computer science	nodes, vertices	edges, links
Biology	nodes, vertices	edges

Table 2.1: Network terminology

There are different ways of categorizing graphs based on their edge type, orientation, presence of weights, number of edges etc.

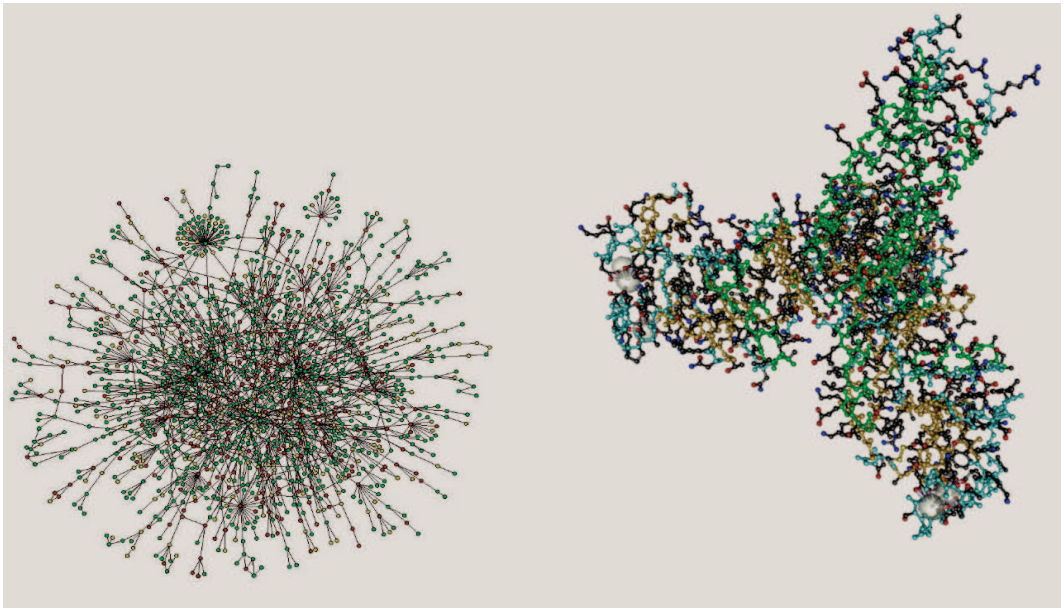
1. **Simple graphs** have only undirected edges without any weights. That means the edges do not have any orientation and are symmetric. They represent binary relationships between nodes.

2. **Directed graphs** or **digraphs** have directed edges between nodes. Each edge has an orientation.

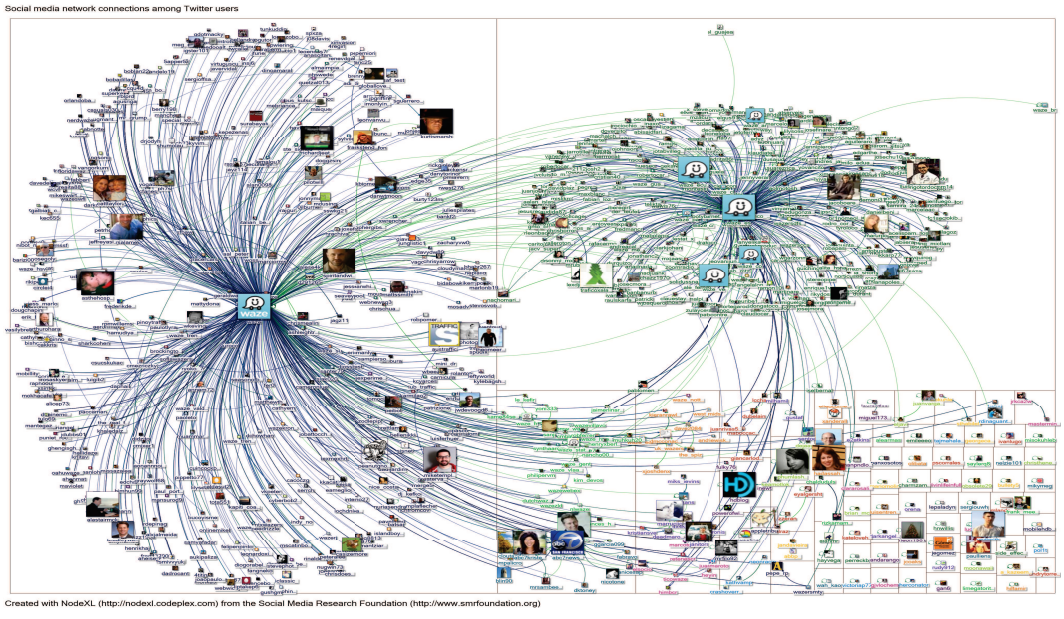
3. **Pseudographs** [Harary, 1969] are the graphs where we have the possibilities of having multiple links between nodes (multi-graphs). They can also contain self-loops (link from a node to itself). They have links that can be directed as well as un-directed. Multi-graphs with multiple edges having different labels or types, can be represented as layers of graphs, each having same nodes but only one type of edges. These are called *Multiplex* graphs which is more deeply explained in Chapter 5.

4. **Weighted graphs** [Barrat et al., 2004; Newman, 2004a] have weights assigned to edges. These weights are generally real numbers. Multi-graphs, in many occasions,

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



(a) The protein-protein interaction network of yeast showing a scale-free topology: a few proteins interact with a large number of other proteins, while most proteins have only one or two links. (source. `http://plaza.ufl.edu/rkirch05/cis6930/`) [Barabási and Oltvai, 2004]



(b) Twitter social interaction network constructed using NodeXL (source. `www.flickr.com/photos/marc_smith`)

FIGURE 2.1: Complex networks from different data sources

are transformed into weighted graphs in such a way that the number of edges connecting two nodes is reflected in the edge weight of the new graph.

5. **Hypergraphs** are graphs where an edge connects more than two nodes. The edges are referred to as *hyperlinks* or *hyperedges*. A popular example of hypergraphs can be a folksonomy. Folksonomies are networks having users, resources and tags as nodes. Users are people who participate in the network. Resources are any online resources like online documents, images music files, web links (urls) and so on. Tags are words or terms selected by users to annotate the resources they use on the network. An edge in the network is composed of a user node, a resource node and a tag node. It represents an association involving three entities. Figure 2.2 shows a small example of such a network.

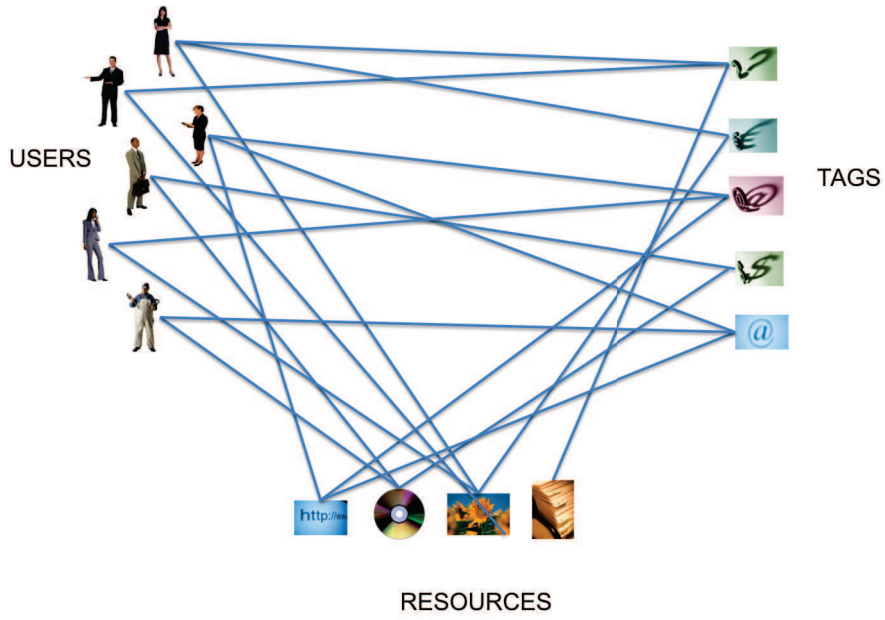


FIGURE 2.2: A folksonomy with hyperlinks

A more detail description about structures of complex networks and their applications can be found in the work of E. Estrada [Estrada, 2011]. Figure 2.4 shows few examples of graphs with different types of edges. A common example of use of a simple graph is co-authorship network where nodes represent authors and edges are added if two authors have written at least one article together. In some cases weighted graphs are used for the same purpose and the weights on each edge is simply the total number of articles written and published by two authors. In another case the same can also be presented as multi-graph where there will be many links between two author nodes corresponding to the numbers of articles published together. Each edge can have an attribute showing the time or venue of publication. Figure 2.4 shows different types of graphs. It also shows a complete graph (fig 2.4(f)) that has all nodes linked directly to each other. That means every two nodes in the network have an edge between them. Real complex networks which are not completely connected most of the time, can have many smaller subgraphs which are completely connected.

Another kind of categorization of networks can be done based on type of node's connectivity. It creates sets of nodes in which a node is never connected to a node in its own

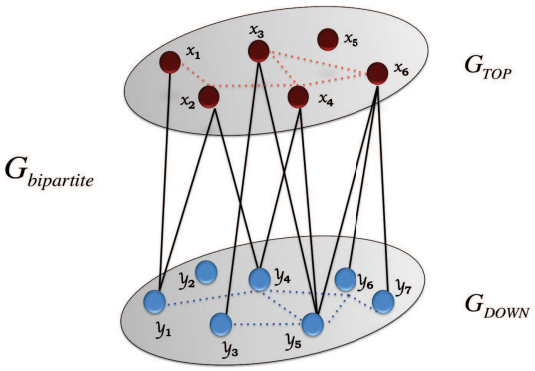


FIGURE 2.3: A bipartite graph with its projections.
$G_{bipartite}$ is the bipartite graph that has two projected graphs G_{TOP} and G_{DOWN}

set but can be connected to any node in other sets. This concepts allows us to categorize graphs as:

1. **Unipartite graphs**: A unipartite graph does not have any node partition. It has only one sets of nodes and each node has a possibility of being connected with any other node. A common example is scientific collaboration network or co-authorship network.

2. **Bipartite graphs**: A bipartite graph has two sets of nodes and a node from one set can only be linked with nodes from other sets. There is no connection within the same set of nodes. One example of bipartite graphs can be author-publication graph where there are two sets of nodes, one representing articles or scientific papers and other will represent authors. There will be links only between authors and papers representing the fact that an author has written a particular paper. Another example is a user-item network, used for market analysis in e-commerce. Here nodes are user (customers) and items (a certain kind of product). A user is linked to an item if it was bought by the user. Network analysis is used here for the purpose of recommending users the items of his/her choice, thereby increasing the chances of selling the products. It is possible to have unipartite projections of the bipartite graph and the links in projected graphs are decided based on the bipartite links. For example author-publication bipartite graph can have two projections. One with only author nodes and the other with only publication nodes. In the projected graphs for authors, two authors are linked if they have a common link to at least one paper in the bipartite graph. Similarly, in the projected graphs for publications, two papers are linked only if they are connected to at least one author in the bipartite graph. This is illustrated with an example in figure 2.3. Figure 2.6 shows some bipartite graphs generated from real network data from Bibsonomy[2].

3. **Tripartite graphs**: Similarly, tripartite graphs have three sets of nodes. Hypergraphs of folksonomy are usually represented as tripartite graphs. They consist of users, resources and user-defined and system-generated tags. Some evident examples of folksonomies are Flicker[3] where resources are images; Youtube[4] where

[2] http://www.bibsonomy.org
[3] http://www.flickr.com
[4] http://www.youtube.com

resources are videos; Bibsonomy[5] and CiteULike[6] with references and online documents as resources; and De.li.ci.ous for sharing bookmarks etc.

Graphs having more than three sets of nodes are called multipartite which is a generalized term used for graphs having one or more partitions, creating more than one sets of nodes. Also the term *k-partite* can be used for the same, k being the number of sets of nodes in the graph. Figure 2.5 shows examples of such graphs. Figure 2.6 shows bipartite graphs created from real data of Bibsonomy.

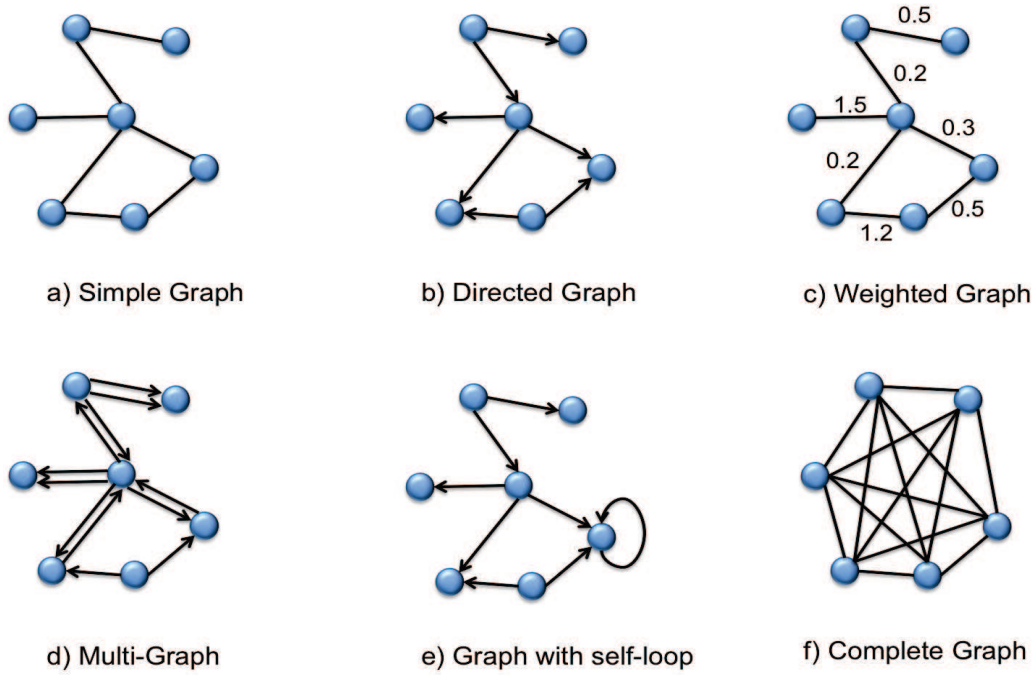


FIGURE 2.4: Different types of edge orientations

2.2.1 Formal definitions

Formally a simple graph is represented as $G =< V, E >$ where $V = v_1, v_2, ..., v_n$, $|V| = N$ is a finite set of nodes and $E \subseteq V \times V$, $E = (v_i, v_j)$, $i \neq j$, $|E| = M$ represents the set of edges in the graph. A graph G can also be represented by a $N \times N$ adjacency matrix A with entries 0 or 1 based on absence or existence of link between two nodes.

$$A_{ij} = \begin{cases} 0 & \text{if } (i,j) \notin E \\ 1 & \text{if } (i,j) \in E \end{cases}$$

Weighted networks can be represented as $G =< V, E, W >$ with an additional parameter $W = w : E \rightarrow R$ where w represents a function that assigns real values as weights to the edges.

In undirected graphs, edge $(v_i, v_j) \Leftrightarrow (v_j, v_i)$ and adjacency matrix A is symmetric (with respect to its diagonal, that consists of all zeros if self loops are not allowed in simple graphs). Conversely, in directed graphs, or digraphs, each edge has an orientation.

[5] http://www.bibsonomy.org
[6] http://www.citeulike.org/

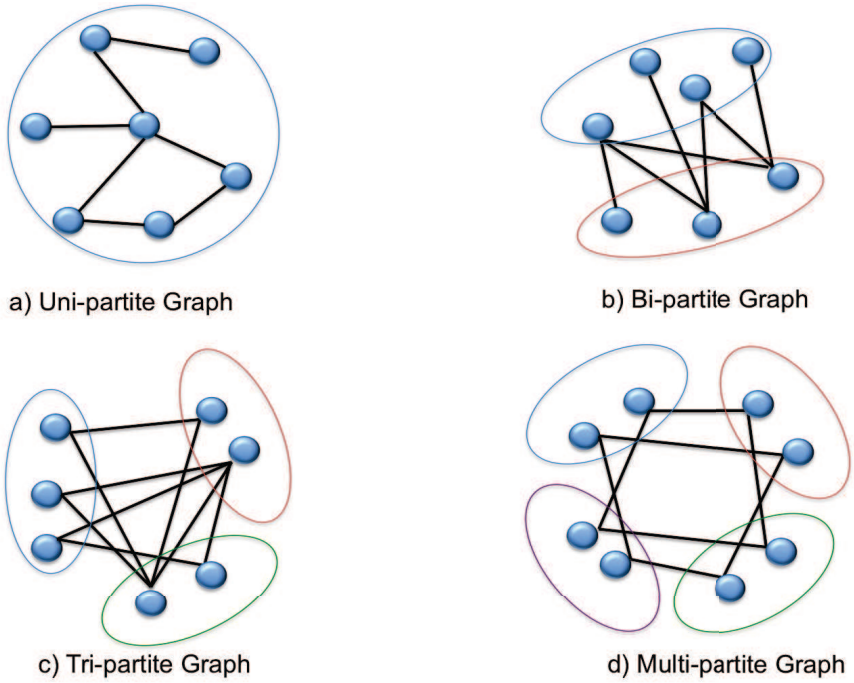


FIGURE 2.5: Different types of graphs

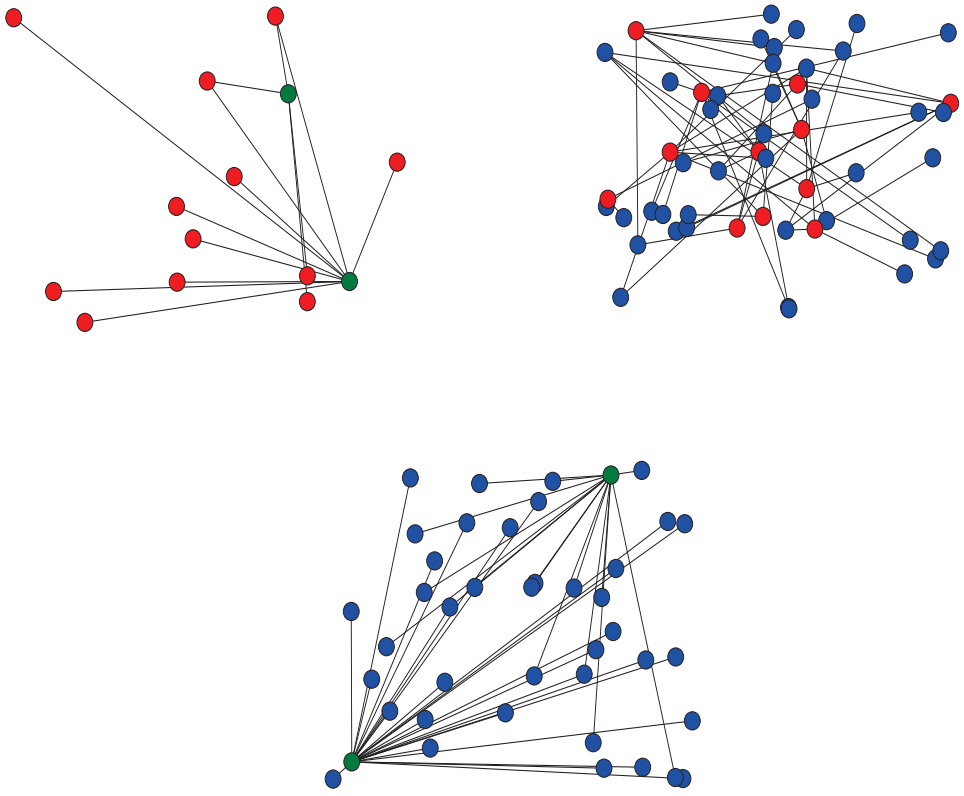


FIGURE 2.6: Bipartite graphs from Bibsonomy dataset for year 1995.
Users are represented by green nodes, resources (articles) by red nodes and tags are blue nodes.
The three graphs represent user-resource, resource-tag and user-tag respectively.

Neighbors of a node in a graph are the set of nodes directly connected to it. Set of neighbors of node v_i is given by:

$$\Gamma(v_i) = \{v_j : (v_i, v_j) \in E\} \tag{2.1}$$

Degree of a node v_i is given by:

$$k_i = \begin{cases} |\Gamma(v_i)| & \text{if } G \text{ is a simple graph} \\ \sum_{v_j \in \Gamma(v_i)} w_{ij} & \text{if } G \text{ is weighted graph} \end{cases} \tag{2.2}$$

In a directed graph the degree is split into *inbound degree* and *outbound degree* computed based on edge directions. In weighted graphs, the term *strength* is used instead of degree.

A *path* between two nodes v_0 and v_k in a simple graph is a non-empty graph $P = (V_p, E_p)$ with a set of distinct nodes $V_p = v_0, v_1, v_2, ..., v_{k-1}, v_k$ and a set of edges $E_p = (v_0, v_1), (v_1, v_2), ..., (v_{k-1}, v_k)$ where $V_p \subseteq V$ and $E_p \subseteq E$. Here V_p is in fact an ordered set of nodes where a node v_i is directly connected to the preceding and succeeding nodes in the list and so on. Same is with the list of edges E_p. *Length* of a path is the number of edges in E_p i.e. $|E_p|$. $paths(v_i, v_j)$ represents a set of all paths and $spaths(v_i, v_j)$ is the set of all *shortest paths* between two nodes v_i and v_j. Shortest paths are the paths having minimum length. It is sometimes possible to have more than one shortest path between any two nodes. *Distance* between two nodes ($dist(v_i, v_j)$) is the length of the shortest path between the two nodes (also known as *geodesic* distance).

A graph G is *connected* if for any two nodes $v_i, v_j \in V$, there exists a path from v_i to v_j. Real networks are not always connected but they are composed of smaller connected subgraphs where each node has at least one path to other nodes. These are known as the *connected components* of a network.

Symbols	Meanings
G	Graph
V	Set of nodes
E	Set of edges
N or n	Number of nodes in the graph
M or m	Number of edges in the graph
$paths(v_i, v_j)$	Set of paths between two nodes v_i and v_j
$spaths(v_i, v_j)$	Set of shortest paths between two nodes v_i and v_j
V_p	Set of nodes in a path p
E_p	Set of edges in a path p
k_i or $deg(v_i)$	Degree of a node v_i
$dist(v_i, v_j)$	Length of the shortest path between two nodes v_i and v_j (Distance)
$\Gamma(v_i)$	Set of neighbors of node v_i

Table 2.2: Notations and terms

2.3 Characteristics of complex networks

Almost all types of complex networks share some common topological characteristics. Some most important ones are listed below.

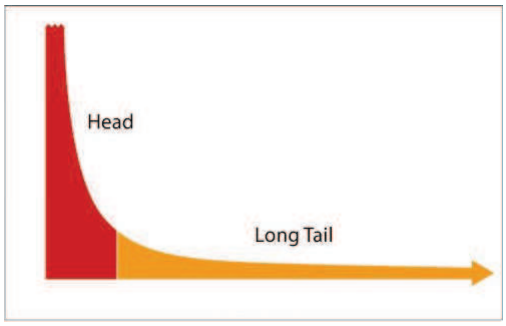


FIGURE 2.7: Power law distribution

Connectedness: Nodes in a complex networks have a tendency to cluster into a number of connected components. Connected components are subgraphs present in the network where there is a path between all pairs of nodes. That means all nodes are directly or indirectly connected to each other. One or two out of these components are extremely large as compared to others. There are many small components in the network.

Degree distribution: Degree distribution is the probability of a node to have k neighbors (degree) in the network. Complex networks have a degree distribution that follows a power law. In statistics, a power law is a functional relationship between two quantities, where one quantity varies as a power of another. So it has small high head and a long tail like structure. Figure 2.7 shows a general shape of power law curve. Such networks are also known as *scale-free* networks, a term coined by Barabasi et al. [Barabási, 2009]. In any network, a degree distribution that corresponds to power law indicates that there are many nodes with very small degree and there are very few of the nodes which have a high value of degree. The coefficient of power law indicates the rate of decrease in degree curve. The higher is the value of power law coefficient, the lesser is the probability of finding a node with high degree.

Clustering coefficient or transitivity: In many real networks, it is often seen that two nodes which are connected to a same node have a tendency of getting connected themselves. In the context of social acquaintance networks, it is equivalent to saying that a friend of my friend is likely to be my friend as well. This property is referred to as transitivity or clustering and is measured by local clustering coefficient. Clustering coefficient measures the probability of the neighbors of a node to be connected to each other. Or in other words it quantifies the presence of triangles in a network. According to the definition proposed by D. J. Watts et al. [Watts and Strogatz, 1998], the local clustering coefficient or local transitivity of a node $v_i \in V$ in a graph $G =< V, E >$, is given as,

$$Cc(v_i) = \frac{N_{triangles}(v_i)}{N_{triples}(v_i)} \tag{2.3}$$

where $N_{triangles}(v_i)$ is the number of triangle having node v_i as one of the nodes and $N_{triples}(v_i)$ is the number of triples formed at node v_i. The local clustering coefficient is actually the proportion of links between the nodes within its neighborhood divided by the number of links that could possibly exist between them. Hence for a undirected graph, $N_{triples}(v_i) = \frac{k_i(k_i-1)}{2}$ and $N_{triangles}(v_i) =\mid \{(v_j, v_k) : v_j, v_k \in \Gamma(v_i), (v_j, v_k) \in E\} \mid$.

The clustering coefficient of a graph is the average of all local clustering coefficients over the total number of nodes i.e.

$$Cc(G) = \frac{1}{|V|} \sum_{v_i \in V} Cc(v_i) \tag{2.4}$$

Complex networks tend to have a very high average clustering coefficient.

Average distance: Distance between two nodes is the length of the shortest paths between the two nodes in a graph. Average distance is the average of all shortest paths in the graph. In real complex networks, this value is often very small. For an unweighted graph G with N nodes, the average distance can be computed as below:

$$Distance_{avg}(G) = \frac{2}{N.(N-1)} \sum_{v_i, v_j \in V} dist(v_i, v_j) \tag{2.5}$$

Diameter: Diameter of a graph is the maximum possible length of a shortest path between any two nodes of a graph. Formally it can be presented as:

$$Diameter(G) = max(\{dist(v_i, v_j) \ \forall \ v_i, v_j \in V\}) \tag{2.6}$$

Computation of diameter has a computational complexity $O(N^2)$. For complex networks, diameters are often very small as compared to the size of the network. Computationally, it is possible to calculate the diameter if the network is connected. If not connected, either maximum value of a shortest path between the nodes (avoiding those pairs which are not connected at all) is taken or the average of the diameters of connected components can be taken into consideration. They also show *small world* phenomenon, a concept proposed by Milgram et al. [Travers and Milgram, 1969]. A *small world* is characterized by very small average distance and a high clustering coefficient. Most of the complex networks have been found to share similar characteristics of high average clustering coefficients and small diameters and average distance. Tables 2.3 and 2.4 illustrate this on a few scientific collaboration networks.

Density: Density of a network is defined as the proportion of links actually present to that of maximum possible potential links. In a graph $G =< V, E >$, the density is given by

$$Density(G) = \frac{2|E|}{|V| \times (|V| - 1)} \tag{2.7}$$

Most of the complex networks have a very less density or in other words they are very sparse.

Community structure: Complex networks always show a tendency to have clusters of nodes in the form of components and communities. Communities are sub-graphs in the network where the nodes share some kind of common interest within them. Roughly speaking, the nodes within a community have more links with each other than with nodes outside the communities. A simple community structure has been illustrated in figure 2.8. These communities can be overlapping or not. Newman and Girvan provide a quantitative measure for such a structure, called modularity in their work [Newman and M. Girvan, 2004].

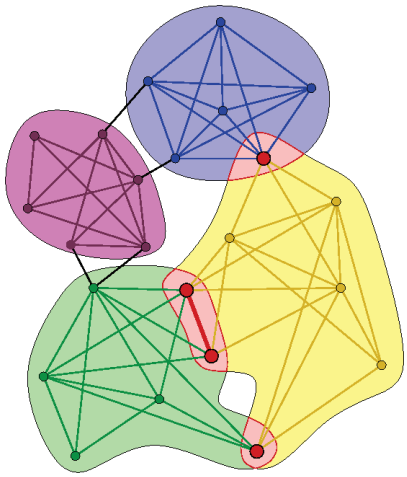


FIGURE 2.8: Community structures in complex networks

In addition to this, the real networks may also have a tendency to change with time. That means there may be appearance and disappearance of nodes and edges with time. This may lead to the change in graph characteristics like average degree, density, average clustering coefficients etc. with time. In [McGlohon et al., 2011] a more detailed description of various statistical properties of different kinds of networks has been provided.

2.4 Network modeling

Network modeling examines hypotheses that explain formation of complex networks. Modeling is mainly done to formalize networks in order to be able to use mathematical and analytical tools to describe the properties of the network in a precise way. Furthermore, this is widely used for prediction of different properties of network. Network generation models allow one to generate graphs from a core or seed graph, that have a structure matching real data properties such as degree distribution, clustering coefficient, diameter etc. Below are three well known and widely used models:

Random graphs: Random graphs are graphs which have a disordered nature of links between nodes. That means a random graph can be created by randomly adding edges to connect nodes. The first probabilistic models for generation of random graphs were proposed by P. Erdős and A. Rényi [Bollobás, 2001; Erdős and Rényi, 1959]. They primarily propose two models. In the first model they generate a network with n nodes and m edges. Starting with n disconnected nodes, the model randomly selects pairs of nodes to connect them, until the number of edges equals to m. The resulting graph is one of the possible outcomes. The second model proposes to generate a random graph by connecting each pair of nodes with a probability $0 < p < 1$. This process may generate an ensemble of graphs, each of which may contain different number of edges. A graph with m edges is found with a probability of $p^m(1-p)^{C-m}$, where $C = \frac{n(n-1)}{2}$ is the total number of possible links. Although the first model seems to be practically more implementable in applications, the second model is mostly used for analytical calculations. Many properties of random graphs proposed by P. Erdős and A. Rényi (ER graphs), have been discovered [Bollobás, 2001], few of which are:

1. If $p > \frac{1}{n}$ the corresponding average degree $k_{avg} = 1$ and when $p >= \frac{\ln(n)}{n}$ almost any random graph created is totally connected.
2. When n is very large, $k_{avg} \simeq p.n$ and the degree distribution $P(k)$ is approximated by Poisson distribution given by

$$P(k) = \frac{{k_{avg}}^k}{k!}.e^{-k_{avg}} \tag{2.8}$$

   For this reason, these graphs are sometimes called Poisson random graphs.
3. The diameter of a random graph varies in a small range of values around $\frac{\ln(n)}{\ln(p.n)} \approx \frac{\ln(n)}{\ln(k_{avg})}$ for $(p.n) \to \inf$ [Chung and Lu, 2001].
4. The clustering coefficient in random graphs are simply equal to p or equivalent to $\frac{k_{avg}}{n}$ [Newman, 2005; Watts and Strogatz, 1998]. This is due the fact that, in a random graph the probabilities of node pairs being connected are by definition independent. So there is no higher probability of two nodes being connected if they have some common neighbors.

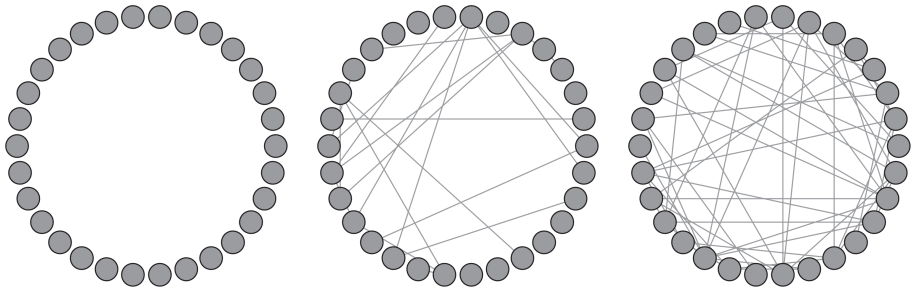


FIGURE 2.9: Random graphs with $n = 30$ and $p = 0, p = 0.02, p = 0.10$ respectively

Small world graphs: *Small world* network model was proposed by D. J. Watts and S. H. Strogatz [Watts and Strogatz, 1998] motivated by the feature of local clustering. They measured that many real-world networks not only have a small average distance, but also a clustering coefficient significantly higher than expected by random chance. A small world network is a graph where many nodes may not be neighbors but most of the nodes can be reached by any other node by a small number of hops or steps. These graphs have a small average shortest path length and a high clustering coefficient. Also, the distance d between any two randomly chosen nodes grows proportionally to the logarithm of the number of nodes n in the network, i.e. $d \propto \ln(n)$. The generation of graph starts with n nodes and l edges per node. Each edge can then be rewired by randomly choosing a node with a probability p. This allows to tune the graph between a regular graph at $p = 0$ and a completely random graph at $p = 1$ with a constraint that each node has a minimum of l connections. For intermediate values of p the procedure generates graphs with low average distance between nodes and high local clustering coefficient. There are also alternative methods to rewire the edges.

Even though small-world model is considerably more relevant to real networks than random graphs, it has many limitations. Firstly, small-world models do not follow the dynamics with which a real network evolves and secondly, the degree distribution of many real networks is not bell shaped, instead it is a power law

indicating the presence of hubs in the networks. However, small world property have been widely found in food web, the World Wide Web [Adamic et al., 2003], power grid networks [Watts and Strogatz, 1998], transportation networks, biological networks [Barabási and Oltvai, 2004], and also in social interaction networks, scientific collaboration networks to name a few.

Scale-free graphs: The models that account for networks with degree distribution deviating from Poisson are called a scale-free models. In many real world networks the degree distribution does not follow a bell curve, but instead does follow a power law.

$$P(k) \sim c\Delta k^{-\gamma} \tag{2.9}$$

where k is node's degree, c is a constant and γ is a positive exponent that mostly varies between two and three. The reason why the exponent fits in that range is still unknown to network scientists and it remains an open question. Having a $P(k)$ that has a decaying tail in the power law (Figure 2.7) means that the vast majority of nodes have low degree and that there exist few nodes, the so-called *hubs*, that have an extremely high connectivity. Hubs play a fundamental role in the evolution, robustness and connectivity of the entire networks. The networks following power law are *scale-free*, because power-laws have the property of having the same functional form at all scales.

There are two types of scale-free models available in the literature: the first one that creates static scale-free networks and the second that creates evolving scale-free networks. The former is simply generated as a special case of random graphs with a given degree distribution. A model that belongs to this category is for instance the *fitness* model [Caldarelli et al., 2002]. It starts from n isolated nodes, and associates at every node i a fitness η_i, which is a real number taken from a fitness distribution $\rho(\eta)$. For each pair of nodes, i and j, a link is drawn with a probability $f(\eta_i, \eta_j)$, with f being a symmetric function of its arguments. The model generates power-law $P(k)$ for various fitness distributions and attaching rules, while it gives ER random graph if $f(\eta_i, \eta_j) = p$ for each i, j.

In the evolving scale-free model, the growth process that determines the structural properties of the network is taken into account. A. Barabasi proposed a network growth model [Barabási, 2009] that was inspired from the formation of the World Wide Web and it is based on two basic factors: growth and *preferential attachment*. The basic idea is that in any network, the nodes with high degree have a higher chance of getting a new link as compared to nodes with lower degree. This is similar to the fact that in a world wide web, the websites with high popularity have a higher probability of acquiring a new hyperlink as compared to websites with low popularity. So an undirected graph is constructed as follows: starting with n_0 nodes, at each time step a new node j is added with $h \leq n_0$ edges added to the network. The probability that a link will connect j to an existing node i is linearly proportional to the actual degree of i. As every new node has h links, the network at time t will have $n = n_0 + t$ nodes and $m = h\Delta t$ edges, corresponding to an average degree $k_{avg} = 2m$ for large times.

2.5 Tasks in network analysis

In [Aggarwal, 2011], the analysis of networks has been divided into two main categories:

1. **Structural analysis:** This type of analysis uses only information about the structure of the network. It is mainly based on linkage information in the network. There is no extra knowledge about the features of nodes and links. This includes statistical analysis of networks, community detection, node classification or labeling, evolution analysis, link prediction and visualization etc. They provide a good overview of the global evolution behavior of the network [Aggarwal, 2011]. A few of purely structure based analysis has been done in [Ahn et al., 2007; Benchettara et al., 2010a; Huang, 2006; Li et al., 2009; Liben-Nowell and Kleinberg, 2007; Liu et al., 2013; Newman, 2012; Yakoubi and Kanawati, 2014].

2. **Content based analysis:** On the contrary content-based analysis exploits other features and content information of the network. This analysis deals with content based mining issues using several different kinds of contents. This largely includes problems of general data mining, text mining, multimedia mining etc. in real networks. Much of the work have been done using structural information. However, some recent research has shown that the inclusion of content information can yield valuable insights about the underlying social network [Hasan et al., 2006; Popescul and Ungar, 2004; Wang et al., 2007].

Complex network analysis involves many different tasks which can require structural information, content information or both. Further, network analysis can exist in different levels namely *Micro*, *Meso* or *Macro* levels which are not necessarily exclusive. Micro level starts from a node, and can extend till small groups of nodes. Thus it includes tasks involving actors (individual nodes), dyads, triads and subgraphs. Meso levels start with a group of nodes and can serve to find relationship between micro and macro level analysis. Macro levels deal with the whole network at the same time.

The different tasks can further exist in various levels as described by S. Wasserman and K. Faust in [Wasserman and Faust, 1994]:

- *Actor level tasks*: This level includes tasks of finding certain kind of importance of a node with respect to other nodes in a network. Common examples are different centralities and prestige of nodes.
- *Dyadic level tasks*: Tasks in this level always involve two nodes. Some of the tasks are finding distance and proximity between two nodes, structural, semantic or other kinds of equivalence between nodes, and finding their tendencies towards reciprocity, probabilities of getting linked (link prediction).
- *Triadic level tasks*: At triadic level, all tasks concern three nodes. Common tasks at this level are finding balance and transitivity or local clustering coefficients.
- *Subset level tasks*: At subset level the tasks involve groups of nodes and often require graph partitioning. The major tasks are finding, characterizing and analysing cliques, cohesive subgroups, components, communities etc.
- *Network level tasks*: Network level tasks take into consideration the whole network to find different properties like connectedness, diameter, centralization, density etc. Visualization and evolution of network can also be put at this level.

We will discuss some of the important tasks below:

Centrality: One of the primary tasks in network analysis is to find important nodes in a network which can have significant roles in diffusion of information or in influencing other nodes in some way. This importance can be based on many different criteria and is often measured as *centrality*. Centrality of a node is the relative importance of the node within a network [Faust and Wasserman, 1992]. There are different types of centralities namely *degree centrality*, *closeness centrality*, *betweenness centrality*, and *eigenvector centrality* etc.

Degree centrality is the most simple centrality measure that is very often used in network analysis. Degree centrality of a node measures the number of nodes to which it is connected or number of links incident at the node. In case of directed network, we have *in-degree* and *out-degree* to represent number of incoming and outgoing links respectively. Often this value is normalized by dividing it by the maximum number of possible links or the maximum possible value of a node's degree in a network. Hence, formally, for a network having N nodes, degree centrality of a node v_i is given by:

$$C_D(v_i) = \frac{deg(v_i)}{N-1} \tag{2.10}$$

Degree centrality has the computational complexity $O(N)$ which makes it very suitable to be used in large scale graphs.

A second view of centrality is *closeness centrality* [Faust and Wasserman, 1992] which takes into account the closeness or distance of a node from all other nodes in a network. A node is central if it is close to all others and can interact quickly with other nodes. This measure is computed as the inverse of sum of distances of a node from all other nodes. Closeness centrality in a connected graph, for $i \neq j$, is given by:

$$C_C(v_i) = \left[\sum_{j=1}^{N} dist(v_i, v_j)\right]^{-1} \tag{2.11}$$

Maximum possible value of closeness centrality for a node can be $(N-1)^{-1}$ when all other nodes are connected to this node and the minimum value can be 0 when none of the nodes are connected to the node. Hence, a standardized version having value between 0 and 1 is:

$$C_C(v_i) = \frac{N-1}{\sum_{j=1}^{N} dist(v_i, v_j)} \tag{2.12}$$

It has a computational complexity of $O(N\log(N) + M)$.

The next important centrality measure is *betweenness centrality* [Faust and Wasserman, 1992; Freeman, 1977] that computes the number of times a node lies on a shortest path between any two nodes of the network other than itself. In any kind of social networks, interactions between two nodes may depend on other nodes especially those which lie on the path between the two nodes. So, a node is central if it is present in maximum possible number of shortest paths in a network. Betweenness centrality of a node is given by:

$$C_B(v_i) = \sum_{i \neq j \neq k} \frac{|spaths(v_j, v_k|v_i)|}{|spaths(v_j, v_k)|} \tag{2.13}$$

Maximum value of this centrality measure is $\frac{(N-1)(N-2)}{2}$. So, a standardized version of this centrality is:

$$C_B(v_i) = \frac{2}{(N-1)(N-2)} \sum_{i \neq j \neq k} \frac{|spaths(v_j, v_k|v_i)|}{|spaths(v_j, v_k)|} \tag{2.14}$$

where $spaths(v_j, v_k|v_i)$ is the set of shortest paths between nodes v_j and v_k having node v_i within. Computational complexity of this centrality is $O(N.M+N^2 \log(N))$ which makes it very difficult to be used in large graphs.

The fourth navigation of centrality is *eigenvector centrality*, which measures the influence of a node in a network. It assigns relative scores to all nodes in a network based on a concept that links to high-scoring nodes contribute more to the score of the node under observation than equal links to low-scoring nodes. Eigenvector centrality of a node v is given by:

$$C_E(v) = \frac{1}{\lambda} \sum_{u \in \Gamma(v)} C_E(u) \tag{2.15}$$

Computation of this centrality has a complexity of $O(N^2)$.

Another concept similar to centrality and used for quantifying the importance of nodes is *Prestige*. This concept is applicable to directed networks. A node is *prestigious* if it is the destination of many incoming links, that means, it has a higher number of incoming links [Faust and Wasserman, 1992]. Google's PageRank is a good example of this and is used to rank web pages (which are nodes in the network) in a webgraph. PageRank is a link analysis algorithm that assigns weights to hyperlinked web page based on the importance of the other web pages linking to that particular web page under observation. It can also be considered as a variant of eigenvector centrality.

Community detection: Communities are groups of nodes that probably share some common properties or play similar roles in a network [Fortunato, 2010]. Community structures can be seen in many real networks such as protein-protein interaction networks [Guimera and Nunes Amaral, 2005; Palla et al., 2005], social communities in social networks [Freeman, 2004], world wide web network [Dourisboure et al., 2007], air transportation networks [Guimerà et al., 2005] etc. Community detection methods try to find dense areas in a network where the nodes share some common interest or characteristics or linkage behavior. This concept is very closely related to that of clustering. It has useful applicability in recommender systems, data structure development, world wide web analysis, classifications of nodes, metabolic network analysis etc. The basic aim of community detection methods is to identify modules and possible hierarchical structures using information from graph topology. In some cases, it is also possible to integrate the content knowledge into community detection process which may leverage the outcome of the approach. We provide more details about community detection algorithms and their use in link prediction in Chapter 6.

Link prediction: Much of the research in complex networks is based on finding linking patterns between nodes. Hence, *link prediction* emerges as an important topic in network analysis. Links represent any kind of association between two nodes of a network. They can be friendship in a social network, co-authorship in academic

collaboration networks, criminal association in a criminal networks, or chemical or bio-chemical interaction in metabolic networks etc. Observing the network until time t, most of the link prediction approaches aim at finding missing links (links at time t) or new links (links at a future time $t+k$). Identification of missing or hidden links helps us to have complete and real structural information of a network, where as finding new links can help us to predict a probable structure of the network at a future point of time. They may also play an important role in many other analysis tasks like detection of communities [Liu et al., 2013; Yan and Gregory, 2009], study of evolution of networks, identification of influential or important nodes [Subbian and Melville, 2011] to name a few. For example, prediction of much likely but non-existent links can be very useful while studying the growth of dynamic networks where new links and nodes continue to get added with time. In the work presented in [Yan and Gregory, 2009], authors have presented their community detection approach using a few vertex similarity measures that are mainly used for link prediction and are closely related to the concept of community structure. They show that the use of these simple measures improves existing community detection algorithms. Another application of link prediction in community detection can be as a task for task-based evaluation of community detection approaches. Link prediction is the main topic of our research presented in this report. Further description and details about different applications of link prediction and various approaches can be found in chapter 3.

Cliques and Hubs: A clique in a network, is a group of nodes which form a complete subgraph. That means any two nodes in this subgraph are directly linked to each other. The task of finding whether there is a clique in a network is NP-hard [Wasserman and Faust, 1994]. A maximum clique is the largest possible size of a clique in a network. Common tasks in network analysis can involve finding the existence of cliques, finding a maximum clique or finding the total number of cliques of size k. In addition, cliques have also been used for performing other tasks in complex network analysis like community detection and link prediction [Liu et al., 2013]. In this work authors have found that links tend to establish cliques in the network. They study the formation of links in the context of missing link prediction. They make three principal observations. First, links are very likely to be established to form a clique in the network. Second links prefer to create larger cliques than smaller ones. And third, links tend to form as many cliques as possible in the networks. Using these observations authors explore link formations in communities. All these also has relevance in studying community and network evolution.

Evolution of networks: Owing to the continuous growth of information, evolution of complex networks has come up as another important issue. In real networks, there can be disappearance and appearance of nodes as well as links with time. This change in structure can affect other characteristics of the network such as communities, link pattern, density etc. They also lead us to deal with enormous amount of data in the form of large-scale networks, which is another topic of research in complex network analysis. One big step in dealing with temporal changes, is introduction of streams or a sequence of network snapshots, each corresponding to a different time stamp, rather than using a static network. A large part of the work in this field are on discovering communities and capturing their changes with time. Other tasks requiring temporal data analysis can be link prediction and information retrieval. There are two major categories of works proposed [Spiliopoulou,

2011]: a) In the first kind of approaches, a single model is generated that captures dynamics of the network by exploiting information of changes in it from one point of time to another. Temporal data is observed as time series that has a beginning and a end. These models provide insights on how the network has evolved and can be used to predict how it will change in future. Such models deliver laws on evolution of network given the past data. b) The second way is where a model is learned at one time point and then it adapts to the data arriving at next time points. These methods observe the temporal data as an endless stream and they deliver insights on how each community evolves. More details about the study of evolution can be found in [Aggarwal, 2011; Spiliopoulou, 2011].

Visualization of networks: Visualization provides an easier and much natural way to summarize the information of a network. It provides a clear and easy way to understand the structure of a network. With growing complexity and sizes of real networks, visualization is becoming an important tool to have insight on the structure and dynamics of complex networks. Many of the present day tools are designed for graph drawing as well as graph analysis and can be used in an interactive way by users. They often combine node-link diagrams with standard statistic visualizations, such as scatterplots and histograms. Many of these tools introduce an iterative approach now. An analyst can have a clear image on the network structure while interacting with these visualizations, clustering and altering the data in the search of more appropriate analytic tools, whose results are fed back into the visualization process. Interesting works in this regard can be found in [Brandes and Wagner, 2004; Correa and Ma, 2011; Freeman, 2000]. A few of graph visualization tools are Gephi[7] [Bastian et al., 2009], UbiGraph[8], GraphViz[9], MuxViz[10], Tulip[11] and Mathematica[12].

2.6 Bibliographical networks

Bibliographical networks contain huge amount of data related to scientific publications in various research domains. They are mostly composed of researchers (or authors), papers or articles, venue of publications, and time (date, year) of publications. They also contain information about organizations or institutes where the researchers are working. They may also contain content-related information for various articles in the form of abstract, tags or key-words. One largely studied network extracted from bibliographical data for scientific collaboration analysis, is co-authorship network where nodes are authors and edges represent the fact that they have written one or more articles together, publication or bibliographical networks where the nodes are documents or articles and edges represent the fact that they have been written by same authors or they have cited each other. Then there are author-publication networks which are represented by bipartite graphs. In these types of networks, there are two sets of nodes i.e. authors and articles and edges exist between an author node and an article node if the author has written that particular article etc. Links can also exist if an author has cited an article in any of his/her work.

[7] http://gephi.github.io/
[8] http://ubietylab.net/ubigraph/
[9] http://www.graphviz.org/
[10] http://muxviz.net/
[11] http://tulip.labri.fr/TulipDrupal/
[12] http://www.wolfram.com/mathematica/

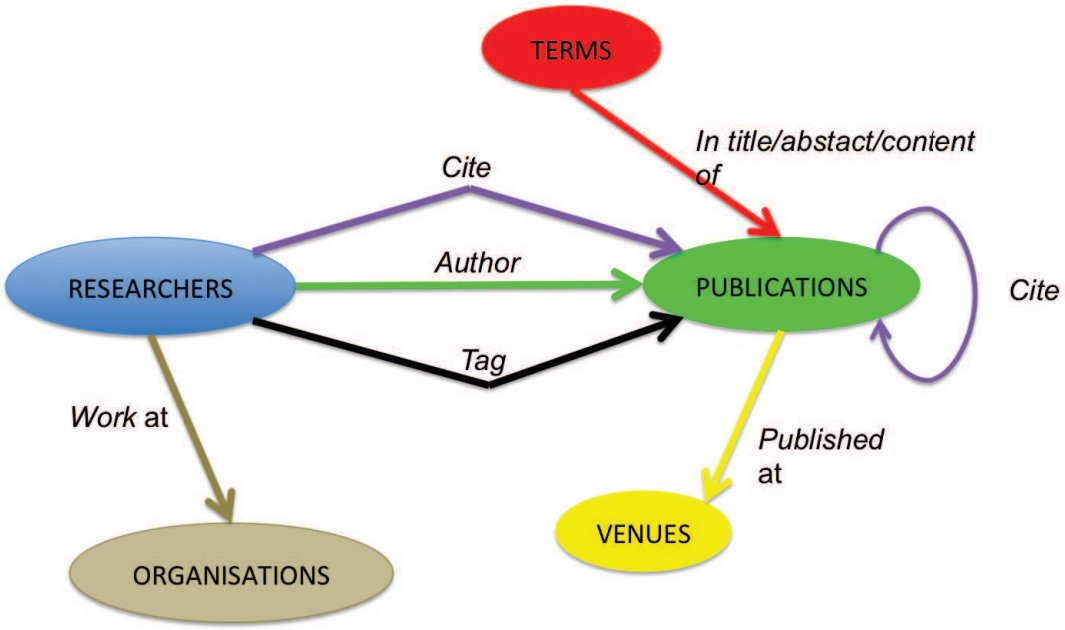


FIGURE 2.10: Bibliographical network

Resource sharing websites also come under these kinds of networks where researchers can share their articles or those which they have read with other researchers. In these networks, the users can also provide keywords for each of the saved article or online link to the article. These kind of networks also come under the category of *folksonomies*. Figure 2.10 shows some of the main components in a bibliographical network and different kinds of possible links in these networks.

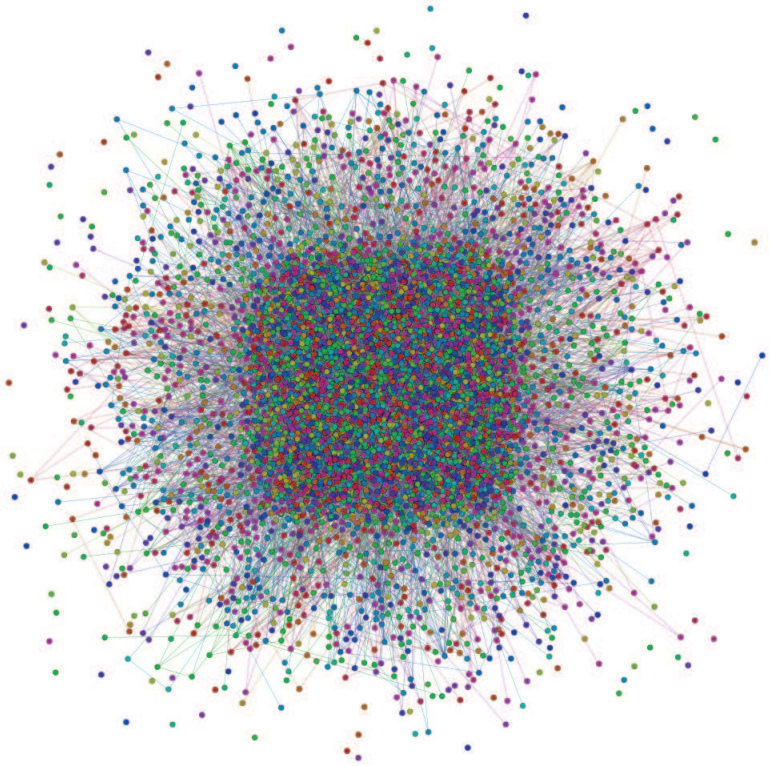
Like any other complex networks, the different kinds of networks formed from scientific collaboration data also have the general characteristics of complex networks. Table 2.3, 2.4 and 2.5 show the graph characteristics of some very well known datasets. Below is the description of a few network data that we have studied.

1. CiteULike[13] is a bibliographic reference sharing website. The data corresponds to years 2005 to 2009. The graph contains three types of nodes *users*, *references* and *tags* which are assigned by users. After pre-processing we get a tripartite graph with 62,513 users, 2,117,337 references and 428,537 tags. While computing the graph features we are considering all nodes to be the same.

2. DBLP[14] is a computer science bibliography website hosted at Universität Trier, in Germany. It contains millions of articles on computer science written by authors from all over the world. The graph we present in the table 2.3, is a co-authorship graph corresponding to year 1970 to 2010. The nodes are authors and the edges represent the fact that two authors have written an article together within a time period taken into consideration. A visualization of a network corresponding to a specific time period is provided in figure 2.11. Other visualization for DBLP co-authorship networks can be found in appendix E.

[13] http://www.citeulike.org/
[14] http://www.dblp.org/

a)



b)

FIGURE 2.11: DBLP Co-authorship network for year a) 1970-1975 and b) 1980-1985

3. Bibsonomy[15] is a social bookmarking and publication sharing system which stores and organizes the bookmarks and publication entries of users and also allows them to assign tags to their entries (we refer them as *resources*). The dataset corresponds to year 1995 to 2008 and contains 1185 users, 22389 resources and 13276 tags. Like for CiteULike, here also we consider all types of nodes together while computing the graph features.

4. Mendeley[16] is a desktop and web program for managing and sharing research papers. It also provides a collaborative platform for researchers who want to collaborate on some projects and discover new knowledge. The dataset used here has 50000 users and 3652285 papers. The graph is a bipartite graph with links between users and papers or references to articles that they have saved and shared.

5. The last data that we are presenting here are obtained from ArXiv[17], which is an archive for electronic pre-prints of scientific papers in the fields of mathematics, physics, astronomy, computer science, quantitative biology, statistics, and quantitative finance etc. and can be accessed online. The graphs that are analyzed here belong to five fields of research Astro-Physics (Astro-Ph), Condense Matter Physics (Cond-Mat), General Relativity and Quantum Cosmology (Gr-Qc), High Energy Physics - Phenomenology (Hep-Ph) and High Energy Physics - Theory (Hep-Th) and are available at Stanford Network Analysis Platform (SNAP[18]). The data corresponds to the time period between January 1993 to April 2003. These graphs were used in the research work of Jure Leskovec et al. [Leskovec et al., 2007]. These are simple co-authorship graphs where nodes represent authors and edges appear only if two authors have written at least one paper together.

	CiteULike	**DBLP**	**Bibsonomy**	**Mendeley**
Nodes	2608387	32849	36850	3702285
Edges	11519879	236897	313475	4848725
Ncc	11171	492	1	142
Density	3.386×10^{-6}	4.391×10^{-3}	4.617×10^{-4}	7.075×10^{-7}

Table 2.3: Different bibliographic networks

	Astro-Ph	**Cond-Mat**	**Gr-Qc**	**Hep-Ph**	**Hep-Th**
Nodes	18772	23133	5242	12008	9877
Edges	198110	93497	14496	118521	25998
Ncc	290	567	355	278	429
Density	1.124×10^{-3}	0.349×10^{-3}	1.055×10^{-3}	1.644×10^{-3}	0.533×10^{-3}
Avg(Cc)	0.6306	0.6334	0.5296	0.6115	0.4714
Avg(degree)	21	8	5	19	5
Diameter	14	15	17	13	18
$Distance_{avg}$	4.194	5.352	6.048	4.673	5.945

Table 2.4: Scientific collaboration networks: Coauthorship graphs for Arxiv Datasets

[15] http://www.bibsonomy.org/
[16] http://www.mendeley.com/
[17] http://arxiv.org/
[18] http://snap.stanford.edu

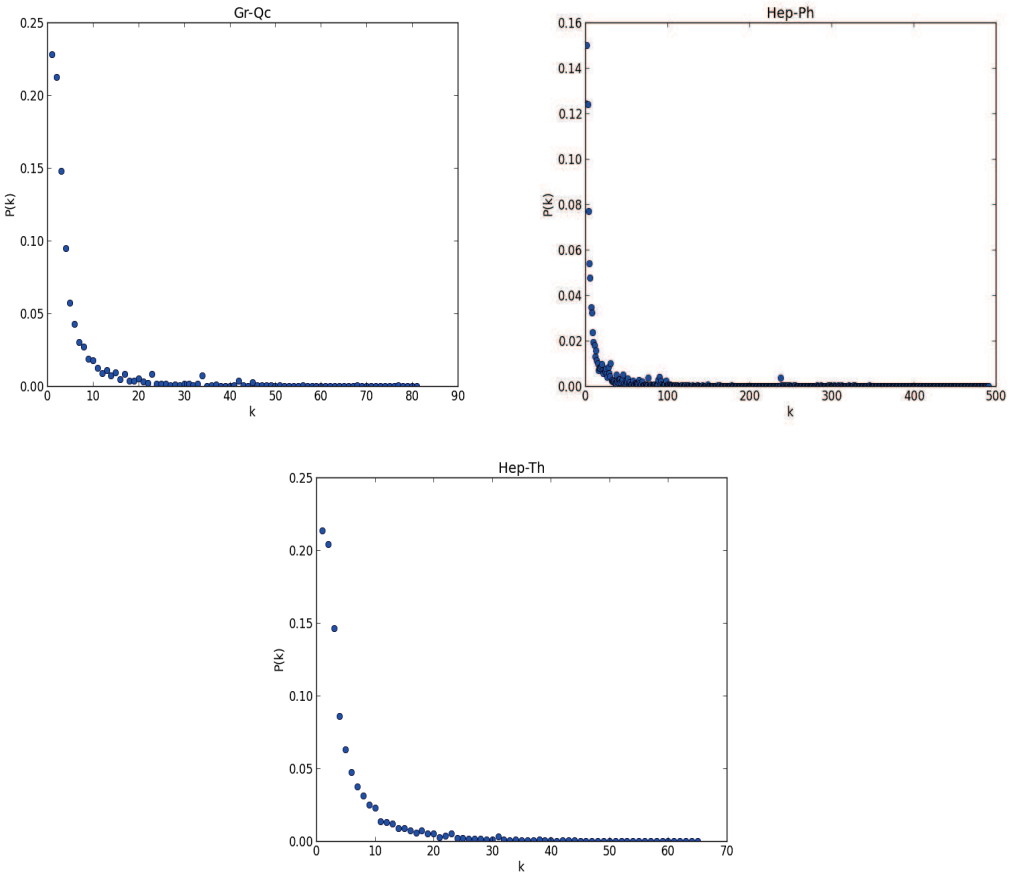


FIGURE 2.12: Degree distribution of Arxiv datasets

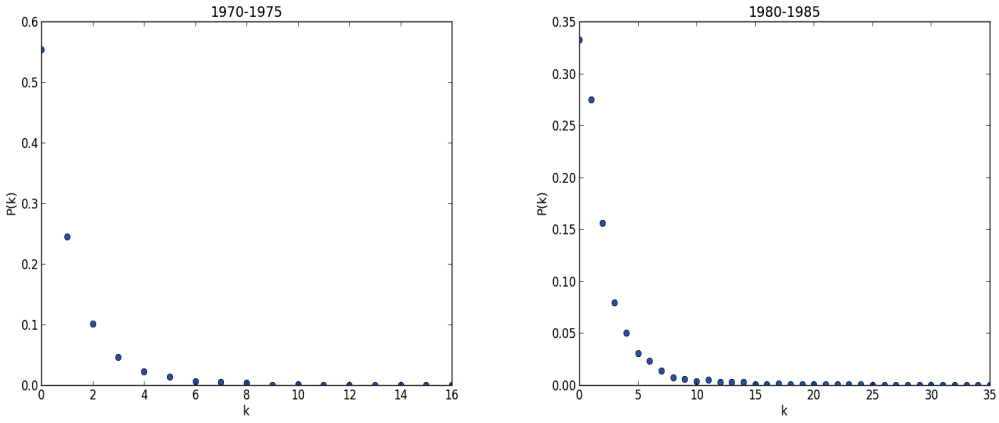


FIGURE 2.13: Degree distribution of DBLP datasets

	DBLP		**Arxiv**		
	1970 - 1975	**1980 - 1985**	**Gr-Qc**	**Hep-Ph**	**Hep-Th**
Nodes	1767	6856	5242	12008	9877
Edges	768	6475	14496	118521	25998
γ	4.376	3.068	2.113	2.080	2.750

TABLE 2.5: Scientific collaboration networks: Power law coefficient for coauthorship graphs for Arxiv and DBLP datasets

Not all of the above presented network datasets were suitable for experimentation in of our research work as some of them lacked time information, a primary requirement for us. In our work we have mainly used DBLP data for link prediction. We have extracted different datasets from the DBLP data corresponding to different periods of time and used it for experimentation and validation of our proposed approaches. For link prediction based tag recommendation (see Appendix A), we have used CiteULike data.

2.7 Conclusion

This chapter summarizes different aspects of complex network analysis. Complex networks are mathematical models of real world networks on which various existing methods and algorithms for analysis can be applied. In the form of graphs, enormous work has been done by using graph theory to extract network information and later apply various kinds of mathematical and statistical algorithm for drawing conclusions on any type of network tasks. In this chapter we summarize the basic structure and characteristics of complex networks. We detail the important characteristics of complex networks with their formal definitions. We present a brief description about the standard methods of network modeling. Then we describe various tasks in complex network analysis. We have provided different levels of these tasks, which primarily classify these tasks into different categories, based on a well known work of S. Wasserman and K. Faust [Wasserman and Faust, 1994]. We describe in detail a few of the important tasks. In our work we focus on the task of *Link Predction*. Lastly we describe, in detail the *bibliographical networks*. Giving a few examples of such networks, we present some of the properties of the networks which are created from real data. In the next chapter, we present the problem of link prediction in complex networks and the state-of-art of various link prediction approaches.

Chapter 3

Link Prediction in Complex Networks: Topological Approaches

3.1 Introduction

Link prediction has attracted the attention of many researchers from different research fields. It consists of estimating the likelihood of existence or appearance of an edge between two unlinked nodes, based on observed links and attributes that contain information about the nodes, edges or the entire graph. It has important applications in many fields including social, biological and information systems etc. Link prediction has been widely used in biological networks like protein interaction network [Airoldi et al., 2006; Eronen and Toivonen, 2012], metabolic networks, food web. It is used for finding missing links and thereby helps in reducing the experimental cost if the predictions are accurate. In social interaction and academic or commercial collaboration networks they can play an important role to predict new associations (new edges) [Fu et al., 2007; Hasan et al., 2006; Liben-Nowell, 2005]. This further has utility in recommendation task: a service provided by almost all social networks and majorly used in e-commerce sites [Huang et al., 2005]. Link prediction can also be helpful in finding hidden links in criminal networks [Clauset et al., 2008; Fire et al., 2013] which is another critical field of research.

Link prediction can be basically of two types: *structural* and *temporal*. Figure 3.1 illustrates the two types of link prediction.

Structural link prediction refers to the problem of finding *missing* or *hidden* links which probably exist in a network [Liben-Nowell and Kleinberg, 2007; Menon and Eklan, 2011; Taskar et al., 2003; Yin et al., 2011]. It mainly focuses on inferring the existence of links that are not directly visible, by using observable data of the network. It has direct application to find unobserved patterns of genes, protein interactions for the medical studies on various diseases like cancer, HIV, Alzheimer etc. [Airoldi et al., 2006; Eronen and Toivonen, 2012]. It can also help to find existing criminal links which often remain hidden in a network [Clauset et al., 2008; Fire et al., 2013].

Temporal link prediction refers to the problem of finding *new links* by studying the temporal history of a network [Benchettara et al., 2010a,b; Berlingerio et al., 2009;

Hasan et al., 2006; Huang and Lin, 2008; Liben-Nowell and Kleinberg, 2007]. So here we have information about the network till time t and the goal will be to predict a new link that may appear at some point of time in future say $t+k$. It has its application primarily in recommendation systems that are being used widely in e-commerce websites for product recommendations, in any search engines to help users with probably relevant terms they might be searching, for recommendation of tags in social resources sharing websites like Flickr[1], YouTube[2], De.li.ci.ous[3] etc. and very commonly used for recommendation of friends in many social networks like Facebook[4] and Twitter[5]. It has another significant use in predicting future collaborations between researchers for academic purposes [Benchettara et al., 2010a,b; Kunegis et al., 2010].

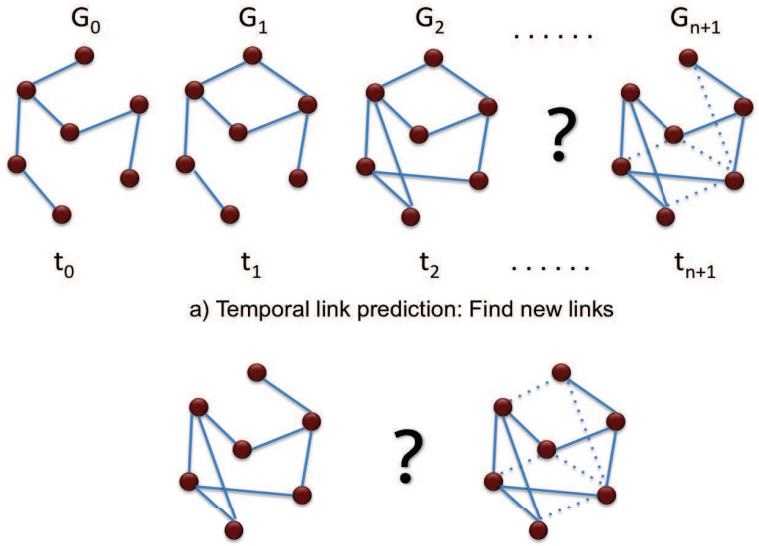


FIGURE 3.1: Link prediction types

Rest of this chapter continues as follows. In section 3.2, we describe the problem of link prediction in a formal way providing a description of notations used, which will be the same in rest of the report. We also provide details about different evaluation methods for link prediction. In section 3.3, we provide detailed description of various link prediction approaches focusing mainly on topological and temporal link prediction methods. We present our way of categorization of various methods that we have studied. Section 3.4 presents a few important challenges in link prediction especially faced by supervised classification based models. Section 3.5 presents our motivation for doing this research work.

[1] http://www.flickr.com
[2] http://www.youtube.com
[3] http://www.delicious.com/
[4] http://www.facebook.com
[5] http://www.twitter.com

3.2 Problem description, notations and evaluation

The problem of prediction of new links (or simply called link prediction problem) refers to a question of inferring the formation of links at a future time, by studying the history of appearance or disappearance of links in a network over a period of time.

In *topology-based link prediction* approaches, only structural properties of the underlying graph are used to implement learning methods and to find a model that will be used to predict links. Suppose we have a complex network graph $G =< V, E >$. The goal of a link prediction approach is to find the likelihood of existence of an edge between two nodes u and v in the form of either a score or rank with conditions that $u, v \in V$ and $(u, v) \notin E$.

For prediction of new links at a certain point of time t_{n+1} having network information till time t_n, the network can be presented as a sequence of graphs representing different snapshots of the network at different points of time $< t_0, t_1, ..., t_n >$. Suppose the temporal sequence of graphs is $G =< G_0, G_1, ..., G_n >$ each having their own sets of nodes and edges. In other words the network can also be represented as a graph $G =< V, E >$ such that $V = \bigcup_{i=0}^{n} V_i$ and $E = \bigcup_{i=0}^{n} E_i$. The goal of a link prediction approach is to find the likelihood of appearance of an edge between any two nodes u and v at a point of time t_{n+1} or t_{n+k}, k being any integer to decide the duration of time for prediction, with conditions that $u, v \in V$ and $(u, v) \notin E$. This is equivalent to finding linking structure of graph G_{n+1} or G_{n+k} assuming that they contain same nodes that have already appeared in any of the graphs during the observation time period i.e. between t_0 and t_n.

Most of traditional link prediction approaches formulate the problem either as label propagation problem where existent and nonexistent links are labeled as *positive* and *negative* respectively or as a problem of existence probability estimation, where links predicted to be existent can have higher existence probabilities [Zhang and Philip, 2014]. Conventionally, if represented in terms of machine learning, a true positive case will be one where a link is classified or predicted as positive and it is actually positive. Same is for false positive case and so on. Table 3.1 presents this better.

	Predicted Positive	**Predicted Negative**
Positive	True Positive (TP)	False Negative (FN)
Negative	False Positive (FP)	True Negative (TN)

Table 3.1: Confusion matrix for link prediction

A link prediction approach can provide either an ordered list of all unobserved and possible links (node pairs) or a score/probability of appearance for each unobserved and possible links. This output is finally used to evaluate the performance of the approach. Different ways for evaluation of link prediction approach is provided next.

3.2.1 Evaluation

As mentioned above, most of the topological link prediction approaches provide as output either ranks of or scores for unlinked pairs of nodes (possible new links) in the concerned network. Out of this, pairs with top-k ranks or top-k highest scores will be considered as

predicted new links. Alternatively, a binary classification like way can be used to label each pair as positive or negative. Many evaluation metrics can be applied on the outputs to measure the performance of the approach. Below is the list of metrics that can be used for such a purpose.

Accuracy: Accuracy can be defined as the number of correctly predicted labels in the test network out of total numbers of possible instances of unobserved links.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.1}$$

Precision: Precision is defined as the proportion of correctly predicted links out of total number predictions made.

$$Precision = \frac{TP}{TP + FP} \tag{3.2}$$

Alternatively where top-k ranks or scores are used as predicted links, the formula becomes

$$Precision_k = \frac{TP}{k} \tag{3.3}$$

Recall: Recall is defined as the proportion of correctly predicted links out of total number of actual new links.

$$Recall = \frac{TP}{TP + FN} \tag{3.4}$$

F1-measure: F1-measure is defined by the harmonic mean of both precision and recall. Formally it is given by

$$F = \frac{2 * Precision * Recall}{Precision + Recall} \tag{3.5}$$

All these above mentioned methods use a fixed threshold (k) to calculate the performance which may not be necessarily available or be the most optimal one. It may be domain dependent and can show a wrongly quantified low or high performance if not selected correctly. To deal with such cases threshold curve based metrics can be used. They have mostly been used in binary classification task to show results. They are especially useful when class distribution is highly imbalanced. Below is the list of such metrics:

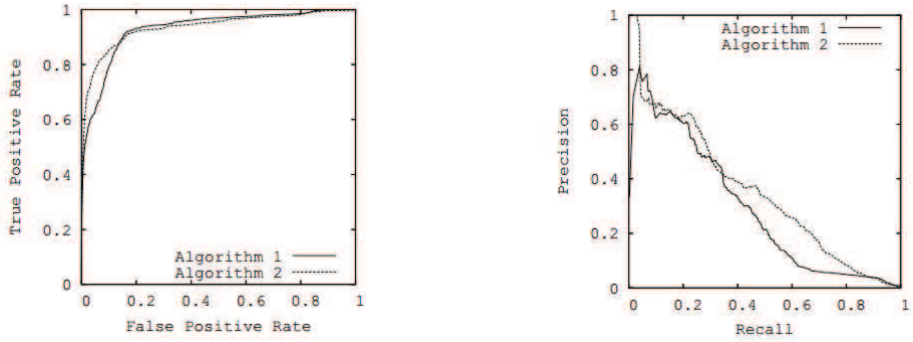
ROC curves: ROC curves are generated by plotting the true positive rate (TPR) against the false positive rate (FPR). It depicts the level of separation between two distributions, one corresponding to the true negatives, and the other corresponding to the true positives, given the scores from a classifier. Formally, $TPR = \frac{TP}{TP+FN}$ and $FPR = \frac{TP}{TN+FP}$.

Area under ROC curve (AUC) is equivalent to the probability of a randomly selected positive instance appearing above a randomly selected negative instance (in terms of scores or ranks). Having a ranked list of all unobserved links (unlinked node pairs), if n independent comparisons are made, of which n_{high} times a true positive has higher score (or rank) and n_{same} times it has the same score (or rank) as the corresponding false positive one, then the AUC can also be computed as suggested in [Lü and Zhou, 2011]

$$AUC = \frac{n_{high} + 0.5 * n_{same}}{n} \tag{3.6}$$

Larger AUC corresponds to better classification results. A value of 0.5 represents pure chance for an identical and independent distribution in a balanced dataset and hence the degree to which the value exceeds 0.5 shows how better a link prediction algorithm is.

Precision-Recall curves: In precision-recall (PR) curves, each point corresponds to a precision and recall value at different score (or rank) threshold. The x-axis is recall and y-axis is precision. Area under PR curve can also be used for the same purpose as AUC i.e. for evaluating a link prediction algorithm. A higher value shows better performance of a model. These curves can give a more discriminative view of performance of different models in presence of extreme class imbalance. Link prediction is such a case as we can have very few number of positive instances as compared to negative instances, hence they have a great utility in evaluating different link prediction algorithms.



a) ROC curve b) Precision-Recall curve

FIGURE 3.2: Samples of ROC and Precision-Recall curves [Davis and Goadrich, 2006]

The major difference between the two curves is that, a PR curve does not account for true negatives and thus is not very much affected by the relative imbalance in class, where as ROC 's measure of TPR and FPR will reflect the influence of heavy class imbalance since the number of negative examples dwarfs the number of positive examples. So in such scenarios, PR curves are much better in illustrating the difference between performances of algorithms especially for predicting true positive instances (minority class in context of link prediction). ROC and AUC will show a very small difference between algorithms. If a model needs to perform equally well on positive and negative classes then ROC is more preferable [Davis and Goadrich, 2006]. A very good analysis on efficient methods to be used for evaluating link prediction approaches is provided in [Lichtenwalter and Chawla, 2012].

3.3 Link prediction approaches

Many link prediction approaches have been proposed in recent years. Some of them use node features or node attributes and some use only the structural information of the graph. The former are known as *node-features based approaches* while the later are known as *topological approaches*. A few of the approaches may use both node-feature information and structural information. Such approaches may be termed as *hybrid approaches*.

In node-features based approaches, one has extra content information regarding the properties or characteristics of nodes. For example in protein interaction graphs, sometimes the features describing the biological properties of proteins are also available. Another example can be a co-citation network where nodes are articles or publications that are linked if they refer each other and we also have the contents of the articles to characterize them with terms in title or abstract and theme. These extra information can be helpful in predicting links between nodes by finding similarities between unlinked nodes and assuming that more is the similarity, the more are the chances of getting connected.

Topological approaches, on the other hand, refer to those works which involve only exploitation of graph structure. They are based on computing the linking probability for pairs of unlinked nodes based on only the graphical features of the network and without any extra information about the individual properties of nodes. They observe how the connections have been established between nodes and how they change over time. Based on former they try to predict a missing link or based on the later they predict a new link. In the work of Zan Huang [Huang, 2006], the authors believe that well-studied topological measures like clustering coefficients and average path length can have direct implications on link prediction. They explore the connection between link prediction and graph topology focusing mainly on the predictive value of the clustering coefficient which is a monadic (involving a single node) measure. They generalize the standard clustering coefficient to capture high order clustering tendencies and they propose a link probability model. In this model, probability of occurrence of a link is determined by the number of cycles (of different lengths) that will be formed by adding this link. The proposed framework consists of a cycle formation model, a method for estimating model's parameters based on observed generalized clustering coefficients, and model-based link prediction. Experimentally, using dataset extracted from Enron email[6] and Facebook[7], they demonstrate that their proposed cycle formation model corresponds closely with the actual link probabilities.

Additional node features are more useful when the network graph is very sparsely connected and not much can be learned from graph topology. But content-based approaches necessarily need the presence of content descriptions of the data. Whereas, topological approaches are very efficient in the absence of content or feature information and are more generic in nature. Both have their own utility and at times a combination of both can come out to give a very good predictor. These kind of approaches can be termed as *hybrid* approaches. For our work, we are more interested in studying and developing topological link prediction approaches due to their generic nature.

There can be various ways of categorizing topological approaches. They can be categorized as *temporal* or *non-temporal/static* based on the fact that whether they take into consideration the dynamic aspect of the network or not. Another way to categorize them can be as *dyadic*, *community/subgraph based*, or *global* approaches based on level at which scores or probabilities are computed. The ones in which scores are locally computed for a pair of unlinked nodes are dyadic approaches. When the same is done in a community or subgraph, the approach can be a community/subgraph based approach. And when the entire graph is taken into account for computing the scores it is a global approach. They can also be classified as *unsupervised*, *supervised* and *semi-supervised*, based on the kind of model learning method used. Unsupervised approaches involve ranking of

[6] http://www.enronemail.com/
[7] www.facebook.com

unlinked node pairs based on some topological attribute scores. They do not necessarily need learning of a model to do prediction and hence do not really need labeled data. However, in link prediction ground truth about the network structure is available most of the time and so supervised methods can be easily implemented. Supervised approaches generate a model using many topological scores for unlinked node pairs and the available ground truth about structure of a network, to predict new links. Also there exist few approaches which use semi-supervised methods of learning where a model is generated by using partially labeled training data.

In the sections below, we describe some of the prominent unsupervised, supervised and semi-supervised approaches.

3.3.1 Unsupervised approaches

There are many unsupervised dyadic methods for predicting links where link scores are computed for unlinked pairs of nodes based on the network structure. These scores represent some kind of similarities between two nodes which can indicate the possibility of having a link between them.

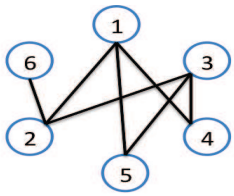
A seminal work on link prediction is the work of D. Liben-Nowell et al. [Liben-Nowell and Kleinberg, 2007]. They specifically analyze academic co-authorship networks. In this work authors have shown that simple topological features representing relationships between pairs of unlinked nodes in a complex network, can be used for predicting formation of new links. They have experimented with many different types of topological features or attributes to characterize pairs of unlinked nodes, mostly concentrating on proximity based attributes. They rank pairs of unlinked nodes by different attribute values and compute their individual performance in link prediction task by considering the top-k ranked node pairs as the predicted links.

Let's consider the case of applying numbers of common neighbors as a topological attribute. Common neighbors counts the numbers of nodes (i.e. neighbors) that are directly connected to both the nodes under observation. Newman used this quantity for studying collaboration networks [Newman, 2004a], while Kossinets used it while analyzing large-scale social networks [Kossinets, 2006]. Formally it is given by:

$$CN(x, y) =\mid \Gamma(x) \cap \Gamma(y) \mid \tag{3.7}$$

Now let $\mathcal{L}$ be the list of pairs of unlinked nodes. We have $\mathcal{L} = \{(x, y)\}$, where $x, y \in V$ and $(x, y) \notin E$, for a graph $G =< V, E >$. Common neighbor score for x and y can be represented by $CN(x, y)$ and is computed as defined earlier. The list $\mathcal{L}$ is then sorted according to the values obtained by applying the common neighbors function to pairs of unlinked nodes. The top k pairs of nodes are then returned as the output of the prediction task. The assumption here is that, the more a pair of unlinked nodes share common neighbors, the more they are likely to have a link in the future. In [Liben-Nowell and Kleinberg, 2007], k is equal to the number of really appearing new links. Other types of topological measures can be applied for the same purpose. Figure 3.3 illustrates the same with the help of a sample graph.

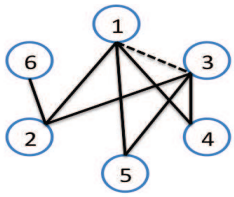
This proximity or similarity between two unlinked nodes can be computed using various other types of topological features. These features can be based on *neighbors* of nodes



	$CN = \|\Gamma(x) \cap \Gamma(y)\|$	*Ranked List*
(1, 3)	3	(1, 3)
(1, 6)	1	(2, 4)
(2, 4)	2	(2, 5)
(2, 5)	2	(4, 5)
(3, 6)	1	(1, 6)
(4, 5)	2	(3, 6)
(4, 6)	0	(4, 6)
(5, 6)	0	(5, 6)

Predicted Edges

(a) Prediction of links taking $k = 3$



	$CN = \|\Gamma(x) \cap \Gamma(y)\|$	*Ranked List*
(1, 3)	3	(1, 3)
(1, 6)	1	(2, 4)
(2, 4)	2	(2, 5)
(2, 5)	2	(4, 5)
(3, 6)	1	(1, 6)
(4, 5)	2	(3, 6)
(4, 6)	0	(4, 6)
(5, 6)	0	(5, 6)

Actual New Edge

(b) Actual new links

FIGURE 3.3: Prediction of links based on numbers of common neighors (CN)

(Common neighbors is the most simple example of this type) or *paths* connecting the nodes. They can also be found by aggregating the topological features of nodes in some way (sum, product, average etc.). In the following subsections, we describe some of the topological similarity metrics of each type.

3.3.1.1 Neighborhood based features

Many of the link prediction approaches are based on the idea that two nodes are similar and more likely to form a link in the future if they are connected to same or similar neighbors. That means their sets of neighbors have large overlap. For example friendship formation through common acquaintances has often been used to justify this concept in

many researches. Apart from common neighbors, there are many other measures based on local neighborhood of the nodes. These are listed below.

Jaccard coefficient: Jaccard coefficient calculates the ratio of number of common neighbors to that of the total number of neighbors of the two nodes [Jaccard, 1901]. Here they normalize the similarity score computed by common neighbors by dividing it with total number of neighbors of the two concerned nodes. Conceptually, it is equivalent to finding the probability that a common neighbor is selected when a random selection of node is done on the combined neighbors set of the two nodes in question. This coefficient is defined as below:

$$JC(x,y) = \frac{\mid \Gamma(x) \cap \Gamma(y) \mid}{\mid \Gamma(x) \cup \Gamma(y) \mid} \tag{3.8}$$

Adamic Adar coefficient: This coefficient was proposed by L. Adamic and E. Adar to find similarity between two web pages [Adamic and Adar, 2003]. For two web pages x and y, sharing a set of features z, this coefficient is computed as:

$$\sum_{z:feature\ shared\ by\ x\ and\ y} \frac{1}{log(frequency(z))} \tag{3.9}$$

In a general sense it is a meta-measure that can be calculated for any two nodes (actors) in a network and for a variety of topological features. In the context of link prediction it was used by Liben-Nowell using common neighbors as the topological feature [Liben-Nowell and Kleinberg, 2007]. This metric proposes to weight the common neighbors based on their connectivity while computing the score. It gives more weight to less connected neighbors increasing their contribution in the score. Formally it can be presented as:

$$AA(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{log \mid \Gamma(z) \mid} \tag{3.10}$$

Resource allocation: This metric is based on resource allocation dynamics on complex networks [Ou et al., 2007]. Like Adamic Adar coefficient, this index also depresses the contribution of high-degree common neighbors. It is formally given as:

$$RA(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\mid \Gamma(z) \mid} \tag{3.11}$$

Neighbor's clustering coefficient: This neighborhood based measure is based on the clustering coefficients of common neighbors. This metric computes the clustering coefficient for the common neighbors of any two nodes, which can then be aggregated using any functions like average, maximum, minimum etc. and the value thus found can be used for prediction of links. The assumption here is that, if the common neighbors of two unlinked nodes have a high clustering coefficient, it can imply a greater linking probability between the two nodes. A way of computing the coefficient is

$$NCF(x,y) = \frac{\sum_{z \in \Gamma(x) \cap \Gamma(y)} Cc(z)}{\mid \Gamma(x) \cap \Gamma(y) \mid} \tag{3.12}$$

$Cc(z)$ is the local transitivity or clustering coefficient of the node z (See equation 2.5 in chapter 2). Such a measure has been used in a Naive Bayes based link prediction models as the conditional probability of having a pair of node linked [Liu et al., 2011; Tan et al., 2014].

3.3.1.2 Path based features

Path based features rely on the paths (see chapter 2, section 2.2) between unlinked node pairs. They may use the lengths of path or the time required to cover those paths to reach from one node to another. The basic idea is that two nodes can be similar if they have less distance between them. There are two major categories in this: distance based features and random walk based features. Below is the description of various methods falling in the two categories.

Distance based
These features are mostly based on shortest paths or paths of specific lengths. They make use of either number or lengths of shortest paths to find the proximity of between two nodes.

Shortest path length: It is equal to the number of edges in the shortest path between x and y in G. It is also known as the geodesic distance between nodes. More is the distance, lesser is the similarity between the nodes and also the chance of having a link between them. This metric captures the fact that the path between two nodes in a social network can affect the formation of a link between them following the fact that friend of a friend can be a friend in a social network.

Katz's index: One of the well known scoring index, commonly known as *Katz index*, has been proposed by L. Katz [Katz, 1953]. It is based on paths between nodes in a graph. It sums over a collection of paths and is exponentially damped by length to give shorter paths more weights. Mathematically it is defined as,

$$Katz(x,y) = \sum_{l=1}^{\infty} \beta^{\ell} \times \mid path_{x,y}^{(\ell)} \mid \tag{3.13}$$

where $path_{x,y}^{(\ell)}$ is the number of paths between x and y of length ℓ and β is a positive parameter (i.e. damping factor) having value between $[0,1]$, which favors shortest paths. The same can be presented using adjacency matrix as:

$$Katz(x,y) = \beta A_{xy} + \beta^2 (A^2)_{xy} + \beta^3 (A^3)_{xy} + \ldots \tag{3.14}$$

A_{xy} is the adjacency matrix where the values are either 1 or 0 based on whether x and y are directly connected. $(A^2)_{xy}$ is the matrix showing numbers of paths of length 2 between x and y and so on. A very small β leads to a score close to number of common neighbors because long paths contribute very little. So the matrix showing Katz score between all pairs of nodes can be found as:

$$K = (I - \beta A)^{-1} - I \tag{3.15}$$

β must be lower than the reciprocal of the largest eigenvalue of matrix A to ensure the convergence of above given equations demonstrated in [Lü and Zhou, 2011]. The

computational complexity of this measure in $O(N^3)$. Due to high computational complexity, sometimes it becomes difficult to use Katz coefficient especially in large networks. In such cases one can chose to stop after a certain length l_{max}. This is known as truncated Katz coefficient [Lü and Zhou, 2011] and is computed as:

$$TKatz(x,y) = \sum_{l=1}^{l_{max}} \beta^{\ell} \times \mid path_{x,y}^{(\ell)} \mid \tag{3.16}$$

In terms of matrix, it is:

$$TKatz(x,y) = \beta A_{xy} + \beta^2 (A^2)_{xy} + \beta^3 (A^3)_{xy} + \ldots + \beta^{l_{max}} (A^{l_{max}})_{xy} \tag{3.17}$$

When l_{max} is extremely large or nearer to infinity, this measure is equivalent to Katz coefficient.

Path betweenness centrality: We propose a new path based measure to be used for link prediction. The idea is to capture the importance of the shortest paths between two unlinked nodes. The importance of a path is computed in terms of centrality which can be defined as the fraction of shortest paths in a graph that contain this observed path within them.

Let $G =< V, E >$ be a network, with V is the set of nodes and E is the set of edges. Let $paths(u,v)$ be the set of shortest paths between nodes u and v. We say that $nsp(u,v) = |paths(u,v)|$ is the number of shortest paths and $dist(u,v)$ is the shortest path length. The betweenness centrality for a path $p \in paths(u,v)$ is defined as :

$$c_B(p) = \sum_{s,t \in V and\ (s,t) \neq (u,v)} \frac{nsp(s,t \mid p)}{nsp(s,t)} \tag{3.18}$$

$nsp(s,t \mid p)$ is the number of shortest paths between s and t passing through path p. If the number of shortest paths between two nodes is more than one, then the path betweenness centrality of a pair of nodes is the maximum of the centralities found for all the shortest paths between them. Another way is to apply the average, sum, min (minimum), max (maximum) or any other suitable function to aggregate these multiple centrality. But for the time being we will apply max function. More details about this measure and its use for link prediction is provided in appendix B. Experimental details about its performance for co-authorship prediction is also reported in the same appendix.

Random walk based

These methods use paths of any length randomly chosen while traveling from one node to another.

Matrix forest index: Matrix forest index computes the similarity between two nodes as the ratio of number of spanning rooted forests such that the two nodes belong to the same tree rooted at one of the nodes of all the spanning rooted forests of the network. It can be computed as $M = (I - L)^{-1}$, I being the identity matrix and $L = D - A$ is the Laplacian matrix of the network where D is the degree matrix and A is the adjacency matrix [Chebotarev and Shamis, 1997]. This index was used for collaborative recommendation task in the work of F. Fouss et al. [Fouss et al., 2006].

Hitting time and commute time: Hitting time is a random walks based feature that counts the time required by a random walker to go from node x to node y in a graph. It is defined as the expected number of steps required for a random walker to walk from one node to the other. Shorter hitting time may denote the nodes are similar and can have higher chance of linking in future. As this metric is not symmetrical, often for undirected graphs, average commute time is used instead. If $HT(x, y)$ is the hitting time to reach node y from node x, average commute time is given by

$$CT(x, y) = HT(x, y) + HT(y, x) \tag{3.19}$$

A negated value of hitting or commute time can be used as a score for predicting links. A major disadvantage of using these measures is their sensitive dependence on parts of graph far away from nodes x and y even when x and y are connected by very short paths.

Rooted Pagerank: Pagerank denotes the importance of a node x by summing up the importance of all other nodes linked to x. This importance can also be represented by stationary distribution weight of a node. This feature can be altered to find a similarity score between two nodes and is termed as *rooted pagerank* in [Liben-Nowell and Kleinberg, 2007]. The similarity between two nodes x and y is measured as the stationary probability of y in a random walk that returns to x with probability $1 - \alpha$ in each step, moving to a random neighbor with probability α. Rooted pagerank for all node pairs can be computed as follows.

$$RPR = (1 - \alpha)(I - \alpha N)^{-1} \tag{3.20}$$

where $N = DA^{-1}$ is adjacency matrix with row sums normalized to 1 and D is the diagonal degree matrix.

PropFlow: PropFlow captures the probability that a restricted random walk starting from one node x ends at another node y in l or less steps using link weights as the transition probabilities. The restriction is that a walk terminates on reaching y or on revisiting any node including x. The walk selects links based on their weights which produces a score to estimate likelihood of new links. This measure is a more localized measure of propagation and is insensitive to topological noise far from the source node [Lichtenwalter et al., 2010].

3.3.1.3 Aggregation of node topological features

These category advocates the idea that two nodes can be similar if they have similar topological features. The individual node features can thus be aggregated to use them suitably to characterize pairs of nodes and use it for link prediction. Various ways of aggregation can be used starting from simple *min*, *max*, *sum* and *product* to more complex ones.

Preferential attachment is a very well known metric which combines the degrees of the two nodes and was proposed by A. L. Barabasi in the context of analyzing scaling in random networks [Barabasi and Albert, 1999]. The work proposes that the probability of appearance of a new link is directly proportional to the degree of the observed nodes. So, it can be used as a score for predicting links and is computed as below:

$$PA(x, y) = \mid k_x \times k_y \mid \tag{3.21}$$

For a simple un-directed and un-weighted graph the degree of a node is equal to the number of neighbors i.e. $k_x = \Gamma(x)$.

Similarly many other aggregation based measures that can be used for link prediction have been listed next.

Sum of neighbors: In the work of [Hasan et al., 2006], the authors have used sum of neighbors as a topological feature for characterizing an unlinked node pairs. Formally, it can be defined as

$$Sum_{CN}(x, y) = \Gamma(x) + \Gamma(y) \tag{3.22}$$

It represents the social connectivity of the nodes. It advocates the fact that the more connected two nodes are, the more will be their likelihood of forming new links.

Aggregation of clustering coefficients: As described in chapter 2, clustering coefficients of a node quantifies the probability of the neighbors of the node to get connected to each other.

$$cf(x) = \frac{3 \times \#Triangles\ adjacent\ to\ x}{\#Possible\ triples\ adjacent\ to\ x} \tag{3.23}$$

This property can also be used for link prediction by taking an aggregation (sum or product) of the clustering coefficients of two unlinked nodes. So the similarity score for any two nodes x and y will be

$$PCF(x, y) = cf(x) \times cf(y) \quad or \quad PCF(x, y) = cf(x) + cf(y) \tag{3.24}$$

It presents the idea that two actors in the network that have a high tendency of having links between their respective neighbors must be very active in forming links themselves and may end up forming a link with each other too in the future.

3.3.2 Supervised approaches

The ground truth about the existence or absence of links is almost always available from the history of the network which makes it suitable to be used with supervised algorithms. Moreover a classifier trained with only one topological attribute can outperform rankings generated by sorting the node pairs based on scores of the attribute if there are multiple differentiating boundaries in the domain topological attribute value. Also supervised algorithms are able to capture important inter-dependency relationships between topological properties [Lichtenwalter et al., 2010].

3.3.2.1 Supervised machine learning based approaches

Following the work of Liben-Nowell et al. [Liben-Nowell and Kleinberg, 2007] many attempts were made to combine the effects of individual topological metrics in order to enhance the overall prediction performance of the approach. Most of these works involve the application of *Machine Learning* algorithms.

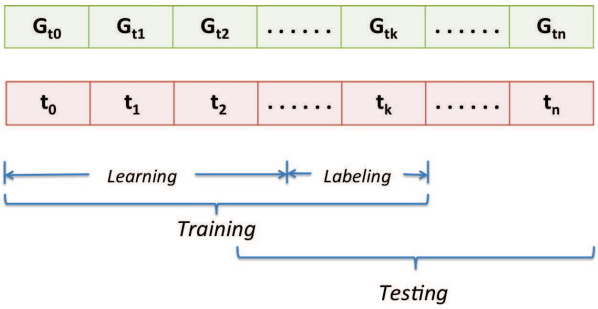
In machine learning language, the unlinked pairs of nodes are called *examples* or *instances*. If the time aspect of the network are to be considered, then the examples can be generated as follows. Let $G =< G_1, \ldots, G_n >$ be a temporal sequence of an evolving network. The whole sequence is divided into two parts: *training* and *testing*. Each part is then again divided into two sub-sequences one for generation of examples and another for labeling those examples. Thus, in training we shall have a *learning* and *labeling* phases resulting in graphs namely G_{learn} and G_{label} generated by making union of the temporal sequences of the graphs for the corresponding time slots. The training data is constructed as follows. An *example* for learning is a couple of nodes (x, y) that are not linked in G_{learn} but both belonging to the same connected component. The class is obtained by checking whether the couple of nodes is indeed connected in G_{label}. If such a connection exists then it will be a *positive* example in the supervised learning task and if no connection exists, it will be a *negative* example [Benchettara et al., 2010b]. Thus, examples are generated from these graphs for both training and testing. These examples are also characterized by a given number of topological attributes computed on learning (or test) graphs. Figure 3.4 illustrates the process diagrammatically.

The first approach we studied is the one proposed by Mohammad Al Hasan et al. in [Hasan et al., 2006]. They convert the problem of link prediction in graphs into a binary classification problem where examples are unlinked node pairs and are characterized by a vector of topological attribute values. Having a graph for generation of examples and computation of topological attributes and one for labeling as described before, one can construct set of instances to be fed to any classification algorithm to generate a model which can further be used to classify test instances with same vectors of attributes. An example of this is given in figure 3.5. The network has 6 nodes and 7 edges during learning period and one new edge during labeling. So we get here one positive example and 7 negative examples. They are characterized by two topological attributes namely the common neighbors (CN) and Jaccard coefficient (JC). The authors also make a comparative analysis on the suitability of many learning algorithms to be used in link prediction based on their prediction performance. Another interesting study that they made was to use ranks of the attributes based on various factors in order to compare and judge their relative strength in a prediction task.

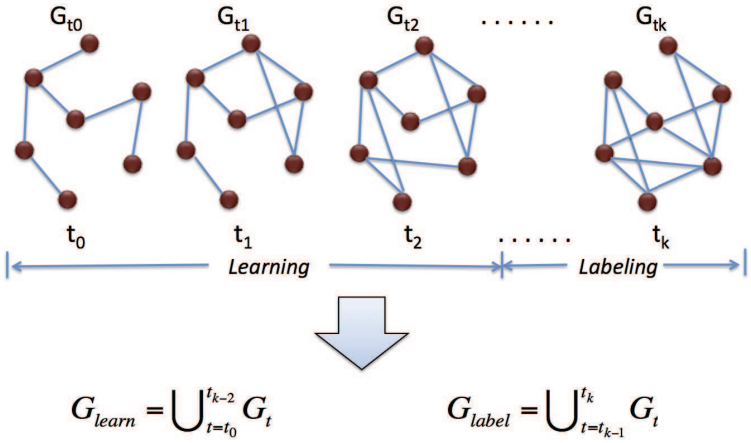
The work done by N. Benchettara et al. [Benchettara et al., 2010b] is a temporal approach for link prediction based on supervised machine learning where link prediction is done by using Decision tree algorithm with boosting. This is a dynamic approach where the evolution of the network is taken into account. The authors have very well proved the enhancement in the prediction result by considering the dynamic aspects of the network. Their work is mostly based on bipartite graphs and they introduce the concept of indirect topological measures computed using the projected graphs. For a bipartite graph $G =< V_1, V_2, E_{bip} >$, the projected graphs will be $G_1 =< V_1, E_1 >$ and $G_2 =< V_2, E_2 >$. For any topological attribute $X(i, j)$, if it is directly computed on G_1 for any two nodes $i, j \in V_1$, it becomes a direct attribute for projected graph G_1 and we represent it as $X_{G_1}(i, j)$. The associated indirect attribute is computed on other projected graph G_2 as:

$$X_{indirect}(i, j) = f_{x \in \Gamma_G(i), y \in \Gamma_G(j)}(X_{G_2}(x, y)) \tag{3.25}$$

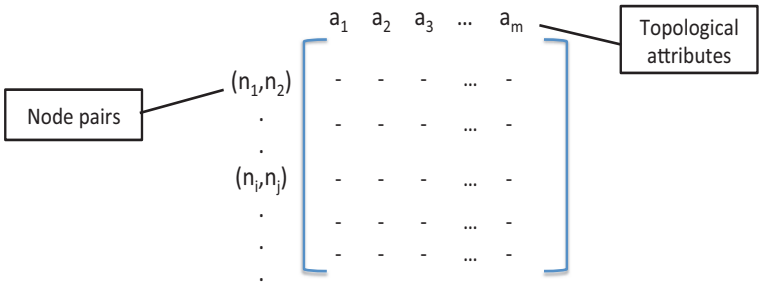
f is some aggregate function like $min, max, average, \ldots.$ etc. and $x, y \in V_2$. Authors show how the use of indirect attribute greatly affects the final prediction result in a positive way.



(a) Division of time for learning, labeling and test



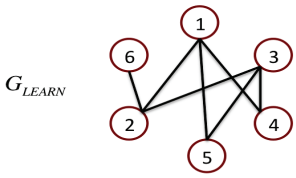
(b) Construction of learning and labeling graphs

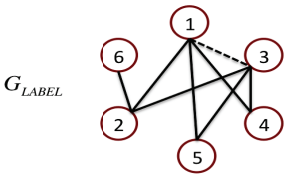


(c) Generation of examples from graphs

FIGURE 3.4: Creation of examples for supervised machine learning

Another work that uses supervised machine learning for classifying unlinked node pairs to predict missing links is the work proposed by M. Fire et al. [Fire et al., 2011]. In this paper the authors propose a set of simple and computationally efficient topological features to be used for link prediction in various social networks. They use various neighborhood based features and their variants, edge subgraph features and shortest path length in both directed and undirected graphs. They also propose a variant of Katz measure namely *friends measure*. This feature estimates how well friends of two users know each other. Here they assume that two nodes have higher chances of getting connected if they have higher number of connections within their neighborhoods. Clearly, *friends measure* is a specific case of Katz measure where $\beta = 1$ and $l = 2$. The authors applied various supervised machine learning algorithms to generate a model from two types of training sets: *easy set* generated by randomly choosing 25000 positive and 25000





Examples	$CN = \|\Gamma(x) \cap \Gamma(y)\|$	$JC = \frac{\|\Gamma(x) \cap \Gamma(y)\|}{\|\Gamma(x) \cup \Gamma(y)\|}$	*Class*
(1 , 3)	3	1	Positive
(1 , 6)	1	0.33	Negative
(2 , 4)	2	0.66	Negative
(2 , 5)	2	0.66	Negative
(3 , 6)	1	0.33	Negative
(4 , 5)	2	1	Negative
(4 , 6)	0	0	Negative
(5 , 6)	0	0	Negative

FIGURE 3.5: Generation of examples on a sample graph

negative edges and *hard set* that contains the same number of randomly chosen positive and negative edges but now the negative edges have a distance of two hops. By positive edge we mean that the edges exist in the observed graph and negative are those that do not exist in the graph. They evaluate their model using 10-fold cross validation.

3.3.2.2 Matrix based approaches

Matrix based approaches represent a network in the form of an adjacency matrix representing link relationships between nodes. It is a $n \times n$ matrix represented by say A for a graph $G =< V, E >$ and $n = |V|$. The values of the matrix are either 0 or 1 showing absence or presence or they are numbers of edges between two nodes. Matrix factorization models map the nodes to a joint latent factor space such that the interaction between nodes are modeled as inner products in that space.

In the work presented by A.K. Menon et al. [Menon and Eklan, 2011], the authors use supervised matrix factorization approach for link prediction. The model learns latent features from the structure of a graph. The authors show that combining these latent features with explicit node features and also with outputs of other models can be used to make better predictions. They propose a new approach to deal with class imbalance problem by directly optimizing a ranking loss function. The model is optimized with stochastic gradient descent and also scales to large graphs.

Another work on temporal link prediction given in [Gao et al., 2011] is a model based on matrix factorization. Authors exploit multiple information sources in the network to predict link occurrence probabilities as a function of time. They propose a unique model combining global network structure, content information of nodes and local proximity information. For combining the temporal information of the network, they use a weighted exponentially decaying model to build an aggregate weighted link matrix over a set of T time slices.

3.3.2.3 Probabilistic approaches

Probabilistic models are supervised models that primarily use the Bayesian concept, to obtain a co-occurrence probability of un-connected node pairs. These models aim at abstracting a structure from observed data of the network and then predict links by using the learned model. Given a target network, a probablistic model will optimize a target function, to establish a model composed of a group of parameters, which can best fit the observed data of the target network. The probability of existence of a link between two nodes x and y is then estimated by the conditional probability $P(A_{x,y} = 1|\Theta)$ where A is the adjacency matrix representing the network and Θ is the set of parameters.

C. Wang et al. [Wang et al., 2007] have presented a *local probabilistic* graphical model to estimate joint co-occurrence probability of link formations. The method explores probabilistic models to enhance the result of topological and semantic models. The first step of the approach is to identify a *central neighborhood set* for a pair of nodes (say x and y) for which the linking probability is to be estimated. There can be many ways of finding this neighborhood set. A straightforward option is to consider shortest paths. All nodes in a shortest path between candidate nodes can be a part of their central neighborhood set. So there is possibility of having more than one central neighborhood sets. The second step is to learn a maximum entropy Markov's random field (MRF) model that estimates the joint probability of the nodes inside the central neighborhood set. These models are local MRF models constrained on non-derivable frequent itemsets from the local neighborhood. The co-occurrence probability, thus found can be used as a feature and can be used in any supervised learning algorithm along with topological and semantic features.

A *hierarchical probabilistic* model has been proposed by A. Clauset et al. [Clauset et al., 2008]. This model involves a hierarchical organization of nodes in the network, in which nodes are divided into groups that further subdivide into groups of groups and so on. The learning task uses observed network data to fit the most likely hierarchical structure through statistical inference, to find missing links. In this work, statistical inference is obtained by using Maximum likelihood approach and Monte Carlo sampling algorithm. A Markov's chain Monte Carlo method is used to sample possible dendrograms and then, one of these sampled dendrogram that is most likely to explain the network structure, is selected for link prediction. Using the dendrogram thus obtained, for any two nodes, the number of links between the two is calculated, normalized by the total possible number of links. This value decides the probability of having a link between the two concerned nodes in the network. However there is no guarantee of accuracy in such approaches and also such methods are unsuitable for large networks due to the computational complexity of obtaining the hierarchical structures.

Another interesting approach is the *stochastic block* model [Lü and Zhou, 2011] based approach. It is one of the most general network model in which the nodes are partitioned into groups and the probability that two nodes are connected depends solely on the groups to which they belong. Let us consider a partition M where each node belongs to only one group. Say, for two groups α and β, the probabilities of two nodes within a group being connected is $P_{\alpha\alpha}$ and probabilities of two nodes in different groups is $P_{\alpha\beta}$. Also, $e_{\alpha\beta}$ is the number of edges between nodes in group α and β and $n_{\alpha\beta}$ is the number of pairs of nodes such that one node is in α and another is in β. Then the likelihood of

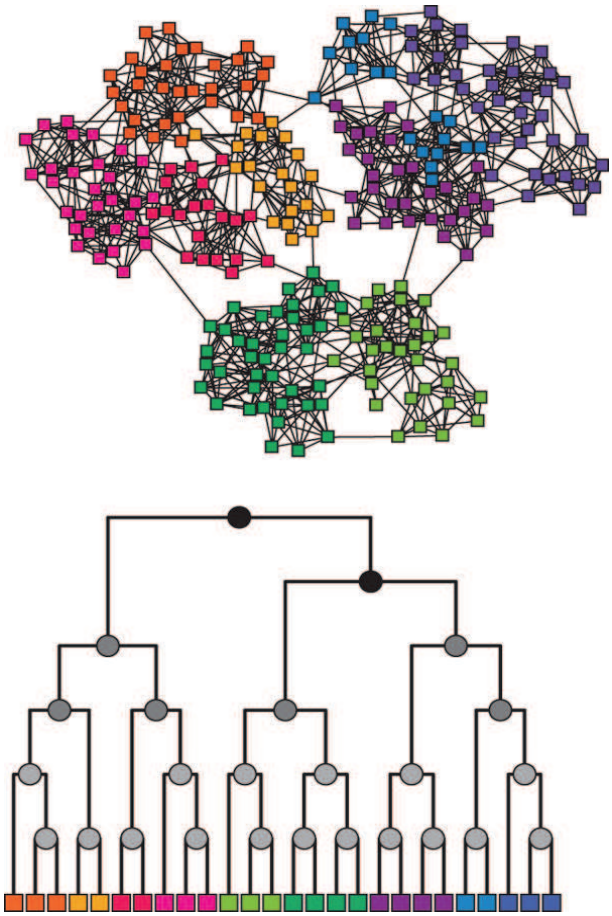


FIGURE 3.6: Hierarchical structure of a random network [Clauset et al., 2008]. Each internal node the dendrogram is associated with a probability that a pair of vertices in the left and right subtrees of that node are connected.

observed network structure will be:

$$L(A|M) = \prod_{\alpha \leq \beta} P_{\alpha\beta}^{e_{\alpha\beta}} (1 - P_{\alpha\beta})^{n_{\alpha\beta} - e_{\alpha\beta}} \tag{3.26}$$

Another important approach is that of *probabilistic relational* models which provide a way to incorporate both node and edge attributes to model a joint probability distribution of a set of nodes and the links that associate them. These kind of approaches are mostly based on either Bayesian networks considering relational links to be directed [Getoor et al., 2003] or Markov's networks that consider the links to be undirected [Taskar et al., 2003]. These models represent a joint probability distribution over the attributes of a relational dataset. They allow the property of an object (node/link) to depend, in a probabilistic manner, both on other properties of that object and on properties of related objects. A typical probabilistic relational model use three graphs: a data graph (G_D), a model graph (G_M) and an inference graph (G_I):

- A *data graph* is a graph that represents the original target network. The data objects are represented as nodes and the relationships are edges. Each node and edge are associated with a set of attributes corresponding to their type.

- A *model graph* represents the dependencies among the attributes at the level of object (node/edge) type. As mentioned earlier, an attribute of an object (node/edge) can depend probabilistically on other attributes of same object as well as attributes of other related (or similar) objects in the data graph. A data graph can be decomposed into multiple parts corresponding to each type. Using this, a joint model of dependencies among type attributes can be built. Hence a model graph has two parts: A dependent structure of all type attributes and the conditional probability distribution associated with each node in the model graph.
- An *inference graph* represents probabilistic dependencies among all variables in a single test set. It can be instantiated by a roll-out process of data graph and model graph. Each object-attribute pair in data graph gets a separate copy of corresponding conditional probability distribution from the model graph. The relations in model graphs determine the way data graph is rolled out to form the inference graph.

This probabilistic relational models are originally designed for attribute prediction problem for relational data. In case of link prediction, the existence or absence of links needs to be considered (assuming only the binary case for simplicity). B. Taskar et al. [Taskar et al., 2003] have proposed to consider a set of potential links between nodes. Each potential link is associated with a tuple of node attributes, but it may or may not actually exist. They denote this event of existence or absence of a link, using a binary attribute *Exists*, which is true if the link between the associated nodes exists and false otherwise. Then the link prediction task is reduced to the problem of predicting the existence attributes of these link objects.

3.3.3 Semi-supervised approaches

Semi-supervised learning is a type of learning that makes use of unlabeled instances along with a small amount of labeled instances. Traditional supervised learning methods require labeled data to train a model (especially in tasks of classification). Labeled instances however are often difficult, expensive, or time consuming to obtain, as they require the efforts of experienced human annotators. At the same time, unlabeled data may be relatively easy to collect, but there has been few ways to use them. Semi-supervised learning addresses this problem by using large amount of unlabeled data, together with the labeled data, to build better classifiers. Because semi-supervised learning requires less human effort and gives high accuracy, it is of great interest both in theory and in practice. We suggest readers to refer to the work of X. Zhu [Zhu, 2005] for a detailed survey on semi-supervised learning.

This type of learning has not been very well explored in the context of link prediction. To our knowledge only a few work exist, of which a prominent one is the work of H. Kashima et al. [Kashima et al., 2009]. Authors have dealt with the problem of predicting the unknown (or future) parts of a network structure from the known part of the structure. It can also be seen as the problem of completing an adjacency matrix representing the network. The proposed method is a node-information based approach and uses the concept of label propagation (which was originally developed for node classification) to predict links between pairs of nodes. Authors extend the principle of label propagation to fit it for a pair of nodes. The principle is that if two pairs of nodes are similar to

each other, they can share link pattern (absence or presence of a link) and link type. Kronecker sum and product similarity has been applied in this work to find similarities between any two triplets of the form $(node1, node2, link_{type})$. In the later part, authors have also proposed a conjugate gradient based method to deal with scalability problem.

Another interesting work in this field is the work proposed by C. Brouard et al. [Brouard et al., 2011] which is based on Output Kernel Regression. In this work, the link prediction task which has been previously represented as a binary classification task, is converted to an output kernel learning task. A target output kernel is assumed to encode similarity between nodes in a graph. The function to find these similarities is to be approximated using appropriate input features. Once the output kernel is learned using kernel tricks, a threshold can be put on the kernel values of pairs of input nodes, to predict the presence or absence of links.

Although semi-supervised methods provide an interesting alternative for link prediction, their applicability is limited to prediction of missing links. Whether these kind of approaches are equally useful for predicting new links, is an open issue. Moreover, both the above mentioned approaches use node feature information for finding similarity between nodes. This does not assure their performance in a purely topology based scenario.

3.4 Challenges in link prediction task

Link prediction in complex networks comes with some important challenges especially when the problem is dealt as a supervised classification problem. These challenges are mostly due to the large size and sparsity of data available in real world networks. We describe a few important issues here based on the list presented in the work of M. Hasan et al. [Al Hasan and Zaki, 2010] and Z. Bao et al. [Bao et al., 2013].

1. **Class skewness:** Class skewness is the problem of having imbalance in the ratio of instances belonging to different classes or class distribution in any dataset. In a typical case of supervised machine learning based classification task, normally the class ratio is balanced. It is expected to have the same probability of randomly choosing a positive or negative example. But, when it comes to link prediction, there is an extreme class imbalance owing to the fact that the number of actual new links is very small as compared to the number of possible links. In [Al Hasan and Zaki, 2010], it is stated that the number of possible links is quadratic times of the number of nodes in the network while the number of actual new links is only a very small fraction of that. Also, with the evolution of the network, the number of negative links grows quadratically while the number of positive links grows linearly [Rattigan and Jensen, 2005]. Thus, in any supervised approach for link prediction, during learning of models and its validation, number of negative examples are many times more than the number of positive examples. This makes it more difficult for an algorithm to generate a good model with a good performance on the test data. Also in presence of large class skew, the information carried by the positive examples gets diluted in the vast negative class. Moreover unlike classical classification problem in machine learning context where overall prediction accuracy is important, in link prediction, correct classifications of positive examples are more important. Another aspect is that the performance of a learning algorithm greatly depends on the variance in the model estimates. Even a low proportion

UNSUPERVISED	1) [Liben-Nowell and Kleinberg, 2007] 2) Neighborhood based metrics [Adamic and Adar, 2003; Jaccard, 1901; Liu et al., 2011; Ou et al., 2007; Tan et al., 2014] 3) Path based metrics [Chebotarev and Shamis, 1997; Fouss et al., 2006; Katz, 1953; Lichtenwalter et al., 2010; Lü and Zhou, 2011] 4) Path betweenness centrality 5) Node features based metrics [Barabasi and Albert, 1999]
SUPERVISED	1) Supervised classification based approaches [Benchettara et al., 2010b; Fire et al., 2011; Hasan et al., 2006] 2) Local probabilistic model [Wang et al., 2007] 3) Matrix based [Gao et al., 2011; Menon and Eklan, 2011] 4) Stochastic block model [Lü and Zhou, 2011] 5) Hierarchical probabilistic model [Clauset et al., 2008] 6) Probabilistic relational model [Getoor et al., 2003; Taskar et al., 2003]
SEMI-SUPERVISED	1) Link propagation [Kashima et al., 2009] 2) Output kernel regression model [Brouard et al., 2011]

DYADIC	1)[Liben-Nowell and Kleinberg, 2007] 2) Neighborhood based metrics [Adamic and Adar, 2003; Jaccard, 1901; Liu et al., 2011; Ou et al., 2007; Tan et al., 2014] 3) Node features based metrics [Barabasi and Albert, 1999] 4) Supervised classification based approaches [Benchettara et al., 2010b; Fire et al., 2011; Hasan et al., 2006]
SUBGRAPH	1) Local probabilistic model [Wang et al., 2007] 2) Stochastic block model [Lü and Zhou, 2011]
GLOBAL	1) Path based metrics [Chebotarev and Shamis, 1997; Fouss et al., 2006; Katz, 1953; Lichtenwalter et al., 2010; Lü and Zhou, 2011] 2) Path betweenness centrality 3) Matrix based [Gao et al., 2011; Menon and Eklan, 2011] 4) Hierarchical probabilistic model [Clauset et al., 2008] 5) Probabilistic relational model [Getoor et al., 2003; Taskar et al., 2003] 6) Link propagation [Kashima et al., 2009] 7) Output kernel regression model [Brouard et al., 2011]

TABLE 3.2: Summary of categorization of link prediction approaches that we have studied, based on two different dimensions

of negative instances that are similar to positive instances can cause the model to end up with a large number of false positives. A straight forward solution to the problem of class imbalance is *under-sampling* or *down sampling* of negative instances. Under sampling of the majority class is a good way to improve the sensitivity of the classifier towards minority class. M. Kubat et al. [Kubat et al., 1997] have proposed to selectively under-sample majority class while keeping all instances of majority class. N. Chawla et al. [Chawla et al., 2002] propose to use over-sampling of minority class along with under-sampling of majority class. They use the product of number of positive examples and the length of attribute vector for increasing the number of positive examples for learning. However, oversampling approaches can increase the size of dataset and also the training time.

Other approaches include making the learning process active and cost sensitive [Al Hasan and Zaki, 2010]. However, under-sampling comes with the risk of losing valuable information and so, careful selection should be made on the criteria deciding which examples are to be discarded. More details about class imbalance problem can be found in [Al Hasan and Zaki, 2010; Lichtenwalter and Chawla, 2012]. In [Lichtenwalter and Chawla, 2012], there is a detailed description about how the predictor performance changes with sampling of test data. They also provide valuable information about which performance measure is to be used for evaluating different link prediction techniques.

2. **Model calibration:** Sometimes calibrating a model is more crucial than finding right algorithm to build a classification model [Al Hasan and Zaki, 2010]. Model calibration is a process to find a function that transforms the output score of a model to label. By varying or biasing this function the ratio of false positive error and false negative errors can be controlled. This will also depend on the requirements of the network on which a link prediction model is being developed. For example in a terrorist network missing a positive link is more serious than in online social networks where recommending a negative link can be a bigger mistake.

3. **Selection of attributes or features:** In network topology based approaches, appropriate selection of attributes is very essential. They affect link prediction in two ways. First the performance of the prediction depends highly on the prediction capabilities of the attributes. And second the computational efficiency of the attributes will decide the overall computational complexity of the link prediction approach. If we have prior knowledge about the performance of the different topological predictors, then it is easy to select the best performing ones as attributes to have a good prediction result. But this is not the case in reality as the performance of different topological predictors vary with the kind of networks on which they are being used. So some methods to find the importance of these as attributes is needed. One way of dealing with this issue is to use *principal component analysis* (PCA) as in the work of Z. Bao et al. [Bao et al., 2013]. Authors propose a framework of three steps. First principal component analysis is done to determine principal components (PCs) out of all attributes. These components are statistically independent and are ranked in the decreasing order of their contribution to the variance of result. Then out of them only those m predictor variables are selected which have the highest eigenvalues and which are grouped into h clusters. From each cluster the attribute closest to the mean of the cluster is selected. In the third and final step, considering only the attributes selected in the previous step, multiple linear regression method is applied to find weights for the selected h components. Using these weights and selected features the link prediction is done.

4. **Dynamic update of models:** Complex networks are dynamic in nature. Especially social networks continuously evolve with time. Hence for any link prediction approach one of the important challenge is to deal with the temporal aspects of the network. In such networks, with time more informations may be added in the form of introduction of new nodes and links or disappearance of a few nodes and links. This information can play a crucial role to affect the prediction results. Hence in many works on link prediction this time aspect have been included [Acar et al., 2009; Benchettara et al., 2010b; Cooke, 2006; Dunlavy et al., 2011; Gao et al., 2011; Huang et al., 2008; Huang and Lin, 2008; Lahiri and Berger-Wolf, 2007; Ouzienko et al., 2010]. Dynamic update of models is needed in order to adapt the model with changes that arrive with time. This aspect is more essential when a link prediction approach is to be implemented in a real evolving network like in applications such as recommender systems in various social networks. In such cases the trade-off between complete rebuilding and updating the model should be taken into consideration [Al Hasan and Zaki, 2010]. A few work that propose such temporally adaptive models for online social networks are given in [Aggarwal et al., 2012; Song et al., 2009].

5. **Heterogeneity:** In general many of the link prediction approaches have dealt with only homogeneous networks that have same kind of nodes and links. But the real networks are in fact very diverse in nature. So in many of complex networks, link prediction task may include prediction of links between different types of nodes and also prediction of different kinds of links between same type of nodes. Considering heterogeneity in a complex network can also be helpful to improve the performance of a link prediction approach, owing to the fact that complex networks are very sparse and much more information can be added by using the linking patterns in different dimensions [Aggarwal et al., 2012; Davis et al., 2011, 2013; Eronen and Toivonen, 2012; Pujari and Kanawati, 2013; Wang and Sukthankar, 2013; Yu et al., 2012].

3.5 Motivation

In our research, we were very much interested in discovering topological approaches for link prediction because of its generic nature. While studying all these different topological approaches, we realized that none of the works try to combine the effects of different topological features using rank aggregation method. These are methods that combine rankings provided by different experts on a set of candidates and conceptually come from social choice theory. We already had an in-hand experience of working with rank aggregation methods, applying it in the context of tag recommendation on folksonomy (see appendix A). So we were quite hopeful about its applicability in the context of link prediction. Hence we developed a *supervised rank aggregation based link prediction* approach which is detailed in Chapter 4. For experimentation we used scientific collaboration networks which are a part of bibliographical networks. Bibliographical networks come with a diverse kind of information. We saw that two authors in scientific collaboration network can be linked in many different ways. For example they can be linked if they publish papers in same conferences/journals or attend same conferences. Another way of linking them is based on the references they have used in their works. All this made us think that if we can exploit this heterogeneous link information, the prediction of co-authorship links may be improved. Thus we were inspired to work on multiplex

networks. Multiplex networks are a form of heterogeneous networks where the network is represented as layers for graphs, each having same nodes but different kinds of edges. They have a simple structure with a possibility of applying existing topological measures without much difficulty. Moreover, there is not much work done in the field of link prediction in multiplex networks. So we developed a link prediction approach for multiplex network using simple topological attributes and their extended versions for a multiplex scenario. This approach has been described in Chapter 5. At the same time, another concept which caught our attention was that of existence of communities in social networks. We were highly interested to explore the utility of communities in the context of link prediction. So after studying few works on community detection approaches, we came up with our approach of using them for filtering of examples in link prediction. This sampling may provide a solution to deal with the problem of creating a better prediction model in presence of huge class imbalance in the data. This has been reported in Chapter 6.

3.6 Conclusion

In this chapter, we have presented the detailed description the analysis task of *link prediction*. We define the problem with a formal presentation. We also highlight different applications of link prediction. We then present a brief state of art of different link prediction approaches concentrating mainly on topological approaches. We discuss different ways of categorizing link prediction approaches based on various criteria. We present our two dimensional categorization of approaches. An approach can be unsupervised, semi-supervised or supervised in one axis, while it can be dyadic, community/sub-graph based or global in another. After that, we highlight some major challenges of link prediction task in real world networks especially when they involve supervised learning. We discuss in brief the existing solutions to these problems. In the category of unsupervised dyadic link prediction methods, we introduce a new concept of topological measure namely path betweenness centrality, that can be used for finding linking probability between two unlinked nodes in a complex network. There is detailed account of this new path based topological feature in Appendix B. This concept is a very naive attempt to use shortest path in a different way for link prediction and has not been explored enough presently. We present our motivation for the research work on new algorithms for topological link prediction. Focusing only on the topology of a network, we are also interested to make use of heterogeneous link information and community information to enhance the prediction results of a supervised model. All these are presented in detail in subsequent chapters.

Chapter 4

Applying Rank Aggregation to Link Prediction

4.1 Introduction

After having an overview of the work done to solve the problem of link prediction, we came to a conclusion that none of the previous work attempt to combine the prediction power of individual topological measures by applying *computational social choice* algorithms or simple voting rules. These methods are a part of social choice theory and were mostly applied to political and election related problems [Black et al., 1998; de Borda, 1781; Young and Levenglick, 1978]. A detailed description about voting methods and their history can be found in the work of C. Dwork et al. [Dwork et al., 2001], D. Black et al. [Black et al., 1998], and H.P. Young et al. [Young and Levenglick, 1978].

In computer science these methods have been studied mostly in the context of information meta-search in web [Aslam and Montague, 2001; Dwork et al., 2001], multiple search, similarity search [Fagin et al., 2003] etc. and are popularly termed as preference aggregation or rank aggregation methods [Chevaleyre et al., 2007; Dwork et al., 2001]. *Rank aggregation* can be defined as a process of combining a set of ranked lists of candidates to get a single aggregated list that has least possible disagreements with all the voters or experts who provide these lists.

These techniques were designed to ensure fairness among voters while combining their rankings and hence all voters are given equal weights. Expressing the link prediction problem in terms of a vote is straightforward: candidates are examples (pairs of unlinked nodes), while voters or experts are topological measures computed for these pairs of unlinked nodes. Then we have a voting problem with quite huge set of candidates and rather a reduced set of voters (contrary to the social choice set up where number of voters is huge and number of candidates is small). These settings are also similar to those encountered when considering the problem of ranking documents in a meta-search engines where voting schemes has also been applied with success [Aslam and Montague, 2001; Dwork et al., 2001; Montague and Aslam, 2002].

This chapter starts with problem description in section 4.2. Section 4.3 lists the classical rank aggregation methods. This section mostly summarizes concepts from social choice theory. Section 4.4 describes few of the works that apply weighted or supervised rank

aggregation for diverse purposes. Section 4.5 presents our proposed method for supervised rank aggregation and and in section 4.6 we describe how we use supervised rank aggregation for link prediction. The experimental details have been provided in section 4.7.

4.2 Rank aggregation problem

In this section, we provide a brief description of *ranked list aggregation/rank aggregation* process and the existing methods.

4.2.1 Rank aggregation

Rank aggregation refers to a process of combining a number of ranked lists of candidates to get a single list and with least possible disagreement with all the voters who provided these lists. The lists may have same or different elements. The process of rank aggregation differs from the process of simple list aggregation by the fact that in rank aggregation methods, the order or ranks of candidates in the input lists are also taken into consideration. Formally if we have a set of ranked lists $L = [L_1, L_2, L_3, ..., L_m]$ provided by m *experts* or *voters* and each containing n candidates, the goal is to find a ranked list $L_{aggegate}$ of same n candidates, which has least possible disagreement with the rankings of candidates provided in the lists in L. The rank of a candidate x in a list L_i is given by $rank(x, L_i)$

Depending on the candidates, the input lists can be categorized as *Full* lists, *Partial* lists or *Disjoint* lists. *Full* lists are those which contain exactly the same candidates but with a different ordering, *partial* lists may have some of the candidates in common but not all and *disjoint* lists have completely different elements. For example we consider four lists:

$$L_1 = [A, B, C, D] \qquad L_2 = [B, D, A, C]$$

$$L_3 = [A, B, C, D, E] \qquad L_4 = [E, F, G, H]$$

In this example, L_1 and L_2 are full lists, L_1 and L_3 are partial lists, and L_1 and L_4 are disjoint lists. The aggregation of full lists, partial lists and disjoint lists are three different challenges which call for three different streams of research in the field of computational social choice theory. In this work we deal only with the aggregation of full lists.

In rank aggregation, distance metrics are used to find the disagreement between two ranked lists. Two well-known distance measures are *Spearman Footrule* distance and *Kendall Tau* distance. For two ranked lists L_1 and L_2 of n candidates, the two metrics are defined as

- Spearman Footrule distance: This computes the distance between two ranked lists by computing the sum of differences in ranks of each candidate. Formally, it is given by

$$F(L_1, L_2) = \sum_{i \in n} \mid rank(x_i, L_1) - rank(x_i, L_2) \mid \tag{4.1}$$

- Kendall Tau distance: This counts the number of pairs of elements that have opposite rankings in the two input lists i.e. it calculates the pairwise disagreements.

  $$K(L_1, L_2) = \mid (x_i, x_j)\ s.t.\ rank(x_i, L_1) < rank(x_j, L_1)\ and\ rank(x_i, L_2) > rank(x_j, L_2) \mid \tag{4.2}$$

  where x_i and x_j are any two candidates present in both input lists L_1 and L_2, but not necessarily in the same order.

Spearman footrule distance has a computational complexity of $O(n)$ where as Kendall Tau distance has a computational complexity of $O(n \log n)$. In the example given before, if we select lists $L_1 = [A, B, C, D]$ and $L_2 = [B, D, A, C]$, the Spearman footrule distance will be $F(L_1, L_2) = 6$ and Kendall Tau distance will be $K(L_1, L_2) = 3$.

A normalized value of any of the two distances can be obtained by dividing it by number of voters(or experts) or number of input lists.

4.2.2 Weighted rank aggregation

Before describing the approaches that use supervised rank aggregation, we give a definition for weighted and supervised rank aggregation.

Weighted rank aggregation refers to the same process of combining ranked lists but giving different importance to the experts. So here, each expert has a weight associated with it. In case, these weights are learned during a training process in a supervised manner the method is called *supervised rank aggregation*. Formally for a set of ranked lists $L = [L_1, L_2, L_3, ..., L_m]$ containing n candidates and associated weights $W = [w_1, w_2, w^-3, ..., w_m]$, the goal of weighted rank aggregation is to find a ranked list $L_{aggegate}$ containing same n candidates, which has least possible disagreement with the rankings of candidates provided in the lists in L. In supervised rank aggregation these weights in W can be learned.

4.3 Rank aggregation methods

Rank aggregation methods can be broadly categorized into two types: *score-based* and *order-based*. Score-based aggregation methods use score information from voters while order-based methods use only the rank information [Liu et al., 2007]. Score based methods use a scheme of weighting or giving scores to the candidates in order to determine their overall order of preference. Order-based methods on the other hand use binary comparison to ascertain whether there is a candidate that can defeat all other candidates by a simple majority.

Another important concept that we need to discuss before moving towards various classical rank aggregation methods is the *Condorcet principle*. *Condorcet principle* as proposed by Marquis de Condorcet [Condorcet, 1785], says that if there exists some candidate that defeats every other candidate in a pairwise simple majority voting, then that candidate should be selected as a winner. Such a winner is known as *Condorcet winner*. In some circumstances it is possible to have no Condorcet winner because there is no candidate who is preferred by voters to all other candidates. Such a situation is known as *Condorcet paradox*. Each rank aggregation methods complying with Condorcet principle can

have their own ways of dealing with such situations but in any case they assure to rank a Condorcet winner on the top whenever it exists. For example in the work of Duncan Black [Black et al., 1998], the method proposed chooses a Condorcet winner if it exists otherwise uses Borda's count method (described below). There are also possibilities of having an ordinary tie when two or more candidates have tie with each other but defeat all other candidates. Ties can be broken by random choice or some other concepts like which of the winners has the most first choice vote etc. The way Condorcet paradox is handled can be an important point of difference between different rank aggregation methods complying with Condorcet principle (also called *Condorcet methods* sometimes). Condorcet methods fit within two categories:

- Two-method systems, which use a separate method to handle cases in which there is no Condorcet winner.
- One-method systems, which use a single method that, without any special handling, always identifies a winner to be the Condorcet winner.

An extended Condorcet criterion as proposed by M. Truchon says that if there is a partition $\{T, U\}$ of the set of n candidates $\{1, 2, ..., n\}$ such that for any $x \in T$ and any $y \in U$ the majority prefers x to y, then x must be ranked above y in the final aggregation [Truchon, 1998].

We now describe a few of the standard methods for rank aggregation.

Borda's rank aggregation: Borda's method is a truly positional method as it is based on the absolute positioning of the ranked candidates rather than their relative rankings. A Borda score is calculated for each candidate in the lists and based on this score, the elements are ranked in a aggregated list. For a set of full lists $L = [L_1, L_2, L_3, ..., L_m]$, the Borda's score for a candidate x in a list L_i is given by:

$$B_{L_i}(x) = \{count(y) \ \mid \ rank(y, L_i) < rank(x, L_i) \ \& \ y \in L_i\} \tag{4.3}$$

The total Borda's score for a candidate is given as:

$$B(x) = \sum_{i=1}^{m} B_{L_i}(x) \tag{4.4}$$

Borda's method is mostly applicable to full lists and is not very suitable for partial lists and it does not comply with the Condorcet principle. Its main advantage is its linear computational complexity of $O(nm)$.

Kemeny's optimal aggregation: Kemeny optimal aggregation [**?**] uses Kendall Tau distance to find the optimal aggregation. The first step is to find an initial aggregation of input lists using any standard method. The second step is to find all possible permutations of candidates in the initial aggregation. For each permutation, a score is then computed which is equal to the sum of distances between this permutation and the input lists. The permutation having the lowest score is considered as optimal solution. For example, for a collection of input ranked lists $\tau_1, \tau_2, \tau_3, ..., \tau_m$ and an aggregation π , the score is given by:

$$SK(\pi, \tau_1, \tau_2, \tau_3, ..., \tau_m) = \sum_{i=1}^{m} K(\pi, \tau_i) \tag{4.5}$$

where $K(\pi, \tau_i)$ is the Kendall Tau distance between a permutation π and an input list τ_i.

The specialty of Kemeny optimal aggregation is that it complies with *Condorcet principle* which is not the case with positional methods like Borda's algorithm.

In spite of all advantages, the major limitation of Kemeny optimal aggregation is that it is computationally hard to implement. It is a NP-hard problem even for four ranked lists [Dwork et al., 2001]. So while looking for an alternative solution that gives similar kind of aggregation but is computationally feasible, we are led to another approach named *local Kemenization* [Dwork et al., 2001]. A full list π is locally Kemeny optimal aggregation of partial lists $\tau_1, \tau_2, \tau_3, ..., \tau_m$, if there is no full list π' that can be obtained from π by performing transposition of a single pair of adjacent elements and for which

$$SK(\pi', \tau_1, \tau_2, \tau_3, ..., \tau_m) < SK(\pi, \tau_1, \tau_2, \tau_3, ..., \tau_m)$$

In other words, it is impossible to reduce the total distance of an aggregation by flipping any adjacent pair of elements in the aggregation (which is not equivalent to saying that no flipping of any two elements can decrease the distance). Every Kemeny optimal aggregation is locally Kemeny optimal but the converse may not be true. Local Kemenization in fact allows us to have an approximate optimal aggregation.

A simple example illustrating the difference between Borda's count method and Kemeny optimal method is given in figure 4.1. There are five lists L_1, L_2, L_3, L_4, L_5 containing four color dots. Each one presenting a ranking based on preference of some experts or some criteria. For Borda's method a score is computed for each color dot based on its absolute position in each of the five lists. Final aggregation is found by ordering the dots from low to high Borda's score. Where as, in Kemeny optimal aggregation pairwise comparison of ranks is made. Taking two color dots at a time, the number of times first dot is ranked above the second in the lists is counted and the final aggregation is found based on this comparison. We can see that the final ranking found by the two methods are different.

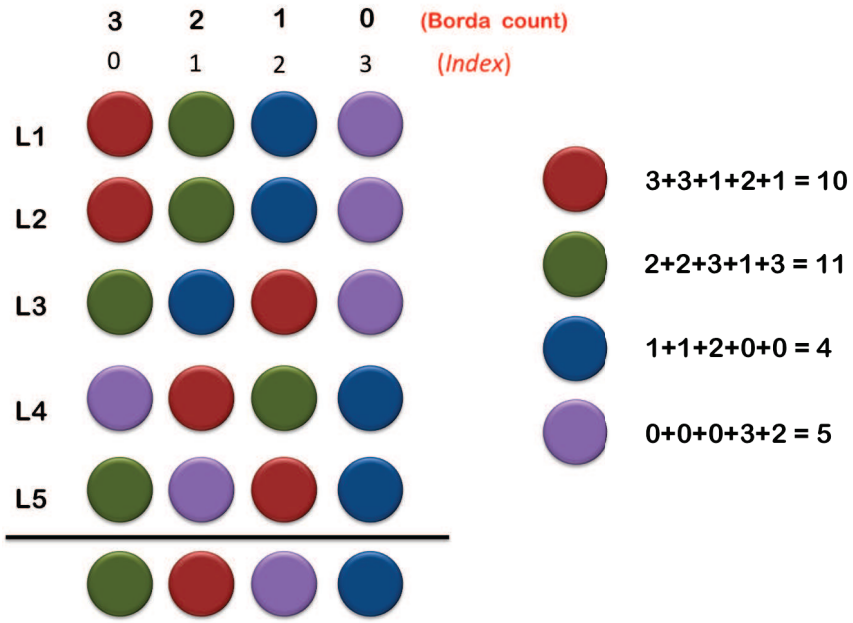
Median rank aggregation: The third method is Median rank aggregation (*MedRank*) [Fagin et al., 2003] which is based on a ranking heuristic that sorts all candidates based on the median of their ranks in the lists provided by a certain number of voters. That means it aggregates a set of complete ranked lists by using median rank for each candidate. This method can produce footrule optimal aggregations which are within a constant bound of Kemeny optimal aggregation. They satisfy extended Condorcet criterion and may be computationally more efficient than Kemeny optimal aggregations. If we have m ranked lists $[L_1, L_2, L_3, ..., L_m]$ with n candidates in each, a score for any candidate x to have a rank r is computed as the number of lists in which x has a rank r.

$$score(x, r) = count(L_i) \; where \; rank(x, L_i) = r$$

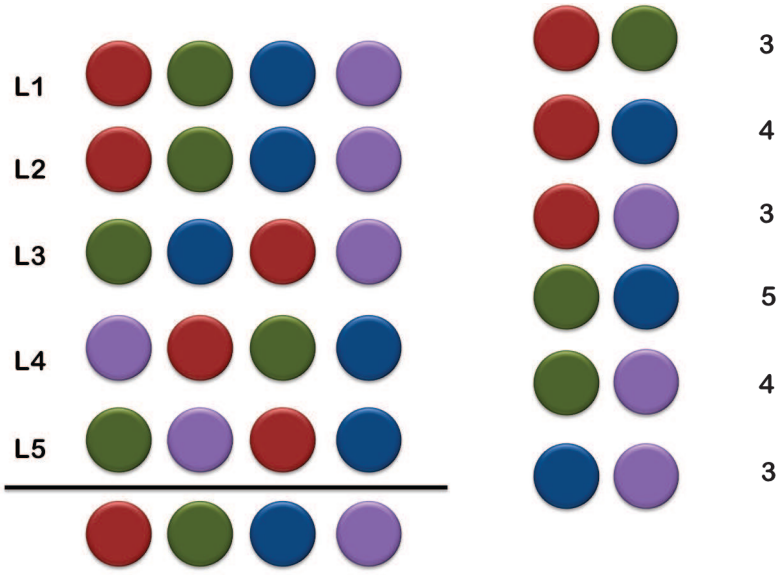
The *MedRank* score for a candidate x is

$$M(x) = \sum_{r \in (1,n)} score(x, r)$$

The first candidate with a score greater than some threshold θ gets rank 1, the second such item gets rank 2 and so on. The ties are randomly broken. A standard value of



a) Aggregation found by Borda's method



b) Aggregation found by Kemeny optimal method

FIGURE 4.1: An example to show Borda and Kemeny optimal aggregation

$\theta = \frac{m}{2}$. So a candidate must appear in at least half the lists to get a rank in aggregate list, in case of partial or top-k list aggregation.

Markov's chain based rank aggregation: Markov's chain methods represent the candidates of ranked lists as states (nodes of graph) and the transitional probabilities from one state to other is defined by the relative rankings of the candidates in different input ranked lists. The stationary distribution of the Markov Chain is utilized to rank the candidates. C. Dwork et al [Dwork et al., 2001] proposed four methods (denoted as MC1, MC2, MC3, and MC4) to construct the transition probability matrix of the Markov Chain. This method is suitable for aggregation of full as well as partial lists but does not guarantee a most optimized solution [Dwork et al., 2001; Sculley, 2007].

Other work: Another work, proposed in [Besson and Robardet, 2007] is based on the Condorcet voting count principle. Authors try to preserve most of the individual pairwise preferences while minimizing the removal of a set of pairwise preferences to erase all cycles, when the individual rankings are represented as weighted directed graphs. This turns the rankings into partial orders but as close as possible to the total order.

In [Sculley, 2007] an attempt has been made to address the problem of aggregating ranked list of candidates with defined similarity. Author makes use of candidate similarity in order to enhance the performance of the standard methods for rank aggregation. They re-define the distance measures and the well known rank aggregation methods adding a factor that quantifies the similarity between the candidates in various ranked lists. They show that introducing similarity between candidates greatly enhances the performance of rank aggregation methods especially in presence of noisy, incomplete or even disjoint data.

After studying all these rank aggregation methods, we come to a conclusion that may be Kemeny optimal aggregation will best serve our purpose of having an efficient method for link prediction due to its capability of considering both positive and negative preference on candidates by the majority. *Positive preference* means the choice of an expert to give a candidate a higher rank. On the other hand if the expert does not want a candidate to be ranked at higher position rather prefers to give it a lower rank then it will be a *negative preference*. We opt to use the concept of local Kemenization in order to avoid the issues of having very high computational complexity and to develop an approximately optimal aggregation to be used for link prediction. We are also interested to see the utility of a positional method, i.e. Borda's count method for the same, knowing that it comes with an advantage of low computational complexity. That can be an advantage if it performs well in our context.

4.4 Related work

Looking into the work based on rank aggregation techniques, we can say that not much have been explored when it comes to application of rank aggregation to link prediction. Moreover, most of the other related works apply unsupervised rank aggregation algorithms, giving equal weight to all voters or experts.

One work is weighted majority algorithm proposed in [Littlestone and Warmuth, 1989] where the authors have proposed to use weights for predictors (voters), all having equal weights in the beginning. There is a master predictor which makes the final prediction

based on the class which corresponds to maximum total weights of predictors. If the final prediction is wrong then weights of all predictors who disagreed with that label, is increased by a factor β such that $0 \leq \beta < 1$ and thus reducing the effect of unworthy predictors at each iteration. This approach has a limitation that the performance of the master predictor can be at most equal to the best performing predictor. On the contrary, the use of rank aggregation can provide even better prediction at times. This may be due the fact that, in these algorithms, the *"likes"* of majority of the predictors is given higher preference. At the same time, the *"dislikes"* are given least preference. So these algorithms are much more spam/noise resistant.

Another work on weighted rank aggregation is Borda Fuse proposed by J.A. Aslam et al. [Aslam and Montague, 2001] which can be viewed as weighted Borda count for meta-search. Specifically, different experts who provide rankings on candidates are assigned different weights, while the weights are trained separately by using labeled training data. For example, the weights can be calculated based on the MAP (Mean Average Precision) scores of the experts. Experimental results show that Borda Fuse indeed improves upon normal Borda Count. The problem with Borda Fuse is that the weights of the ranked lists are calculated independently and by using heuristics. It is also not clear whether the same idea can be applied to other methods [Liu et al., 2007].

A significant work on supervised rank aggregation has been done in [Liu et al., 2007] where authors propose supervised aggregation by Markov chain to enhance the ranking result on meta-searches. Authors argue that to have an improved accuracy of rank aggregation it is better to use a supervised learning approach in which an order based aggregation function is trained within the optimization framework of labeled data. Hence they propose supervised learning to perform task with improved accuracy. Learning is formalized as an optimization task which minimizes the disagreement between ranking result and the labeled data. They further transform optimization of Markov chain into that of a semi-definite programming to improve computational efficiency. However, it has been shown that local Kemenization improves on Markov chain-based approaches [Dwork et al., 2001].

A very recent work by K. Subbian et al. [Subbian and Melville, 2011] proposes use of supervised rank aggregation to find influential nodes by posing the problem as a predictive task. Authors compare different measures of influence like degree centrality, PageRank etc. on their ability to accurately predict which users in Twitter network will be virally re-tweeted in near future. Authors have proposed their own supervised Kemeny aggregation method based on quick sort which represents a variation of local Kemeny aggregation [Dwork et al., 2001] with approximation. In the algorithm quick sort is done on the candidates by using majority based comparisons.

This work is very close to what we have done in terms of supervised Kemeny aggregation, however their domain of application is finding influential nodes where as ours is finding potential node pairs which may have a link in the future. The work is concentrated around individual nodes whereas in our work, we study the topological features of a node pair or a potential link. For the part of supervised rank aggregation we propose our method based on Merge sort algorithm. The reason why we use merge sort is that it is seemingly more stable than quick sort. Stability of a sorting algorithm is important when two candidates have equal importance. A stable sorting algorithm is the one which never affects the relative order of two candidates who are equal in ranks. This feature may be important when we have ties. Thus as said in [Dwork et al., 2001], the advantage

of using merge sort is that the issue of inconsistent answers never arises and thereby simplifies the execution of algorithms. Further one may say that a natural option for combining the influences of different topological measures for link prediction can be the use of supervised machine learning classifiers which can be trained to predict the target links. But if the individual measures produce an ordering of potential candidates (which is also our case) and not just a point-wise score then rank aggregation methods seem to be a better choice. Moreover a method that complies with extended Condorcet criterion is highly preferred in our context because it can eliminate the possibility of inferior candidates being included in the final ranking thereby affecting greatly the result of prediction task in a positive way.

Another aspect is to look at the problem of learning to rank. In this kind of works, the goal is to find a suitable way to learn a ranking on the potential candidate node pairs, by using various topological and content based features. In a work presented recently in [Tabourier et al., 2014] a new method for link prediction based on rank learning has been proposed. Authors propose a method named "RankMerging" based on a sliding window concept, to aggregate the ranks provided on unconnected node pairs, by different topological classifiers. They use this rank learning method to predict links between users in a telephone calls network. They compare their method with well known Borda's rank aggregation method and a few machine learning algorithms like decision tree, AdaBoost, nearest neighbours. Another work proposed in [Freno et al., 2011] formalizes link prediction problem from the flexible perspective of preference learning. The goal is to learn a preference score between any two nodes. The model uses neural network and an objective function that can be optimized by stochastic gradient descent. The limitation of this work is the need for node content.

These works on learning to rank are conceptually interesting but are a bit away from our point of interest which is not to develop a new way of ranking, but rather is to find a better and robust ranking by aggregating various ordered lists provided by different methods (in this case those are topological measures). It can also be interesting to consider the outcome of these rank learning methods and aggregate them using our proposed framework. But for the moment we are not going to deal with these issues. It can be left as one of the perspectives.

4.5 Supervised rank aggregation

The existing methods for rank aggregation described in the previous section, usually give equal weights to all experts who provide the input rankings. But sometimes, there is possibility that these experts have different importance in identifying the correct order of elements. These facts motivate us to think that assigning a weight to each expert may enhance the aggregation results significantly. We thus propose to use weighted Borda's method (proposed as Borda's Fuse in [Aslam and Montague, 2001]) and a new approach for weighted local Kemeny optimal method. We call them supervised Borda and supervised local Kemeny as we learn the weights that are to be used. Generation of weights is described in next section.

Supervised Borda: Weights can be introduced into Borda's method in the following way. While computing the Borda score for each element, the rank of the element will be multiplied with the weight of the expert who provides this rank. Suppose,

$(w_1, w_2, ..., w_n)$ are the weights for n experts (and thus for the ranked lists provided by them) , the Borda score for individual element can be obtained as follows:

$$B(x) = \sum_{t=1}^{n} w_i * B_{L_i}(x) \tag{4.6}$$

The main advantage of such an method is its simple linear time computational complexity of $O(n.m)$, n being the number of ranked lists and m is the number of candidates. But the disadvantage of this method is that it doesn't not comply with the extended Condorcet criterion.

Supervised local Kemeny aggregation: Our supervised local Kemeny aggregation makes use of a comparison matrix M to compare the pairwise preference between all ranked items. For each pair of items (x, y) in the ranked lists, we compute a score using the fact whether x is preferred over y in the individual rankings and weights corresponding to experts providing those rankings. Formally, for n experts providing rankings $[\tau_1, \tau_2, \ldots, \tau_n]$ for m items

$$score(x, y) = \sum_{i=1}^{n} (w_i * Pref_i(x, y)) \tag{4.7}$$

where $Pref_i(x, y) = \begin{cases} 0 & \text{if } \tau_i(x) > \tau_i(y) \\ 1 & \text{if } \tau_i(x) < \tau_i(y) \end{cases}$.

Same score is also calculated for (y, x). If this score of (x, y) is more than 50% of the sum of all the expert weights $w_T = \sum_{i=1}^{n} w_i$, then we consider that x is preferred over y by most of the experts and this ranking should be preserved in the final aggregation also. Hence we insert $M(x, y) = true$ and $M(y, x) = false$. The final aggregation is found starting from selecting any of the input ranked lists as initial ranking and using merge sort algorithm, swapping the items only when their reverse preference is true in comparison matrix M. Algorithm 1 describes our proposed approach for finding supervised local Kemeny aggregation.

The computational complexity of merge sort algorithm is $O(m. \log m)$. But the computation of the matrix using n ranked lists of m candidates has a complexity $O(n.m^2)$, which is a bit space and time consuming when the number of candidates in ranked lists is more bigger. So we came up with another way to avoid creation of the $m \times m$ matrix and make the computation more easier. Instead of having a comparison matrix, we compute a score and decide on weighted preference on the spot during the sorting process. So in the modified algorithm, we apply merge sort on an initial aggregation. Each time a comparison is to be made between two candidates x and y, a score for (x, y) is computed using equation 4.7. As explained earlier, if the score is more than 50% of w_T, x has a higher weighted preference over y and thus, no swapping is done. Otherwise the candidates are to be swapped. All this is presented formally in algorithm 2. In this way we avoid unnecessary computation of scores for candidate node pairs which are not dealt with during the sorting process. So, the use of merge sort allows this algorithm to have a reduced computational complexity of $O(n.m \log m)$.

We illustrate the concepts by means of the same example of color dots in figure 4.2. There are four colors: Red (R), Green (G), Blue (B) and Purple (P). Supervised Borda score for R is : $B(R) = 3*0+3*1+1*2+2*3+1*4 = 15$. Similarly weighted Borda's score were

Algorithm 1 Supervised local Kemeny aggregation

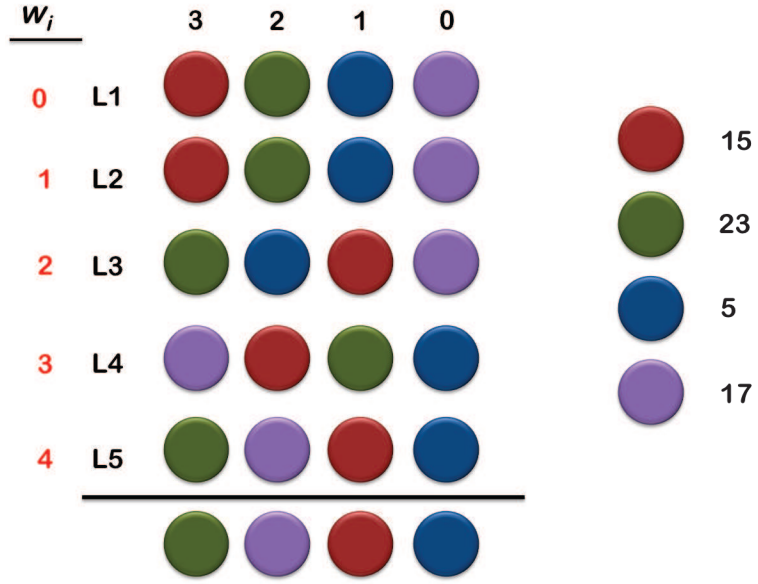
Input: $T = [\tau_1, \tau_2, \ldots, \tau_n]$ where $\tau_i = [e_1, e_2, \ldots, e_m]$ for n experts and m elements
$W = [w_1, w_2, \ldots, w_r]$ where w_i is the weight for expert i and $w_T = \sum_{i=1}^{r} w_i$
$\mu = \tau_1$ where μ can be considered as initial aggregation
Output: An aggregate ranked list

Initialize an empty matrix M
for element $x = e_1$ to e_{m-1} **do**
  for element $y = e_1$ to e_m **do**
    $score = 0$
    for $\tau_i \in T$ **do**
      $$Pref_i(x, y) = \begin{cases} 0 & \text{if } rank(x, \tau_i) < rank(y, \tau_i) \\ 1 & \text{if } rank(x, \tau_i) > rank(y, \tau_i) \end{cases}$$
      $score = score + (w_i * Pref_i(x, y))$
    end for
    if $score > 0.5 * w_T$ **then**
      $M_{xy} \Leftarrow true$
      $M_{yx} \Leftarrow false$
    else
      $M_{xy} \Leftarrow false$
      $M_{yx} \Leftarrow true$
    end if
  end for
end for
Merge sort μ using M.
Return μ

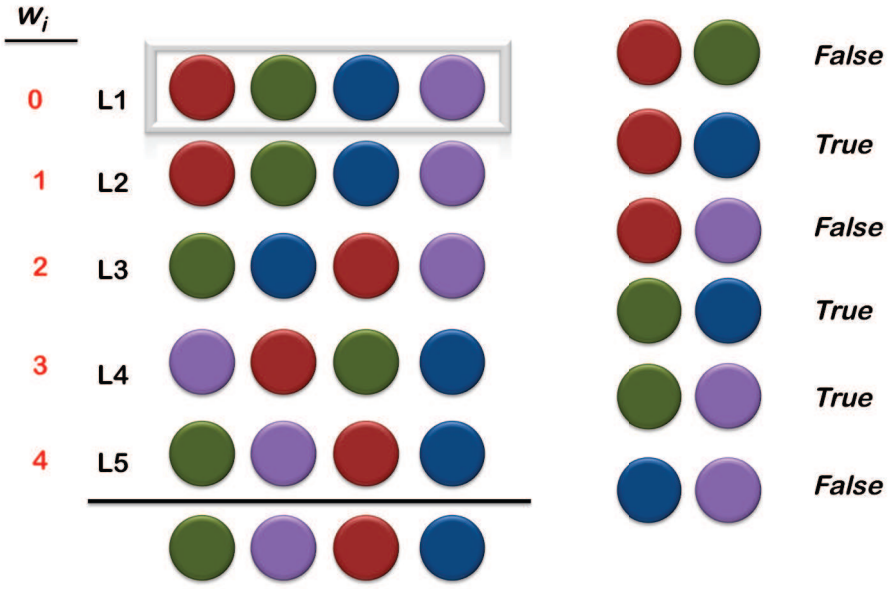
computed for all other color dots. For supervised local Kemeny aggregation, the first ranked list of color dots is taken as the initial aggregation. Matrix M is computed for all candidates as shown in the algorithm **??**. The values in M are either $True$ or $False$ based on a score computed for two candidates. For example, lets consider red and green dots. Now, $score(R, G) = 1*0+1*1+0*2+1*3+0*4 = 4$ and total weight $w_T = 10$. As $score(R, G) < 0.5 * w_T$, $M_{RG} = False$ and the same time $M_{GR} = True$. Take another example of green and blue dots for which $score(GB) = 1*0+1*1+0*2+1*3+1*4 = 8$. As $score(GB) > 0.5*w_T$, $M_{GB} = True$ and the same time $M_{BG} = False$. After completing M for all pairs of candidates, merge sort is applied on the initial ranking, making a swap only if $M_{x,y} = False$.

4.6 Applying supervised rank aggregation to link prediction

The first step of our link prediction approach is to generate training and test examples as described in section 3.2. The examples have a set of topological attributes associated to them. Each attribute of an example, when considered individually, provides some unique information about it. The training examples are ranked based on the attribute values. So, for each attribute we will get a ranked list of all examples. The second step is to compute a weight for each topological attribute. Here an assumption is made that when



a) Aggregation found by supervised Borda's method



b) Aggregation found by supervised local Kemeny method

FIGURE 4.2: An example showing computation of supervised Borda and supervised local Kemeny aggregation

Algorithm 2 Supervised local Kemeny aggregation (Without using Matrix)

Input: $T = [\tau_1, \tau_2, \ldots, \tau_n]$ where $\tau_i = [e_1, e_2, \ldots, e_m]$ for n experts and m elements
$W = [w_1, w_2, \ldots, w_r]$ where w_i is the weight for expert i and $w_T = \sum_{i=1}^{r} w_i$
$\mu = \tau_1$ where μ can be considered as initial aggregation
Output: An aggregate ranked list

Merge sort μ
for each Comparison (x, y) **do**
  $score = 0$
  for $\tau_i \in T$ **do**
    $$Pref_i(x, y) = \begin{cases} 0 & \text{if } rank(x, \tau_i) < rank(y, \tau_i) \\ 1 & \text{if } rank(x, \tau_i) > rank(y, \tau_i) \end{cases}$$
    $score = score + (w_i * Pref_i(x, y))$
  end for
  if $score > 0.5 * w_T$ **then**
    $x \succ y = true$
  else
    $x \succ y = false$
  end if
  if $x \succ y == false$ **then**
    swap(x,y)
  end if
end for
Return μ

we rank the examples according to their attribute values, the positive examples should be ranked on the top. So, considering only the top k ranked examples, we compute the performance of each attribute. This performance is based on either maximization of identification of positive examples (measured in terms of *precision*) in top k positions or minimization of identification of negative examples (measured in terms of *false positive rate*) in top k positions. A combination of both can also be used but for the time being we consider the two separately. Based on the individual performances, a weight is assigned to each attribute. The weight computation is detailed in next sub-section.

For validation, we use examples obtained from the validation graph characterized by same attributes and rank all examples based on their attribute values. So for n different attributes, we have n different ranked lists of the test examples. These ranked lists are then merged using a *supervised rank aggregation* method and the *weights of the attributes* obtained during learning process. The top k ranked examples in the aggregation are taken to be the predicted positive examples. Using this predicted list, we calculate the performance of our approach. k in this case is equal to the number of positive examples in the validation graph.

4.6.1 Weight computation

We propose to compute expert's (topological measures) weights based on their capability to identify positive elements in top k positions of their rankings. Weights associated to

applied topological measures are computed based on the following criteria :

- **Maximization of positive precision:** Based on maximization of identification of positive examples in top k positions of the ranked list provided by a topological attribute, the weight is calculated as

$$w_i = n * Precision_i \tag{4.8}$$

  where n is the total number of attributes and $Precision_i$ is the *precision* of attribute a_i based on identification of positive examples. To remind, precision is defined as the fraction of retrieved instances that are positive.

- **Minimization of false positive rate:** By minimizing the identification of negative examples in top k positions we get a weight as below

$$w_i = n * (1 - FPR_i) \tag{4.9}$$

  where n is the total number of attributes and FPR_i is the $false\ positive\ rate$ of attribute a_i based on identification of negative examples. False positive rate is defined as the fraction of negative instances that are predicted as positive.

In both cases, we are multiplying it with a constant value n (which is totally optional) in order to enhance the numeric value of weight which at times can be very less and close to zero. Also the weights are normalized by dividing them by the total weights of all topological attributes. Other criteria for weight computation can also be applied. For example in [Subbian and Melville, 2011], weights are computed based on AUC of node features on training data.

4.7 Experiment

We evaluate our approach using data obtained from DBLP [1] databases. Our network consists of authors as nodes and they are linked if they have co-published at least one paper during the observed period of time. The data corresponds to year between 1970-1979. We create three graphs out of that. In order to do a justified comparison within the set of unlinked node pairs, that will be considered for prediction, we prefer to use only the largest connected component of the graphs for this experiment. The problem when we consider the whole graph is that, if we have two node pairs (x, y) and (u, v) belonging to a connected component of large size and small size respectively. Suppose the attribute we are considering is common neighbors (CN). There are much chances that $CN(x, y) > CN(u, v)$ and during ranking by value (x, y) will be ranked higher than (u, v). Thus, (x, y) has a greater chance of being selected as a predicted link as compared to (u, v) even if (u, v) comes out to be the real new link. Hence, to avoid all these complications and to have a fair comparison between candidate node pairs, we use the largest connected components as graphs in this experiment and we use the term “graph" or “network" for the same.

Following the procedure described previously in Chapter 3, we generate examples from each of the graphs. Table 4.1 provides information about the training or test graphs while

[1] http://www.dblp.org

table 4.2 summarizes information about the examples generated. Figures 4.3, 4.4 and 4.5 show the visualization of the three concerned co-authorship graphs corresponding to three different points of time. These visualizations are done by using *Gelphi* [2] [Bastian et al., 2009], which is a reasonably simple and interesting tool. (See appendix E for more visualizations of other co-authorship graphs).

Years	$\|V\|$	$\|E\|$	*Density*	*Avg(degree)*	*Avg(Cc)*	*Diameter*	*Avg(Pathlength)*
1970-1973	91	116	0.028	2.549	0.333	14	6.114
1972-1975	221	319	0.013	2.887	0.462	16	7.203
1974-1977	323	451	0.009	2.793	0.404	18	7.504

TABLE 4.1: DBLP Co-authorship graph

Learn/Test	**Label**	**# Positive**	**# Negative**
1970-1973	1974-1975	16	1810
1972-1975	1976-1977	49	12141
1974-1977	1978-1979	93	26223

TABLE 4.2: Examples from co-authorship graph

Dataset	**Learning year**	**Test year**	K
Dataset 1	1970-1973	1972-1975	49
Dataset 2	1972-1975	1974-1977	93

TABLE 4.3: Datasets for experiment

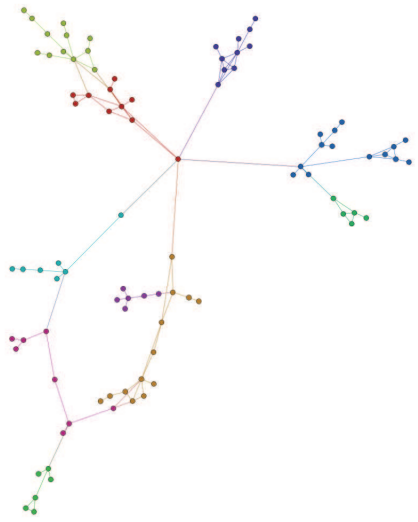


FIGURE 4.3: Co-authorship network for year 1970-1973

We have applied our approach to the datasets as described in table 4.3. K is a parameter used in rank aggregation to decide the top k predictions and is equal to the number of

[2] http://gephi.github.io/

FIGURE 4.4: Co-authorship network for year 1972-1975

FIGURE 4.5: Co-authorship network for year 1974-1977

actual positive links in the test data. For rank aggregation, we have used supervised Borda and supervised Kemeny methods. We wanted to experiment with a score based rank aggregation method as well as an order based one. For score based method Borda was a obvious choice as it very well represents an absolute position based rank aggregation approach. Our second choice was Kemeny aggregation approach as it produces an optimal final aggregation, keeping in tact the preferences of all experts as much as possible. It has also been successfully applied in aggregating search results in a meta-search engine [Dwork et al., 2001] in which its ability to deal with spams has also been shown. So, we chose to use an approximate version of Kemeny optimal aggregation.

We compare our approach with link prediction approaches using supervised machine learning algorithms like Decision tree, Naive Bayes and k-Nearest neighbors algorithm. These methods are simple to implement and represent three different concepts of classification. We name our approaches as Supervised Borda 1 and Supervised Borda 2 based on how the attribute weights are computed. 1 represents weights computed based on maximization of positive precision and 2 represents weights being computed based on minimization of false positive rates. We will follow the same convention to represent supervised Kemeny. The supervised machine learning algorithms are implemented using Orange [3] which is a Python based data analysis and visualization software. We selected the following topological attributes to characterize examples (i.e node pairs): Number of common neighbors (CN), Jaccard coefficient (JC), Preferential attachment (PA) [Huang et al., 2005], Adamic Adar coefficient (AA) [Adamic et al., 2003], Resource allocation (RA) [Zhou et al., 2009], Shortest path length (SPL), Path betweenness centrality (PBC), Truncated Katz (TKatz) and Neighbor's clustering coefficient (NCF).

We compute the performance of our rank aggregation based link prediction methods and link prediction based on supervised machine learning algorithms. We use the three algorithms for supervised machine learning. We also compute the same using ensemble learning with decision tree. We have restricted the number of predictions made by machine learning algorithm to K, the parameter that is selected for rank aggregation based methods too. This is done in order to have a justified comparison between the two kinds of approaches. Figure 4.6 and 4.7 present the results we obtained in terms of precision and AUC for all methods. AUC is computed using the formula given in section 3.2.1 as proposed in [Lü and Zhou, 2011], but the difference is that instead of exact score we compare the ranks of negative and positive examples. So actually, we compute the probability of finding a positive example ranked above a negative example in the list of prediction which is the top-K ranked lists provided by all link prediction algorithms. Also we are unable to take into account the equal ranks between negative and positive examples, as we are not treating ties for the moment. Ties are broken randomly whenever they appear in the score and dealing with ties during ranking can be added to future updates of our work.

While our method based on Borda and supervised Borda failed to provide any substantial results (due to which we have not listed them here), our approximate Kemeny and supervised Kemeny based methods outperform all the supervised machine learning and ensemble methods for both datasets in terms of precision. This shows the validity of our approach. Although in terms of AUC the result is slightly different, with decision tree giving the best AUC for dataset 1. But still the precision for the same is not very high. Both the type of ensemble learning based on decision tree perform badly as compared to rank aggregation and supervised rank aggregation based methods. The low values

[3] http://orange.biolab.si/

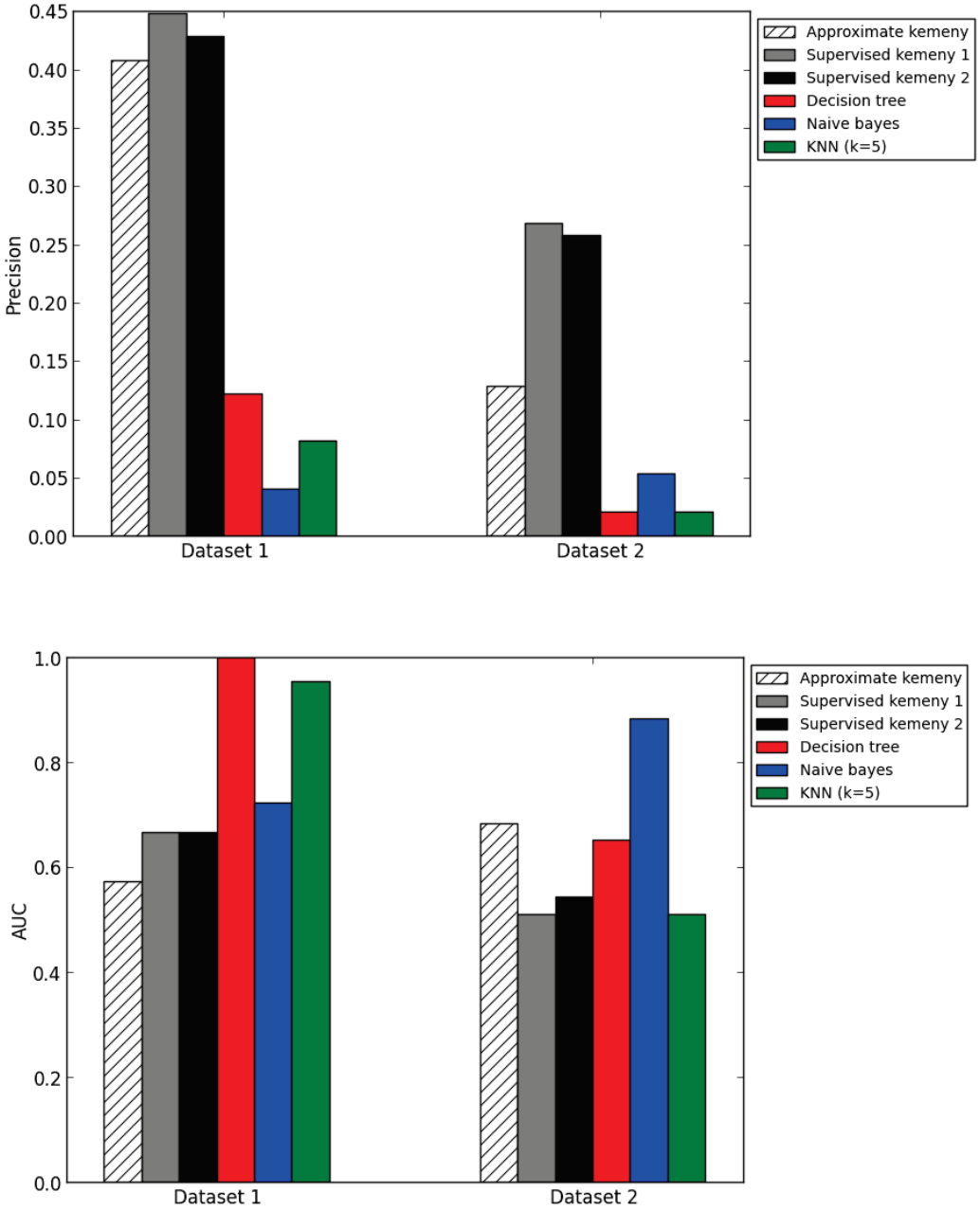


FIGURE 4.6: Results on the two datasets compared with Supervised machine learning

of AUC can be attributed to the fact that we have used raw data without any sort of pre-treatment or refining. To ease the process of supervised Kemeny aggregation further, we make a selection of best performing attributes. In fact we discard all attributes that have a zero weight during learning. That means these attributes failed to identify any of the positive examples in the top k positions during learning. So, the rankings provided by such topological attributes do not seem to be very useful for being used further during prediction of links in the test set. This step can be significant to select the best serving attributes for the prediction task and they also help the execution process.

Precision-recall curves are more indicative of the difference between the performance of algorithms in presence of a class imbalance, having a large number of negative examples as compared to positive examples. So we decided to use them, in order to compare

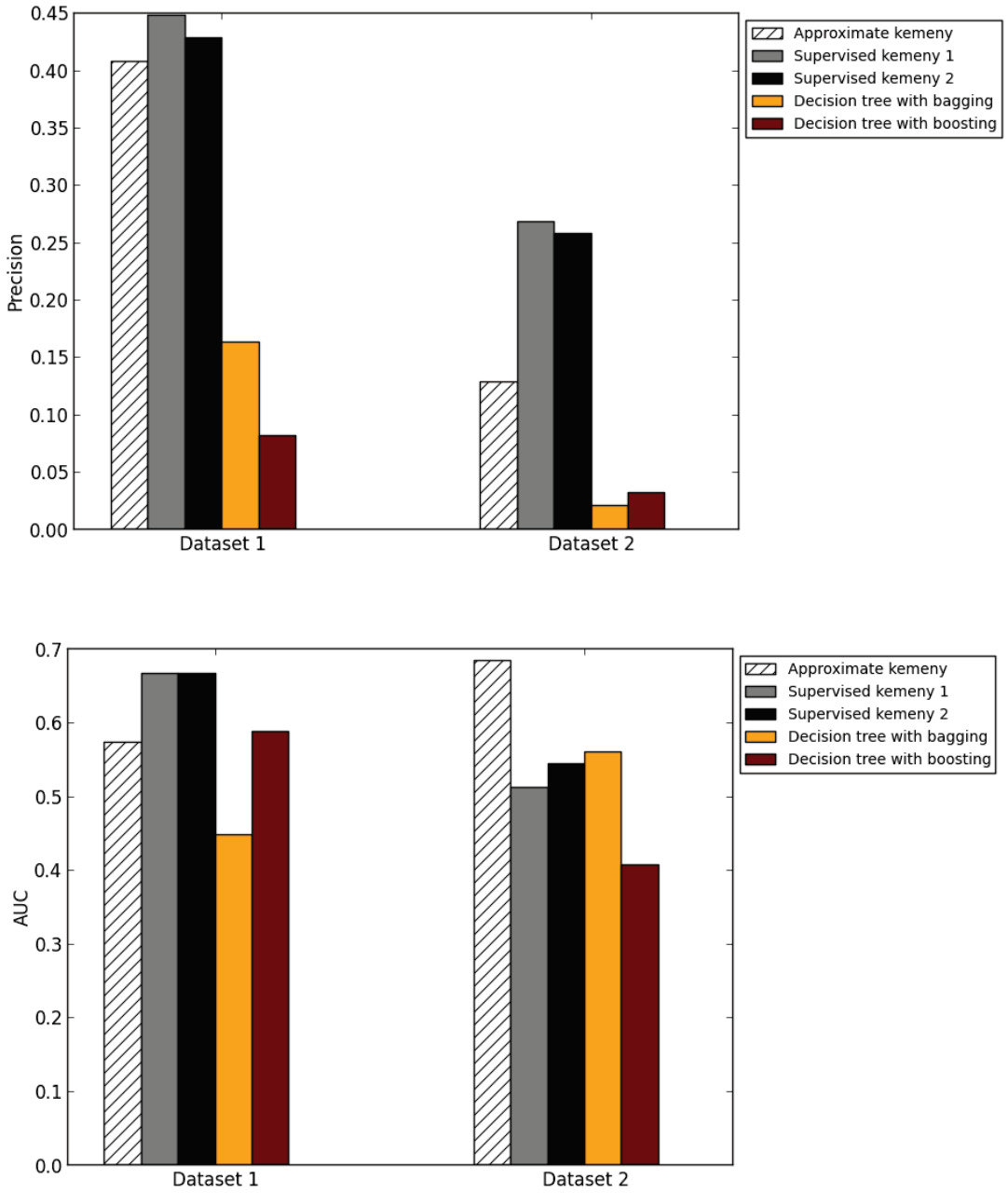
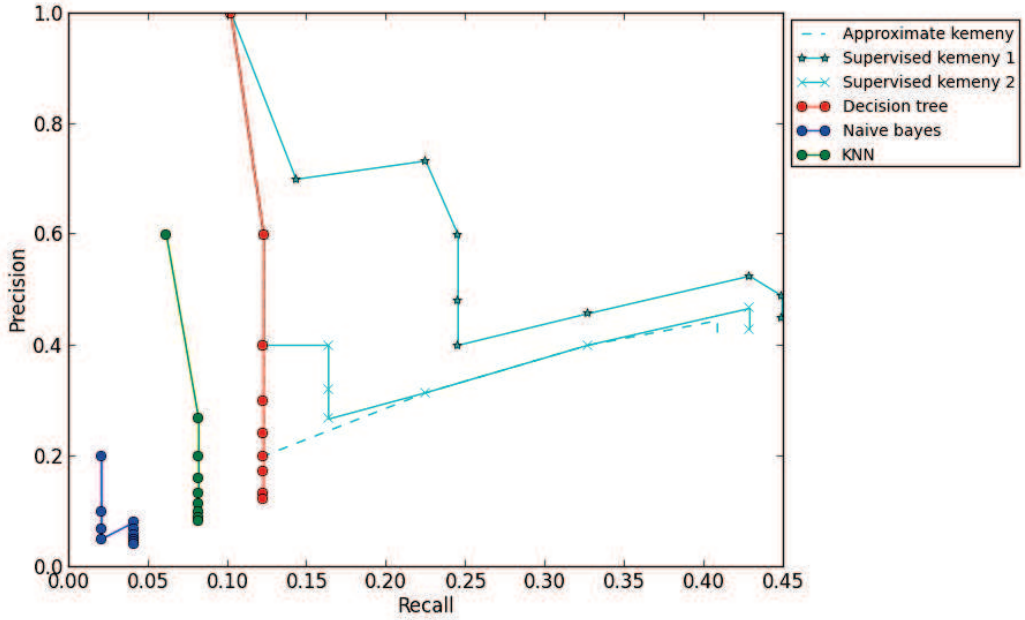


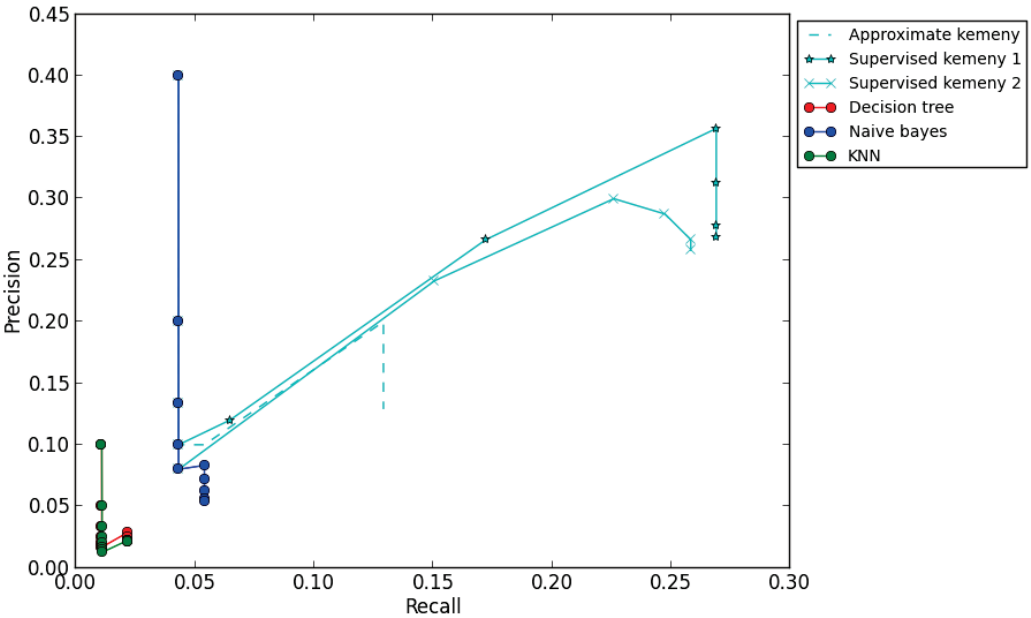
FIGURE 4.7: Results on the two datasets compared with Ensemble learning

different algorithms. Looking at the precision-recall curves in figures 4.8 and 4.9 created by varying the value of K, we can clearly see the difference between various algorithms. For dataset 1, K varies between 5 and 49 (the actual number of positive links in the test set) with an epoch of 5. For dataset 2, K varies from 10 to 93 with an epoch 10. The two figures show that, for both datasets, rank aggregation based methods perform better than the supervised machine learning based methods as their corresponding curves lie above covering greater area than those representing supervised machine learning methods. Also it is evident that our method based on supervised Kemeny aggregation, where weights are computed based on precision outperform all other methods.

Although it is still early to say that rank aggregation based methods are better performing than the other approaches of link prediction, the preliminary results do show that

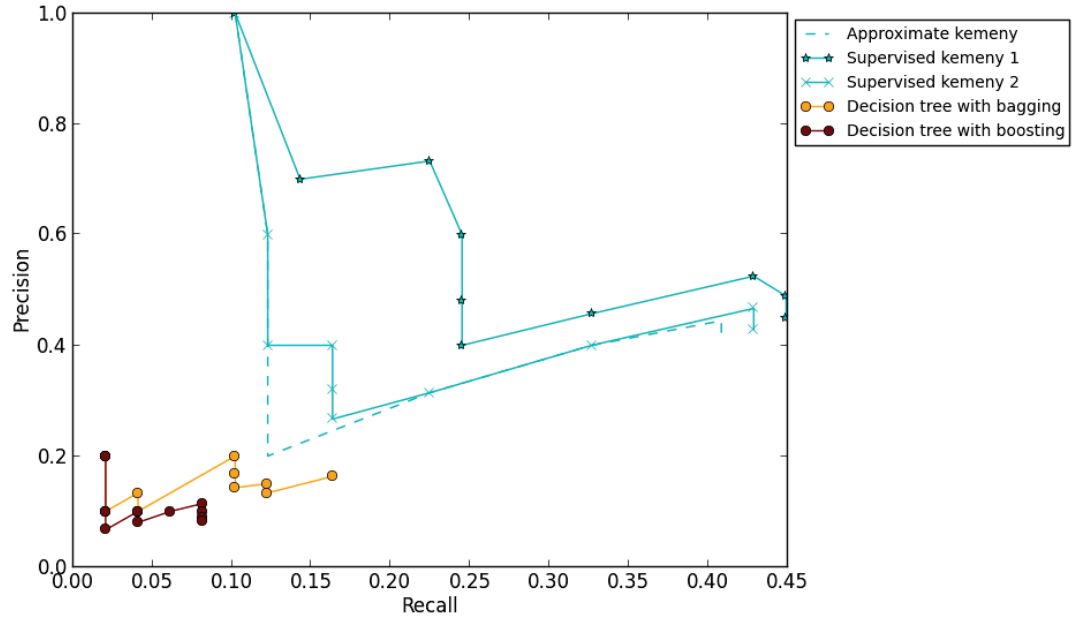


a) Dataset 1, K varies between 5 and 45 with an epoch of 5. The last point corresponds to $K = 49$, the actual number of positive links in the test set.

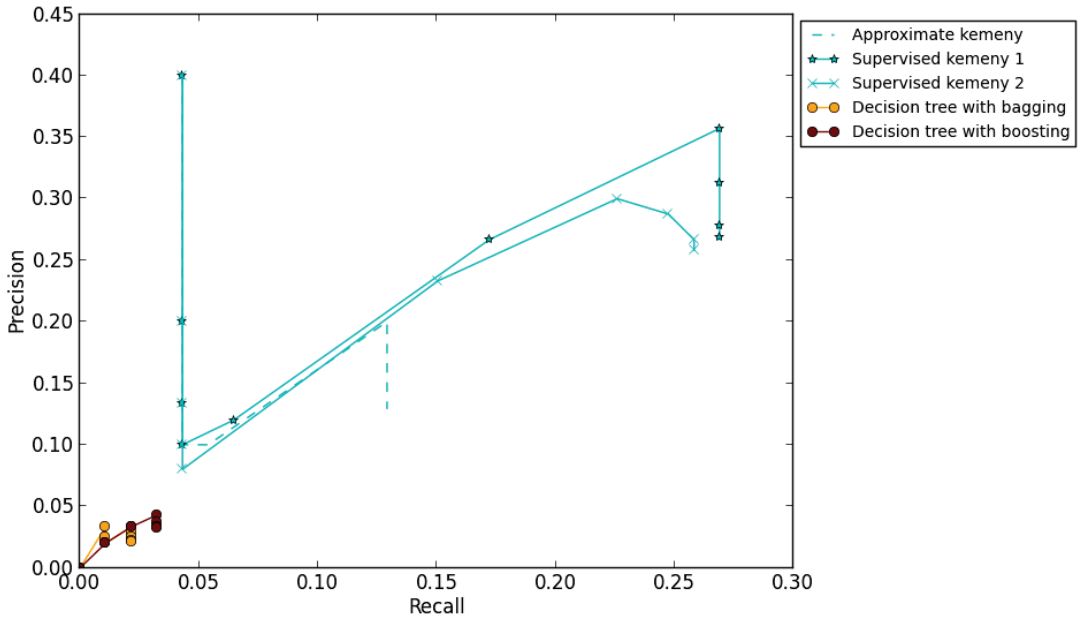


b) Dataset 2, K varies from 10 to 90 with an epoch 10. The last point corresponds to $K = 93$, the actual number of positive links in the test set.

FIGURE 4.8: Precision-Recall curves for the two datasets compared with Supervised machine learning



a) Dataset 1,K varies between 5 and 49 with an epoch of 5. The last point corresponds to $K = 49$, the actual number of positive links in the test set.



b) Dataset 2, K varies from 10 to 90 with an epoch 10. The last point corresponds to $K = 93$, the actual number of positive links in the test set.

FIGURE 4.9: Precision-Recall curves for the two datasets compared with Ensemble learning

rank aggregation especially with Kemeny method indeed adds some useful information which can enhance the result of prediction task. This is quite encouraging for us to continue this work further. Still, the fact remains that rank aggregation methods especially Kemeny based methods have a high computational complexity but some relaxation can be provided by using approximation of optimality.

4.8 Conclusion

In this chapter we have described our proposed link prediction approach based on supervised rank aggregation method. Starting with a detailed description of rank aggregation methods, we present the aggregation task in the context of link prediction. There is a brief account of important related work involving rank aggregation and link prediction. We first define our own algorithm for a weighted rank aggregation based on local Kemeny optimal approach [Dwork et al., 2001]. Then we describe our proposed approach of using it for link prediction. In this, we first learn weights associated with a set of topological features that characterize unlinked node pairs, and then use these weights with our supervised rank aggregation method to predict links in a co-authorship network.

We compared our method with baseline supervised machine learning approaches by experimenting on DBLP data. We found that the use of rank aggregation algorithms improves the performance of link prediction in terms of precision as compared to that of Decision Tree, Naive Bayes and Knn model. We applied two standard rank aggregation methods namely Borda's method and our version of Kemeny based method. While Borda failed to give any concrete result, we observed that Kemeny based methods outperform all others in terms of precision and have a comparative result in terms of AUC. The failure of Borda method for the prediction task can be explained based on the fact that while ranking the candidates, it surely considers the positive preferences of experts but it fails to take into account their negative preferences on the candidates. But this is better captured by the Kemeny based aggregation methods which comply with Condorcet principle. We think this is the reason why approximate Kemeny and supervised Kemeny methods give a better result. Other ways of weight computation and choice of initial rankings for supervised Kemeny approach can also be experimented to see if they are able to give better results. Also, more experiments can be done applying weighted forms of other rank aggregation methods especially median rank method to find its applicability in context of link prediction.

Chapter 5

Link Prediction in Multiplex Networks

5.1 Introduction

Complex networks are often heterogeneous in nature. That means real networks may have different types of nodes and also different types of links. They can be broadly divided into two categories: Multi-mode networks and Multiplex/Multi-layer networks (Figure. 5.1). Multi-mode networks are distinguished by the presence of different types of nodes that may have homogeneous or heterogeneous links. On the other hand, multiplex networks essentially have different kinds of links between same types of nodes. They can be represented as a set of simple networks (layers), each having same type of nodes but different types of links. Multi-layer networks are more general forms of networks where each layer can share some common nodes but not necessarily all. Nodes in all layers belong to the same type but the links in different layers have different types. Multi-dimensional networks also have only one type of nodes and different types of links but

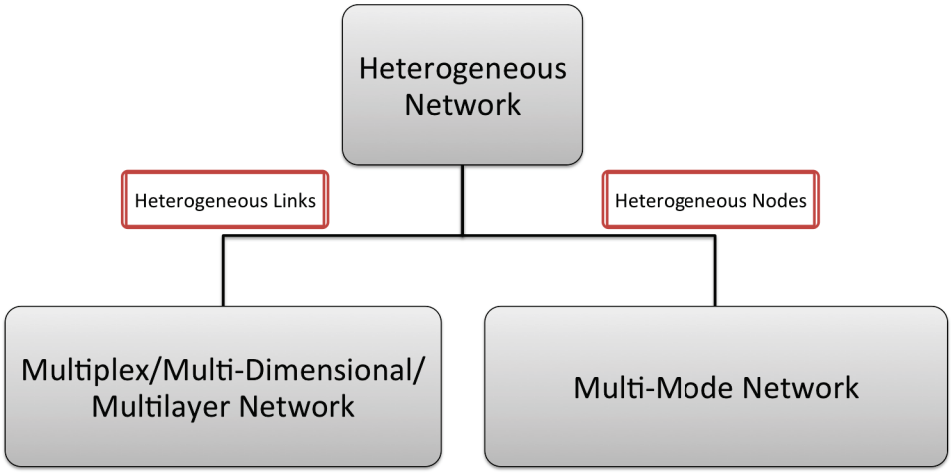


FIGURE 5.1: Heterogeneous networks and branches

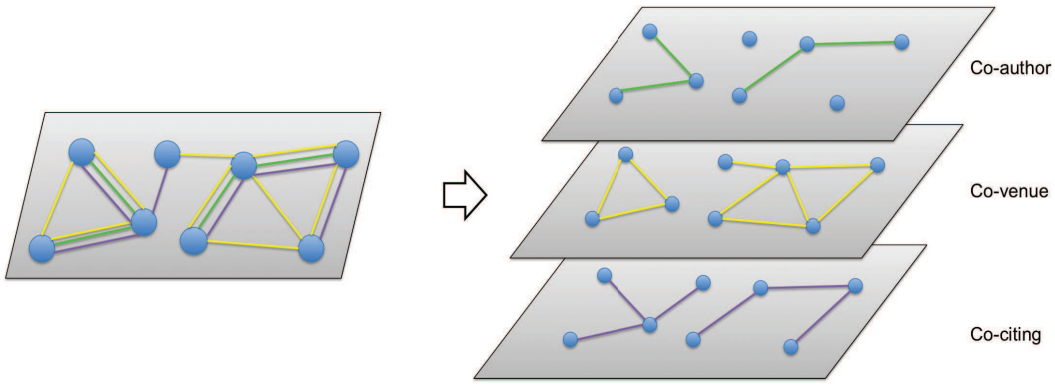


FIGURE 5.2: Multiplex structure in a scientific collaboration network for authors

they do not have a layered structure. This term has been used mostly for networks having multiple links (each having a different type) between same types of nodes.

A common example of multiplex network can be derived from scientific collaboration networks (see figure 2.10) in the form of author-author network. These networks are composed of nodes representing researchers or authors of scientific papers who can be linked if:

- they have co-published/co-authored some articles or
- they have published their articles in the same conferences or
- their domain of research are the same or
- the titles/abstract/content of their articles share some common terms etc.

In figure 5.2, it is shown how an author network can be represented by multiple layers, each having same nodes but different types of links or edges.

All the work that we saw in previous chapters, address the problem of link prediction in only simple networks having homogeneous links. Our main interest here is to extract more information from the heterogeneous properties of the network and to use them to enhance the result of link prediction.

In this chapter we explain how prediction of links can be done in a multiplex setting and how prediction performances can be enhanced using multiplex information. To our knowledge, not much have been explored to add multiplex information for the task of link prediction. Although there are a few recent work proposing methods for prediction of links in heterogeneous networks, networks which have different types of nodes as well as edges [Davis et al., 2013; Wang and Sukthankar, 2013; Yizhou Sun et al., 2011]. There have also been few work on extending simple structural features like degree, path etc. to the context of multiplex networks [Battiston et al., 2013; Berlingerio et al., 2011a] but none have attempted to use them for link prediction. The related work on heterogeneous and multiplex networks is presented in section 5.2. We propose a new approach for exploring the multiplex relations to predict links in one of the layers using metrics based on observation of links on other layers. We apply this to predict future collaboration (co-authorship links) among authors. The applied approach is a supervised-machine learning approach where we attempt to learn a model for link formation based on a set

of topological attributes describing both positive and negative examples. While such a concept has been successfully applied in the context on simple networks [Benchettara et al., 2010a; Hasan et al., 2006], different options can be applied to extend it to the multiplex network context. One option is to compute topological attributes in each layer of the multiplex. Another one is to compute directly new multiplex-based attributes quantifying the multiplex nature of dyads (potential links). Both approaches will be discussed in the section 5.3. Section 5.4 presents the experimental details on multiplex networks derived from DBLP data in the context of co-authorship link prediction.

5.2 Related work

There is not much work done in the field of link prediction in multiplex networks. Although, there has been quite a few recent work on link prediction in heterogeneous networks and on finding new ways to extend the traditional topological properties like degree, distance, centrality etc.

5.2.1 Link prediction in heterogeneous network

In the work of Yizhou Sun and al. [Yizhou Sun et al., 2011], authors propose a method called PathPredict, for co-authorship prediction in heterogeneous bibliographical network. Heterogeneous bibliographical network used in the work contains different types of objects as node such as authors, venues, papers and topics with different types of relationships between them like “write" or “written by" ($write^{-1}$) between authors and papers and “cite" or “cited by" ($cite^{-1}$) between papers etc. So the network is a directed graph with a type information on nodes and links. PathPredict is a meta path-based relationship prediction model. A meta path is a path defined on the network schema, where nodes are object types and edges are relations between object types. So a meta-path is composed of different types of links available in the network. For example, a co-authorship link between two authors can be represented by a meta path consisting of two links between the authors and the paper written together by the authors. That means, if A represents author nodes and P is the paper or article nodes, then the meta path for co-authorship link is $[A \xrightarrow{write} P \xrightarrow{write^{-1}} A]$. Similarly, a co-citation relation between two authors has the meta path $[A \xrightarrow{write} P \xrightarrow{cite} P \xrightarrow{write^{-1}} A]$. PathPredict has two components: a meta path based topological feature definition and a logistic regression based supervised prediction model. In first step, the topological measures like common neighbors, Jaccard’s coefficient, Katz’s measure are extended to use meta-paths instead of simple paths. In the second step a supervised prediction model is created using logistic regression. For a given of pair of authors and for a particular kind of relation (i), a set of meta-path based topological measures (x_i) are computed and then a prediction model is built to learn coefficients associated with each of these measures. Hence, the training set consists of $< x_i, y_i >$, where y_i (positive /negative or 1/0) is the label corresponding to a vector of meta-path based feature x_i characterizing a node pair. The probability of getting two nodes linked is modeled as follows:

$$p_i = \frac{e^{x_i \beta}}{e^{x_i \beta} + 1} \tag{5.1}$$

where β is coefficients associated with each feature in x_i including a constant 1. Maximum likelihood estimation is used to find the coefficient weights associated with the constant and each topological measure used. The goal is to maximize the likelihood of observing all relations in the training data. This is done by using the following equation

$$L = \prod_i p_i^{y_i}(1 - p_i)^{(1-y_i)} \tag{5.2}$$

The authors show that the meta path based topological measures can improve the co-authorship prediction accuracy compared with the baselines that use only homogeneous links. The results are presented in terms of overall accuracy and AUC using cross validation and test set. Although the results of PathPredict are good in numbers, there are some reservations on its performance in real scenario especially because it increases the number of paths available between two un-linked nodes. Also the authors have no explanations about its performance in the presence of huge class imbalance (which happens in real networks) as the only result they present is on a test or training set where the number of positive examples are equal to the number of negative examples.

In the work of D. Davis and al. [Davis et al., 2013], the authors propose a probability based weighted extension of Adamic/Adar measure for heterogeneous information network showing the benefits of using diverse link information especially when the homogeneous links are very sparse. Their method requires an appropriate weighting scheme for different edge type combinations. The weights are found based on counting of occurrence of each unique 3-node substructure in the network. They have presented both unsupervised and supervised prediction schemes for various types of links present in the network, concluding that supervised models are a better choice.

A different kind of approach is the one proposed by Xi Wang and al. [Wang and Sukthankar, 2013] where the authors have explored heterogeneity within co-authorship links. For example, co-authorship links can be distinguished with “affiliation" that represents the type of conferences where two authors have published a paper together. Inside a co-authorship network, they try to find classes of links in terms of communities. The nodes can simultaneously belong to multiple overlapping communities. So a node is supposed to have different kind of links based on the different communities to which the connected links belong to. The proposed approach, Link Prediction using Social Features (LPSF), is a link prediction framework which weights the network using a similarity function based on features extracted from patterns of interactions of nodes in various communities found in the network. First edge clustering is done to find clusters or communities. Then a social feature set is constructed for each node based on how many links they have in each community. That means if n communities are found, a social feature set of a node v is a vector of size n with values equal to number of links x connecting v to the corresponding communities. These feature sets are then used to find similarity between nodes which is further used as link weight in the same network. After having a weighted network, weighted versions of traditional topological measures can be computed and then unsupervised and supervised machine learning based prediction tests can be performed on unconnected node pairs of the network.

Other such work on heterogeneous networks includes Biomine [Eronen and Toivonen, 2012], a system proposed to integrate several biological databases into a graph with different types of edges which are weighted based on their type, reliability and informativeness. The predictions are based on a proximity measure computed on the integrated

graph. Considering different proximity measures, a parameter optimization procedure is done, weighting different types of edges in order to optimize the prediction accuracy.The method is tested on disease-gene networks. Work of [Yu et al., 2012] deals with prediction of citation relationship between papers. They propose a citation probability learning model based on a meta-path based prediction model (as in [Yizhou Sun et al., 2011]) on a topic discriminative search space. The difference here is that the links to be predicted are directed.

5.2.2 Work on multiplex networks

In case of multiplex networks where the heterogeneity of links in a network are represented as multiple layers, to our knowledge, not much work has been done to deal with the problem of link prediction. However there have been few important works in recent times on how to extend the standard topological measures in a multiplex scenario. One such work is proposed by F. Battiston and al. [Battiston et al., 2013], where the authors propose a general framework to describe the multi-layer networks with either weighted or unweighted links. They propose a set of measures to characterize the multiplexity of the networks. These include extension of a number of structural properties like degree distribution, node clustering, shortest paths, betweenness and closeness centralities. They also focus on the quantification

- participation of nodes to the structure of network in each layer
- importance of each node for overall efficiency of the network in terms of node reachability and triadic closure.

The most important aspect of their studies is that they have given much importance to the percentage of edge overlap and interdependence between the layers of network. They also introduce the concept of *Entropy* in the context of multiplex networks which takes into account the distribution of a topological feature in various layers of the network. For example, they present entropy of multiplex degree for a node which is a suitable, quantity to describe the distribution of degree within the layers of a network. Entropy for node degree is zero if all links of the node are in a single layer and is maximum when the links are uniformly distributed over the layers. They also present another similar quantity called *multiplex participation coefficient*, which quantifies the participation of a node in different multiplex layers. This quantity has been previously used to quantify the participation of a node in different communities [Guimera and Nunes Amaral, 2005; Guimerà et al., 2005].

In another work proposed by M. Berlingero and al. [Berlingerio et al., 2011a], the authors present the heterogeneity of the network in the form of multi-dimensional network which is similar to multiplex networks. The sole difference is that the authors present all types of links between nodes in the same graph but with edge labels to represent the various types. So we have here a multi-graph with more than one links between nodes, each with a label to show it's type (dimension). Referring to fig 5.2, we can say that such a multi-dimensional network can be split into various layers of multiplex network or in other words layers of a multiplex network can be combined to form a multi-dimensional network. In this work on multi-dimensional network, the authors propose a way to extend definition of degree of a node. They also propose some new multi-dimensional measures

namely *number of neighbors*; *dimension relevance* that tries to capture the importance of one dimension over others for the connectivity of a node; *dimension connectivity* which studies the percentage of nodes or edges contained in a specific dimension with respect to the total number of nodes and edges present; and lastly *d-correlation* that gives an idea about how redundant the different dimensions are for the existence of nodes and edges.

The work of G. Bianconi points towards a statistical mechanics formulation of multiplex networks in terms of *Entropy* and*Overlap*, the concepts also used in [Battiston et al., 2013]. They introduce the concept of correlated multiplex ensembles where the existence of a link in one layer is correlated to the existence of the link in other layers. A network ensemble can be defined as a set of networks that satisfy a given number of structural constraints, i.e., degree sequence, community structure etc. They also give a clear distinction between uncorrelated and correlated multiplex ensembles. In an uncorrelated multiplex network the probability of existence of a link in one layer does not depend on the presence of links in other layers where as in a correlated one this dependence exists.

A similar work of V. Nicosia and al. [Nicosia et al., 2013] presents a framework for modeling evolution of multiplex networks. The work of A. Halu and al. [Halu et al., 2013] presents a biased random walk based method to compute multiplex *PageRank*. They define four different versions of multiplex PageRank and show how the importance of a node in one layer can affect the importance the node can get in other layers. Another work presented by E. Cozzo and al. [Cozzo et al., 2013] proposes to generalize the concept of clustering coefficients for multiplex networks. In the work of M. Magnani and al. [Magnani and Rossi, 2013] we can find a new definition of geodesic distance that includes the different types of connections. They use the concept of Pareto efficiency to define a new distance called *Pareto distance* and they say tha geodesic distance is a special case of Pareto distance in case of a single layered network. In another work by M. Magnani and al. [Magnani et al., 2013], the authors attempt to find hidden motifs traversing and correlating different layers. They propose to extend betweenness centrality for multiplex networks taking into consideration paths crossing several different layers. In work of M. De Dominico et al. [De Domenico et al., 2013b], authors present ways of extending various existing centrality measures to be used in interconnected multiplex networks. They also show how the ranking of nodes done by computing centralities on multiplex networks is different from the ranking obtained by applying them on a weighted monoplex network obtained by aggregating the multiplex layers of the network. Another work by same authors [De Domenico et al., 2013a], presents the concept of random walks in multiplex networks where they present a new type of walk that can exist only in multiplex networks.

Another axis of research where multiplex networks have been used very recently is that of community detection. Some of the approaches transform the multiplex networks into simple networks and apply the existing methods of community detection [Berlingerio et al., 2011b; Suthers et al., 2013]. This is done by aggregating the layers to form a simple weighted network where different types of links strengthen an actor's (node) connection. The weights on edges can be computed based on different criteria to add the multiplex information on one graph. The different criteria can be binary weights, frequency-based weights, node similarity-based weights and a linear combination. A different proposal of transforming a multiplex network into a uniform hypergraph and then apply community detection algorithms have been made in the work of [Kivelä et al., 2013]. In another category of work, researchers try to extend the existing community detection algorithms

to deal directly with multiplex networks [Lambiotte, 2010]. Such approaches address the problem of simultaneous exploration of all layers of multiplex networks for detection of communities.

No doubt that the use of heterogeneous networks can allow us to apply traditional topological measures very efficiently (with or without edge weights), but the fact that they contain different types of nodes and links in the same platform can make the analytical jobs very complex at times. Where as multiplex networks are easier and simpler to use and it is easy to implement existing algorithms of simple networks on them. However, there also exists the question of how to include the correlation or interdependence of the layers during various computations of topological measures. Our work differs from all the works described before in a sense that we are trying to explore the heterogeneity in the form of multiplex networks and we use the concept of multiplex topological measures for the purpose of link prediction in bibliographical networks. The use of multiplex structure keeps our model simple and easy to implement it for practical application. The networks used in our work are assumed to be correlated which means that the linking probability in one layer can depend on the linking pattern in other layers. The multiplex topological measures include new definitions of simple topological measures in the context of a multiplex network which also attempts to include the correlations of layers in a naive way.

5.3 Link prediction in multiplex network

Our approach includes computing simple topological scores for unconnected node pairs in a graph. Then we extend these attributes to include information from other dimension graphs or layers. This can be done in three ways: First we compute the simple topological measures in all layers; second is to take the aggregation of the scores; and third we propose an entropy based version of each topological measures which gives importance to the presence of a non-zero score in each layer. In the end all these attributes can be combined in various ways to form different sets of attribute values (vectors) characterizing each example or unconnected node pair.

Direct and indirect attributes Formally, if we have a multiplex graph $G =< V, E_1, \ldots, E_m >$ which in fact is a set of graphs $< G_1, G_2, \ldots, G_m >$ and a topological attribute X. For any two unconnected nodes u and v in graph G_i (where we want to make a prediction), $X(u, v)$ computed on G_i will be *direct* attribute and the same computed on all other dimension graphs will be *indirect* attributes.

Multiplex attributes The first category of multiplex attribute computes an ***aggregation of the attribute values*** over all layers. This aggregation can be done any of the existing functions like *min*, *max*, *sum*, *average* etc. For example, we choose to use *average*, so our new attribute is given by

$$X_{average} = \frac{\sum_{\alpha=1}^{m} X(u,v)^{[\alpha]}}{m} \; \forall \; u, v \in V \; and \; (u, v) \notin E_i \tag{5.3}$$

where m is the number of types of relations in the network (dimension or layer).

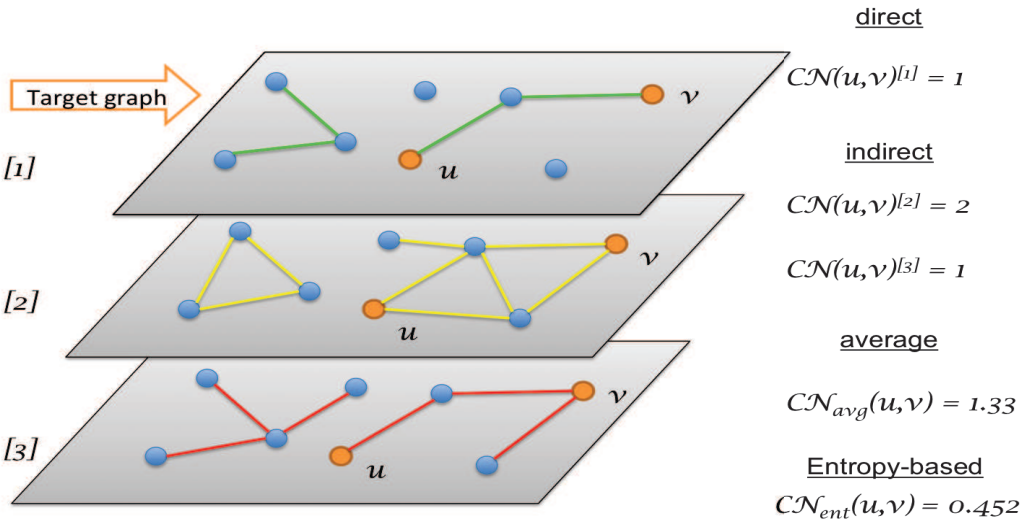


FIGURE 5.3: An example of computing direct, indirect and multiplex attributes based on number of common neighbors ($CN(u, v)$).

In the second category we propose a new attribute called *product of node degree entropy (PNE)* which is based on *degree entropy*, a multiplex property proposed by F. Battistion et al. [Battiston et al., 2013]. If degree of node u is $k(u)$, the degree entropy is given by:

$$E(u) = -\sum_{\alpha=1}^{m} \frac{k(u)^{[\alpha]}}{k_{total}} \log(\frac{k(u)^{[\alpha]}}{k_{total}}) \tag{5.4}$$

where $k_{total} = \sum_{\alpha=1}^{m} k(u)^{[\alpha]}$ and we define *product of node degree entropy* as

$$PNE(u, v) = E(u) * E(v) \tag{5.5}$$

We also extend the same concept to define entropy of a simple topological attribute, and call them ***entropy-based attributes*** X_{ent}

$$X_{ent}(u, v) = -\sum_{\alpha=1}^{m} \frac{X(u, v)^{[\alpha]}}{X_{total}} \log(\frac{X(u, v)^{[\alpha]}}{X_{total}}) \tag{5.6}$$

where $X_{total} = \sum_{\alpha=1}^{m} X(u, v)^{[\alpha]}$. The entropy based attributes are more suitable to capture the distribution of the attribute value over all dimensions. A higher value indicates uniform distribution attribute value across the multiplex layers. We address average and entropy based attributes as *multiplex attributes*.

Figure 5.3 illustrates our concepts using a simple example. We have three layers of graphs and we need to make prediction on the first layer which we call *target layer or target graph*. We compute different versions of common neighbors topological metrics for the selected nodes u and v, excluding and including the multiplex information.

5.4 Experiment

We evaluated our approach using data obtained from DBLP[1] databases corresponding to year between 1970-1979 like before. We create two datasets from three graphs, each corresponding to a different period of time. Each dataset has four years for learning or training and next two years are used to label the examples generated from the learning graphs. The examples are generated on the target layer on which we want to make the prediction. In this case it is the co-authorship layer and we predict the co-authorship links.

Table. 5.1, 5.2 and 5.3 summarize the information about the graphs, examples generated and datasets used for validating the approach. Figure 5.4 shows the visualization of the three graphs.

Years	**Properties**	**Co-Author**	**Co-Venue**	**Co-Citation**
1970-1973	*Nodes*	91	91	91
	Edges	116	1256	171
	Density	0.028327	0.306715	0.041758
1972-1975	*Nodes*	221	221	221
	Edges	319	5098	706
	Density	0.013122	0.209708	0.029041
1974-1977	*Nodes*	323	323	323
	Edges	451	9831	993
	Density	0.008673	0.189047	0.019095

TABLE 5.1: Graphs

Years		**# Positive**	**# Negatives**
Train/Test	**Labeling**		
1970-1973	1974-1975	16	1810
1972-1975	1976-1977	49	12141
1974-1977	1978-1979	93	26223

TABLE 5.2: Examples generated from co-authorship graph

Dataset	**Learning year**	**Test year**	K
Dataset 1	1970-1973	1972-1975	49
Dataset 2	1972-1975	1974-1977	93

TABLE 5.3: Datasets for experiment
(K is the parameter used for supervised rank aggregation based link prediction and is equal to the number of positive examples in the test sets.)

We selected the following topological attributes: Number of common neighbors (CN), Jaccard coefficient (JC), Preferential attachment (PA) [Huang et al., 2005], Adamic Adar coefficient (AA) [Adamic et al., 2003], Resource allocation (RA) [Zhou et al., 2009] and Shortest path length (SPL). These are simple measures which can be easily computed

[1] http://www.dblp.org

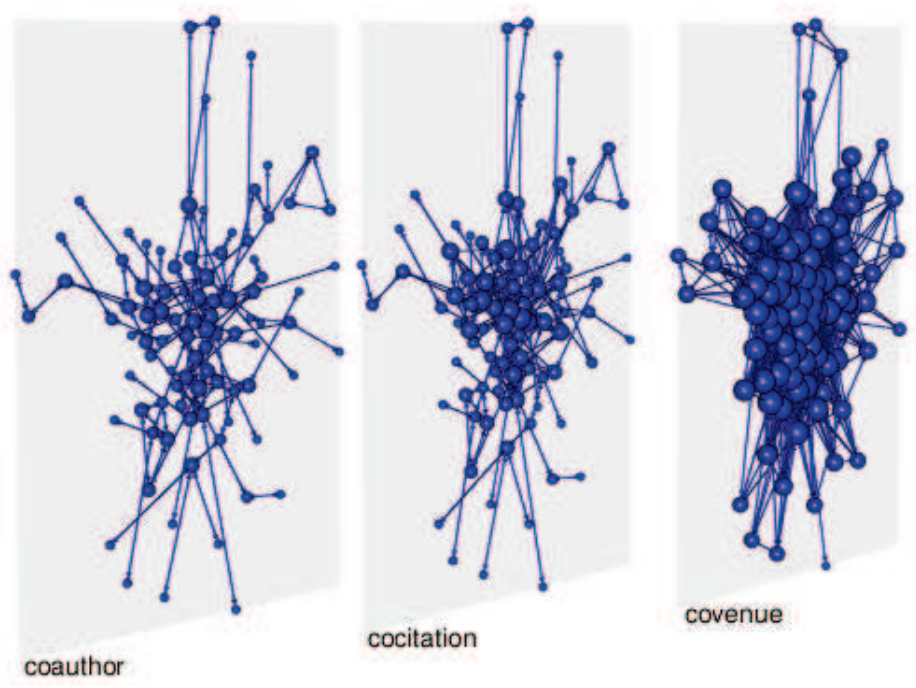


FIGURE 5.4: Multiplex network visualization for year 1970-1973 of DBLP

in less time. All these topological measures were computed on the three graphs. Their average and multiplex versions are also added as additional attributes and we also used product of node entropy as one additional entropy based multiplex attribute. We applied decision tree algorithm on dataset 1 to generate a model and then tested it on another dataset 2 (See table. 5.3). Decision tree was a quick choice owing to its simplicity and popularity. Also, we already had an in hand experience of using this algorithm for prediction task in simple co-authorship network. So, to get some quick results for our experiments, we decided to use this algorithm. We are using data mining tool Orange[2] for this. We use four types of combinations of the attributes creating five different sets namely:

- *Direct* (attributes computed only in the co-authorship graph);
- *Direct* + *Indirect* (attributes computed in co-authorship, co-venue and co-citation graphs);
- *Direct* + *Multiplex* (attributes computed from co-authorship graph with average attributes obtained from three dimension graphs, and also entropy based attributes);
- *Direct* + *Indirect* + *Multiplex* (attributes computed in co-authorship, co-venue and co-citation graphs, with average of the attributes, and also entropy based attributes) and
- *Multiplex* (average of attributes and entropy based attributes).

[2]http://orange.biolab.si

In this experiment, our goal is to predict co-authorship links. So, the target layer is co-author layer. But the same procedure can be used for prediction of links in any other layer. Figure. 5.5 shows the result obtained in terms of F1-measure and area under the ROC curve (AUC). We observed that contrary to our belief, the inclusion of indirect and multiplex attributes does not seem to improve the prediction result in temr of F1-measure. However, we can see that there is slight improvement in the result when we use only multiplex attributes for learning and validating our model. This shows that there is indeed some useful information whic can be captured by multiplex attributes and can be use in link prediction task. Also AUC increases for all the sets that include multiplex and indirect attributes for both datasets. This also justifies our belief on the usefulness of indirect and multiplex attributes.

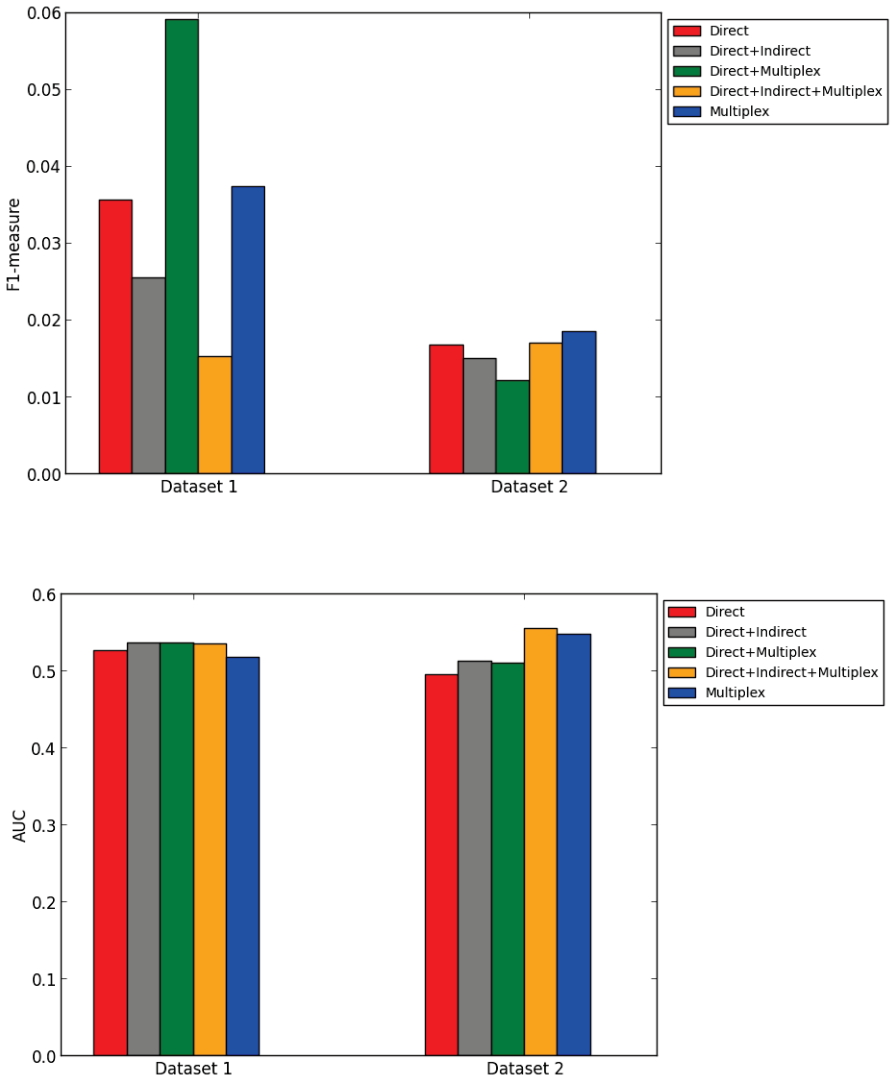


FIGURE 5.5: Results on the two datasets for Decision tree algorithm

We also applied supervised rank aggregation based methods on one of the multiplex networks (dataset 1). We report here the performance in terms of F1-measure in figure 5.6. We can see that rank aggregation based methods do not perform well with the indirect and multiplex attributes on dataset 1. Although with supervised Kemeny 1 the

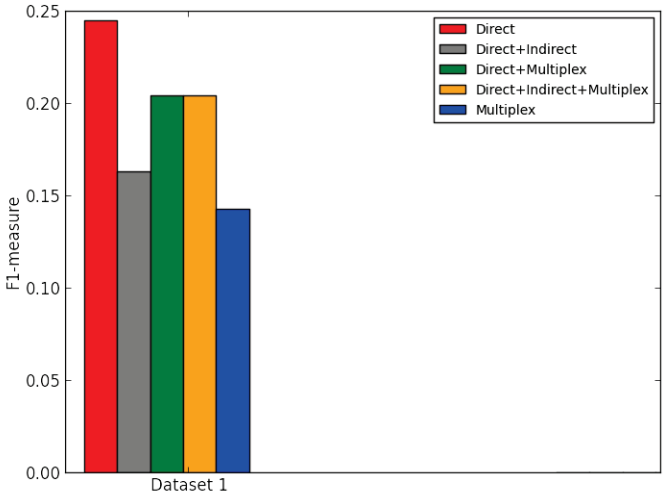
performance on inclusion of multiplex attributes is comparative to that of using only direct attributes.

The reason for not having a very good result with indirect and multiplex attributes can be due to the fact that we have not really verified the edge overlaps between layers. We have used the co-citation layer which has links based on the common bibliographical references made by two authors in their papers and the co-venue layer has links if the authors have published in same conferences. It is possible that these two layers are just the super-graphs of the co-authorship layer. That means, they may contain all the links that are also found in the co-authorship layer plus some new links. So they may not explicitly represent the multiplexity in the network. We believe that considering edge overlap and other such correlation measure can lead to having some different result. Moreover we have made a naive attempt of defining the multiplex topological measures only based on entropy. Other such concepts can also be implemented for the same purpose after a detailed verification of importance and applicability of the concepts. This may also result in more distinct and better performances.

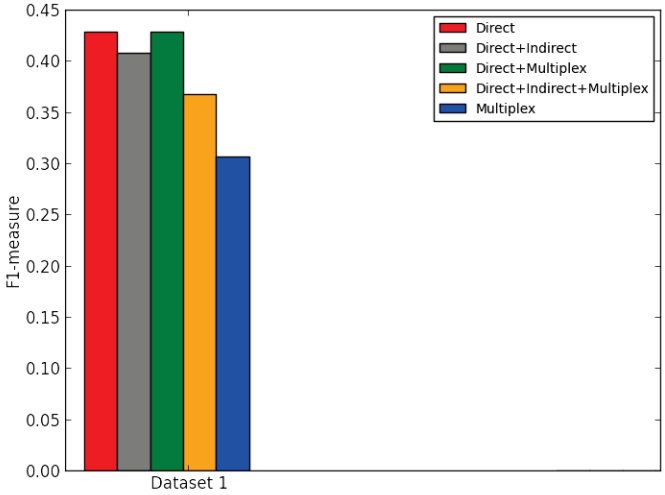
5.5 Conclusion

This chapter presents a brief overview of different works that take into account the heterogeneous nature of complex networks. We present our new approach of link prediction in multiplex networks. We propose some new and extended topological features that can be used for characterizing the unlinked node pairs for link prediction task, including also multiplex relation information. They can be applied to predict links in any of the layers of the network. We tested our supervised model for prediction of co-authorship links on datasets obtained from DBLP databases. The results were not extremely good, however learning and validating a supervised machine learning link prediction model on multiplex attribute set showed slight improvement in the result in terms of F1-measure. The performance of the model in terms of AUC was better on inclusion of indirect and multiplex attributes with direct attributes.

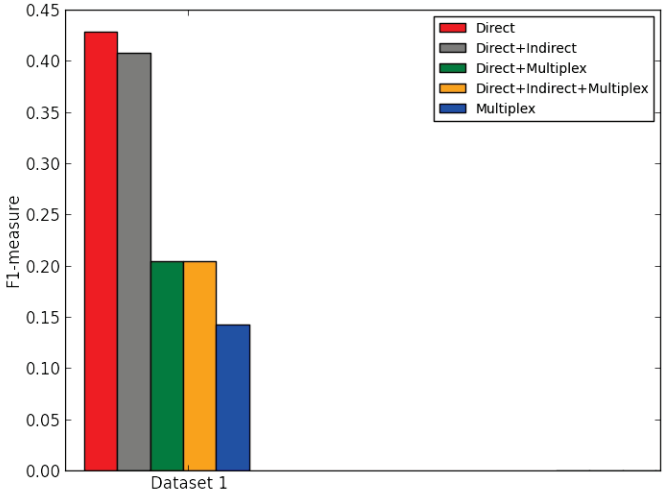
The advantage of our proposed approach is that it is a straightforward approach, simple in implementation with less complexity and can be used for prediction of any type of links in any layer. We have not yet considered the possibility of having inter-layer paths while defining the new multiplex attributes, which has been done in many of the state-of-art work. This can be left as a perspective for the time being and can be explored later to see its applicability in the task of link prediction.



a) Approximate Kemeny based link prediction algorithm



b) Supervised Kemeny based link prediction algorithm (weight computed by precision)



c) Supervised Kemeny based link prediction algorithm (weight computed by false positive rate)

FIGURE 5.6: Results for supervised rank aggregation based models

Chapter 6

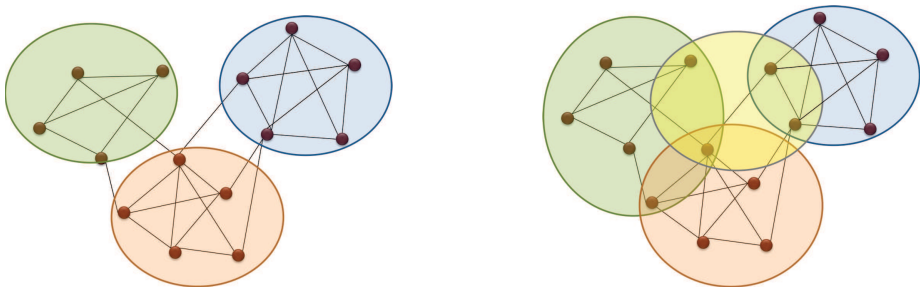
Communities and Link Prediction

6.1 Introduction

Community structures are very common in real world complex networks. Communities can be defined as groups of nodes in a network which are generally more connected within each other than with nodes exterior to the communities. Members of a community are supposed to have some common properties or play some kind of common roles in the network. The semantic interpretation of a community depends largely on the type of network or type of information presented by a network. For example, in a metabolic network or a protein-protein interaction network communities can be a set of proteins (nodes) performing a certain biological function in a cell [Guimera and Nunes Amaral, 2005; Guimerà et al., 2007]. In an e-commerce network communities can consist of a set of customers with similar choices of products or similar purchase history [Benchettara et al., 2010b]. In world wide web they can be a set of web-pages related to same topic [Flake et al., 2002].

Community detection and link prediction are two important fields of research in complex network analysis. They have been running in parallel since long and it is very recently that researchers have come up with the thought of using community information for link prediction [Benchettara, 2011; Soundarajan and Hopcroft, 2012]. However link and path information have been already in use for finding communities [Fortunato, 2010; Newman, 2004a; Yakoubi and Kanawati, 2014]. We were always interested to explore communities for the sake of link prediction task but our real motivation came from a small experiment that we did on scientific collaboration network created from DBLP data. We observed that most of the future collaborations occur between authors belonging to different communities. This observation gave us the idea to use the same concept for sampling the learning dataset in order to learn a better classification model which we use for link prediction.

In this chapter we will discuss how the task of community detection can help in link prediction in various ways. We precisely deal with the problem of sampling huge data used in link prediction. Section 6.2 provides a description about the problem of community detection and various traditional approaches for community detection. Section 6.3 describes a few work on link prediction that use community information. Section 6.4 presents a brief overview on sampling of data in the context of link prediction and our



(A) Disjoint communities (B) Overlapping communities

FIGURE 6.1: Communities in a network

proposed method of using communities for the same. Section 6.5 provides the experimental details on DBLP co-authorship networks.

6.2 Community detection approaches

Community detection refers to the problem of finding subgroups or clusters of nodes in a network that form communities based on some criteria. These subgroups are found according to optimization of some target function. Community detection algorithms can find communities which can be *disjoint* or *overlapping*. Disjoint communities are partitions in a network where one node can belong to only one community at a time. Overlapping communities on the other hand, are soft clusterings that allow a node to belong to more than one community simultaneously. Figure. 6.1 shows the two types of partitions.

Problems of finding disjoint and overlapping communities are NP-hard [Brandes, 2008]. Hence many community detection algorithms go for an approximate optimization using heuristics. The community detection approaches can be broadly classified as group-based approaches, network-based approaches, propagation based approaches and seed-centric approaches [Yakoubi and Kanawati, 2014].

Group-based approaches focus on identifying groups of nodes that are highly connected. Connection pattern can be high mutual connectivity or a slightly relaxed way of high mutual reachability. These approaches mostly involve identification of densely connected subgroups like k-core, cliques and quasi cliques. Examples of such approaches are clique percolation algorithms [Adamcsek et al., 2006; Sun and Gao, 2009]. Another approach is proposed in [Verma and Butenko, 2012] , where authors introduce the concept of k-community which is defined as a connected subgraph of a network in which for every couple of nodes have number of common neighbors equal to or more than k. The computational complexity of k-cores and k-communities is polynomial. The k-communities are mostly used as seeds for computing communities. In another work [Peng et al., 2014], authors propose to compute k-cores as mean to accelerate computation of communities using standard algorithms but on size-reduced graphs.

Network based approaches consider the connection pattern in the entire network to find communities. These include many classical clustering based algorithms like spectral clustering, graph partitioning, hierarchical clustering etc. Spectral clustering tries to partition a network into clusters by using the eigenvectors of the matrices. It consists

of transforming nodes of the network into a set of points in space whose coordinates are elements of eigenvectors. The points are clustered using standard k-means clustering. Such algorithms are presented in [White and Smyth, 2005]. Hierarchical clustering algorithms try to explore hierarchical structure in a network i.e. many levels of grouping of nodes. In such cases smaller clusters may exist inside large clusters. Hence these algorithms reveal multilevel structure of the network. These algorithms group nodes with high similarity. Hierarchical clustering approaches are classified into two categories [Fortunato and Barthélemy, 2007]:

1. Agglomerative approaches, in which forms clusters by iteratively merging the groups of nodes if their similarity is high. These are bottom-up approaches and the algorithm starts by considering each node as a cluster and then moving up towards bigger clusters.

2. Divisive or separative approaches, in which clusters are split by iteratively removing edges connecting nodes with low similarity. These are top-down approaches and the algorithm starts by considering the whole network as one big cluster.

Hierarchical algorithms have been presented in [Blondel et al., 2008; Newman, 2006, 2004b; Pons and Latapy, 2006]. Hierarchical clustering have the advantage unlike spectral clustering, that they do not need any prior information about numbers of clusters or communities to be found. However the main limitation is that they do not discriminate between any partitions obtained by the process and there is no way to choose which level of partitions show a better community structure. To deal with this issue some quality functions are required and hence optimization of a quality function for graph partition came into the scene. A quality function is a function that assigns a score to each partition to quantify the quality of the cluster [Fortunato, 2010]. *Modularity* is the most widely used quality measure in community detection [Newman, 2004a]. It is based on the concept that the possible existence of clusters is revealed by the comparison between the actual density of edges in a cluster and the expected density of the cluster regardless of community structure. It advocates the idea that a good community is more connected inside than with the rest of the network. It is defined as follows. Suppose $P = C_1, C_2, C_3, ..., C_k$ is a partition of nodes in a graph $G =< V, E >$. The modularity of the partition P is given by:

$$Q(P) = \sum_{C \in P} e_{inter}(C) - e_{out}(C) \tag{6.1}$$

where $e_{inter}(C) = \frac{\sum_{i,j \in C} A_{ij}}{2m}$ is fraction of links inside community C and $e_{out}(C) = \frac{\sum_{i \in C, j \in V} A_{ij}}{2m}$ is the fraction of links of nodes inside community with nodes outside the community. The computational complexity of Q is $O(m)$ [Newman, 2004b]. Some recent work have extended the definition to bipartite and multipartite graphs [Du et al., 2008; Liu and Murata, 2009; Murata, 2009a,b, 2010; Neubauer and Obermayer, 2009].

The *Louvain* algorithm [Blondel et al., 2008] is one very well known example of agglomerative approaches. The algorithm is composed of two phases. First, it looks for small communities by optimizing modularity in a local way. Second, it aggregates nodes of the same community and builds a new network whose nodes are the communities. Two adjacent communities merge if the overall modularity of the obtained partition can be enhanced. These steps are repeated iteratively until a maximum of modularity is

reached. The computational complexity of the approach is empirically evaluated to be $O(nlog(n))$.

In divisive approaches different criteria can be used for splitting the cluster. The Newman-Girvan algorithm is the most known representative of this class of approaches that use modularity optimization in a separative way [Newman, 2004b]. The algorithm is based on the simple idea that a tie linking two communities should have a high betweenness centrality. This is naturally true because an inter-community tie would be traversed by a high fraction of shortest paths between nodes belonging to these different communities. Considering the whole graph G, the algorithm iterates for m times, cutting at each iteration the tie with the highest betweenness centrality. This allows to build a hierarchy of communities, the root of which is the whole graph and leafs are communities composed of isolated nodes. Partition of highest modularity is returned as an output. The algorithm is simple to implement and has the advantage to discover automatically the best number of communities to identify. However, the computational complexity is rather high: $O(n^2m + n^3 \log(n))$. This is makes it unsuitable for large-scale networks.

Yet another interesting work has been proposed by P. Pons et al. [Pons and Latapy, 2006] which is based on finding node similarity using random walks. The distance is calculated using the probabilities that a random walker moves from one node to another in a fixed number of steps. The numbers of steps should be large enough to cover a significant portion of the network. The nodes are grouped into communities through an agglomerative hierarchical clustering and modularity is used to find the best partition in the resulting dendrogram. This algorithm is commonly known as *Walktrap*. The algorithm runs with a time complexity of $O(n^2d)$, where d is the depth of dendrogram. d being often small for real graphs which are sparse, the practical computational complexity is $O(n^2 \log n)$ [Fortunato, 2010].

Last but not the least, is the method of *Infomap* partitioning algorithm proposed by M. Rosvall et al. [Rosvall et al., 2009]. With greedy modularity optimization, this method produces a partitioning of the network which are generally of very high quality. S. Fortunato [Fortunato, 2010] evaluated several detection methods and concluded that, of the methods tested, Infomap was the most accurate. This algorithm attempts to identify a coarse-grained representation of how information flows through a network. The goal is to optimally compress information needed to describe the process of information diffusion across the graph. Each cluster is given a name or number and each node inside a cluster is given a local name or number. Nodes in different clusters may have same local names. A random walk in the network can thus have a structure: **[name of cluster C_i- local name of node v_i - local name of node v_j - ... - local name of node v_k - a code indicating a link outside - name of cluster C_j]** . An example of this is given in figure 6.2. The goal of the algorithm is to find a partitioning and labeling of nodes in the network so as to minimize the expected length of a random walk's description. An initial clustering is identified using a greedy algorithm, and simulated annealing is then used to improve the results.

The modularity optimization based approaches make some implicit assumptions that the best partition in the graph is the one that maximizes the modularity and if a network has a community structure, then partitions inducing high modularity values are structurally similar. However in recent works it has been shown that these assumptions may not be true. In the work of B.H. Good et al. [Good et al., 2010], authors show that the modularity function exhibits extreme degeneracy. It accepts an exponential number

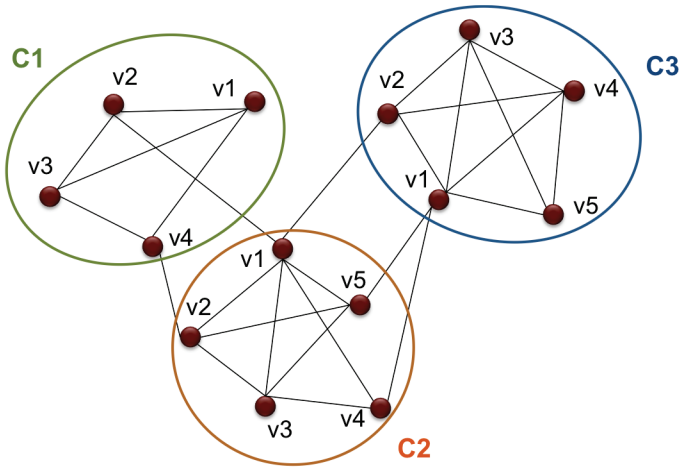


FIGURE 6.2: An example for Infomap.
The nodes can have same local names inside communities. A random path between node v1 of community C1 and node v5 of community C3 is [C1 - v1 - v2 - C2 - v1 - C3 - v2 - v3 - v5]

of distinct high scoring solutions and typically lacks for a clear global maximum. In a few other work [Fortunato, 2010], it has been shown that communities detected by modularity maximization have a resolution limit. These serious drawbacks of modularity-based algorithms motivated the research for alternative approaches. Some interesting emerging approaches are label propagation approaches [Raghavan et al., 2007] and seed-centric approaches [Yakoubi and Kanawati, 2014] which we will describe later.

Propagation based approaches come with an advantage of a comparatively faster execution time unlike modularity based approaches like *Louvain* approach which have high computational complexity that makes them costly to be used in large-scale networks. One prominent work in this category is the work proposed by U. N. Raghavan et al. [Raghavan et al., 2007] which is a simple and fast method based on *label propagation* algorithm. The basic idea is that each node v_i in the network is assigned a label l_i. a synchronous update of labels is done by selecting the most frequent label in the direct neighborhood.

$$l_i = arg\max_{l} |\Gamma^l(v_i)| \tag{6.2}$$

where $\Gamma^l(v_i) \subseteq \Gamma(v_i)$ is the set of neighbors of v_i that have label l. The algorithm iterates until a stable state is reached where no nodes further change their label. The ties are broken randomly. Nodes having same labels are grouped as a community. The computational complexity of each iteration is $O(m)$. Hence the overall computational complexity is $O(rm)$ if r is the number of iterations before convergence. However, these algorithms have a some serious drawbacks:

1. There is no guarantee of convergence to a stable state.
2. It lacks robustness as different runs may produce different partitions due to random breaking of ties.

A few attempts have been made to deal with these limitations. Asynchronous, and semi-synchronous label updating have been proposed to hinder the problem of oscillation and improve convergence conditions [Cordasco and Gargano, 2012; Raghavan et al., 2007]. However, these approaches harden the parallelization of the algorithm by creating

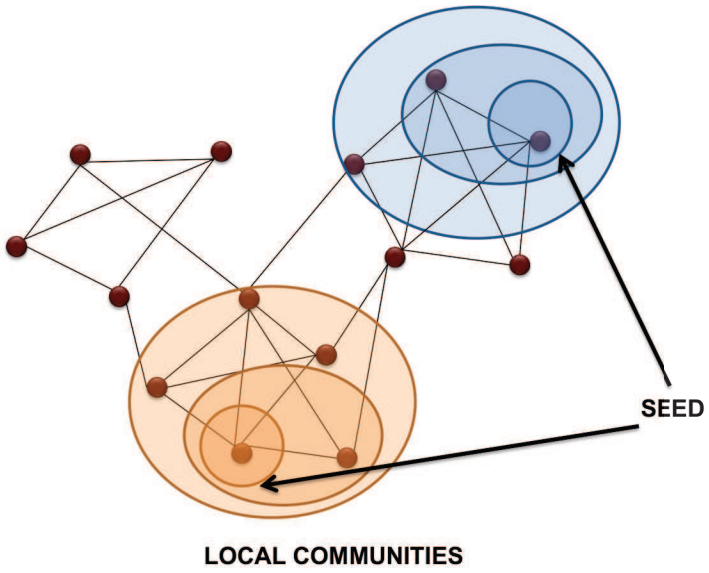


FIGURE 6.3: Seed centric local communities in a network

dependencies among nodes and they increase the randomness in the algorithm making the robustness even worse. Another interesting way to handle the instability of label-propagation approaches consists of simply executing the algorithm k times and apply an ensemble clustering approach on the obtained partitions

Seed centric approaches are based on the idea of identifying some important nodes in the network, called *seed nodes* around which local communities can be computed [Shah and Zaman, 2010; Yakoubi and Kanawati, 2014]. There are three basic steps in these approaches:

1. Identification of seed nodes.
2. Local community detection around the seed nodes.
3. Final community computation from the sets of local communities found in previous step.

Figure 6.3 shows a simple example of such approaches. A special case of seed centric approches is *Leaders driven* algorithms. Nodes of the network are classified into two categories: *leaders* and *followers*. Leaders are the nodes that are representative of communities. Followers can be assigned to most suitable communities identified by the leaders. Different methods can be used for node classification and community assignment.

The work of R. Khorasgani et al. [Khorasgani et al., 2010] propose an approach based on $k - means$ clustering algorithms. With an input of number of communities (k) to be identified, k nodes are randomly selected which are labeled as leaders and the rest are followers. Each leader represents a community and followers are assigned to the community of nearest leader node. Different levels of neighborhood are allowed in this. If a follower is not able to find a nearby leader node, then it is labeled as *outlier*. When all followers are assigned a community, then a new set of leader nodes are computed. At this step the most central node in the communities found are considered as the new leaders. The process is repeated with the new set of leaders until stabilization is reached.

The convergence speed depends on the quality of initially selected k leaders. The best approach to find initial set of leaders, according to experimentation is to select the top k nodes that have the top degree centrality and that share less common neighbors.

Another interesting work is that of LICOD proposed in [Yakoubi and Kanawati, 2014]. It is also a seed-centric approach having leader and follower nodes. The different steps of the algorithm are described below:

1. Search for the nodes likely to be leaders. This can be done using ranking of nodes based on various criteria. Classical metrics of centrality are very useful for this.
2. The list of leaders thus found is further reduced by grouping leaders that have higher probability of being in the same communities.
3. For each node in the network (leaders/followers), membership degrees to all communities (represented by the leaders) is found. A ranked list of communities based on membership degree is obtained for each node. The communities with highest membership degree are ranked on the top.
4. Each node will then update its community preference list by merging it with those of its immediate neighbors. Different rank aggregation techniques can be used for this purpose (See chapter 4). This step will be repeated until stabilization is obtained for the ranked list of communities at each node.
5. In the end each node is assigned to the top community in its final ranked list of membership.

Having a look on a few important works in community detection we will proceed to see how they have been used for the task of link prediction. Further details and surveys on community detection algorithms can be found in [Fortunato, 2010; Leskovec et al., 2010; Papadopoulos et al., 2012; Tang and Liu, 2010].

6.3 Link prediction using community information

Looking at the work combining link prediction with community detection we can say that there are many more things to be explored in this regards. However, few researchers have attempted to use network partitions for link prediction network.

One such work is that of A. Clauset et al. [Clauset et al., 2008], where a hierarchical structure of the network is found by creating a binary dendrogram that joins nodes into groups. It is a bottom-up approach where each node belongs to its own community in the beginning. Each internal node in the dendrogram joins two groups together to form a larger group. For two nodes u and v, number of links between them is calculated and normalized by total number of possible links between the two. The value thus obtained is interpreted as the probability of getting the two nodes linked.

Another category of link prediction models is that of stochastic block model [Airoldi et al., 2006; Lü and Zhou, 2011]. In these kinds of models all nodes are grouped to form partitions. The linking probability between two nodes is then found depending on the group memberships of the nodes. These two methods are also discussed in the state-of-art in chapter 3. The main disadvantage of these approaches is that these are practically

inapplicable to large scale networks due to the complexity involved in finding an optimal dendrogram or partition.

We found some interesting works that use communities in a different way to enhance the values of attributes characterizing candidate node pairs. In the link prediction approach proposed by N. Benchettara [Benchettara, 2011], author have presented a supervised machine learning based approach for predicting new links in bipartite graphs. In this they have used community information as an attribute characterizing unlinked node pairs. This attribute can have value 1 or 0, based on the fact that the two concerned nodes belong to same community or not. They show enhancement of prediction result in terms of F1-measure on including community information during learning and validation.

In a recent work proposed by S. Soundarajan et al. [Soundarajan and Hopcroft, 2012], the authors propose to include community information in attributes but in a different way. They focus mainly on similarity based link prediction measures like common neighbors, Jaccard's coefficient, resource allocation etc. They propose to modify these simple similarity measures to add community related information. The main principle of the work is that two nodes sharing common neighbors in same community can have greater probability of having a new link. The authors propose to assign extra points for neighbors shared between two nodes u and v that are in the same communities as u and v. Also extra points can be given when u and v are in the same communities. Points are also added if both criteria are true. Based on these conditions, authors propose five different ways of adding extra points to the topological scores obtained for pairs of unlinked nodes. In the end, they present a hierarchical link prediction model based on the work of Clauset et al. [Clauset et al., 2008]. They do a 10 fold cross validation to find results in terms of precision and AUC. The community enhanced similarity measures outperform the base metrics in terms of precision while AUC is comparable for both.

A small example of this is provided in figure 6.4. Common neighbors for any two nodes x and y is computed as $CN(x, y) = |\Gamma(x) \cap \Gamma(y)|$. With a set of communities $C = C_1, C_2, ..., C_k)$ found by any community detection algorithm applied to the network, $C(x)$ is the subset of communities that contain node x. We chose two ways to add points to common neighbor score:

1. Extra points are added for each common neighbors that lie in the same community as x and y i.e

$$CN_1(x, y) = CN(x, y) + \sum_{u \in \Gamma(x) \cap \Gamma(y)} |C(x) \cap C(u) \cap C(y)| \tag{6.3}$$

2. Extra points are added if both nodes x and y are in the same communities.

$$CN_2(x, y) = CN(x, y) + |C(x) \cap C(y)| \tag{6.4}$$

Figure 6.4 shows calculations of these versions of common neighbors score for two pairs of nodes in a sample network.

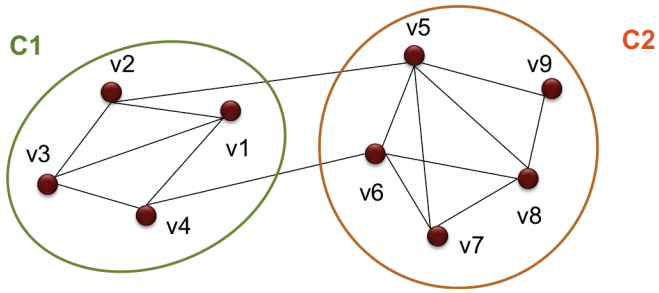


FIGURE 6.4: An example to find modified versions of common neighbors
For node pair $(v2, v4)$: $CN = 2$; $CN_1 = 2 + 2 = 4$; $CN_2 = 2 + 1 = 3$
For node pair $(v2, v6)$: $CN = 1$; $CN_1 = 1 + 0 = 1$; $CN_2 = 1 + 0 = 1$

6.4 Data sampling using community detection algorithms

Under-sampling or down sampling, as we have seen before in chapter 3, is a process of selecting or filtering out a portion of data so as to reduce the size into a manageable amount and with a goal of avoiding or minimizing loss of valuable information. It has great importance in the task of supervised or unsupervised link prediction, especially when we are dealing with huge and sparse networks where the number of potential candidates is often extremely large. The goal of sampling of data should be certainly to reduce the size, while not hampering the final prediction result in an adverse way. Another use of this approach is to deal with the class imbalance problem or class skewness which greatly affects classification based models. In such a scenario the models learned during the training phase are unable to adapt equally for the minority classes as they do for majority class. So while they are used for classification, there are greater chances for having a bias towards the majority class. In link prediction, the correct classification of positive class instances is more important but they represent the minority class, the negative class being the majority. So it is very crucial in link prediction, to remove a few candidates from the majority class (negative class) so as to learn a better model.

A seminal work on under-sampling, is the approach proposed by M. Kubat et al. [Kubat et al., 1997]. In this work, authors discuss the problem of class imbalance in the context of machine learning. They show how learning a model from a dataset having high class imbalance can result in a highly biased prediction by the classifiers. Thereby, they propose a solution named *One-sided selection* in which all positive examples (minority class) are kept intact and random selection is done on the negative examples (majority class) to find a set of representative negative examples. This removes all negative examples that are believed to be borderline or noisy. This approach is very relevant in the context of link prediction where we definitely want to keep all the positive examples while learning the model.

Selection of negative examples can be done randomly, as it is case in the work of M. Kubat et al. [Kubat et al., 1997] or based on certain other criteria relevant in the context of link prediction. In [Lichtenwalter et al., 2010], authors present a distance based sampling method. They explore the role of distance between nodes in determining class imbalance ratio. They suggest to restrict the distances between the nodes in the network to a threshold say d. So with increase in d the numbers of potential candidates for link prediction will also increase and this increase is very sharp. Thus, authors suggest to treat samples obtained for different distance threshold separately in supervised learning task.

The sampling of data is mostly done on the learning data. Sometimes it is required to sample test data also especially when extremely large number of test examples causes unreasonable demands on processing of resources and storage. If for any reason this has to be done, proper care should be taken based on the context where link prediction is to be applied. For example during link prediction in terrorist networks where identifying every possible criminal links is essential, sampling of test data may lead to missing a truly positive link and this will be a serious issue. So in such cases where missing a true positive link is much important, sampling of test data should be avoided. Where as it can be very well used for link prediction in social collaboration networks in the context of recommendation of interesting links to users. In such cases missing a few positive links during prediction is affordable and due to the huge sizes of network which tends to grow even more with time, under-sampling can be very useful. Also while sampling test data, it is important to respect the distribution of positive and negative examples.

6.4.1 Community based under-sampling

During our research work, we often came across the concept of community detection where nodes of the network are grouped based on some common characteristics or role played by them in a network.

Motivation: While working on multiplex networks, we did a small experiment to see the effect of communities in the co-authorship network. We wanted to see how links are distributed across different communities in a co-authorship network. What we did was, we applied a community detection method to the three layers of multiplex networks of DBLP data (see chapter 5, section 5.4). A multiplex network is network having a layered structure showing different kinds of relations between same types of nodes. In this experiment, the three layers represent co-authorship relation, co-venue relation, and co-citing relation between author nodes respectively. The three sets of communities thus found were combined to find a single set of communities. This was used to form a coarsed (see section 6.6) co-authorship graph where the communities (groups of nodes) were represented as nodes and edges were added if there is at least one link between nodes belonging to two communities. Thus we get a weighted coarsed graph, the weights on edges being the number of links shared between the nodes of two corresponding communities. The links can lie inside communities (*intra-community links*) and they can be between nodes belonging to different communities (*inter-community links.* The *inter-community links)* can be of two kinds: One that are shared between communities which are already linked in coarsed graph and the other are the links that are shared between communities which are not directly linked in coarsed graph. So if we categorize links in a network, there will be three basic types:

- **Type 1:** Links inside a community i.e. intra-community links.
- **Type 2:** Inter-community links when the communities are not directly connected in the coarsed graph.
- **Type 3:** Inter-community links when the communities are directly connected in the coarsed graph.

Figure 6.5 illustrates the same diagramatically.

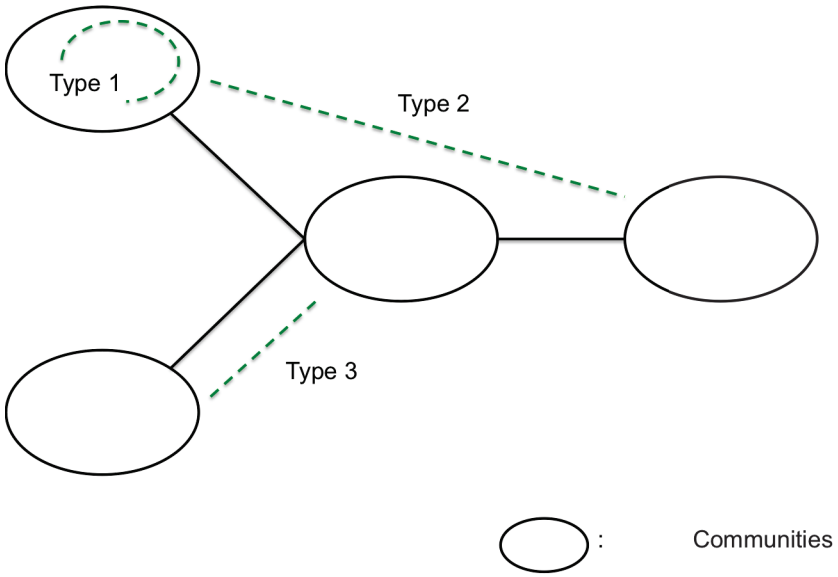


FIGURE 6.5: Distribution of links inside and outside communities.
Type 1: Links inside a community (intra-community links); Type 2: Inter-community links when the communities are not directly connected in the coarsed graph; Type 3: Inter-community links when the communities are directly connected in the coarsed graph.

In our experiment, we focused on the distribution of positive examples in communities to see in which area more positive links appear. We counted the number of positive examples on co-authorship network belonging to three types. This has been accounted in table 6.1. As we can see in the table we used two well known community detection algorithms Label propagation algorithm (LPA) and Walktrap, for finding communities. Communities were found in multiplex network having three layers of co-authorship, co-citation and co-venue. LPA, being an unstable algorithm, was run 50 times on each layer, thus finding 50 sets of communities on each. These were then combined to form a single set. For Walktrap we chose the default random path length i.e 4. The communities thus obtained were used to form the compressed co-authorship graphs for different periods of time.

We observed that most of the positive examples in the co-authorship network are inter-community links. Of these, maximum number of positive examples are formed between nodes belonging to communities which are not linked with each other in the condensed graph. That means, these communities did not have any collaboration between their authors in the past. And the minimum numbers of positive examples were found inside communities. The reason may be that all possible links have already been established inside a community or the authors inside a community prefer to collaborate more outside their community. This can be a reality in the field of scientific collaboration. But we cannot say the same is true in other kinds of network like social interaction networks or biological networks. The pattern of appearance of positive links can differ in these networks. Hence, we would like to specify here that the outcome of this study is very specific to scientific collaboration network.

However this observation made us think if most of the positive examples lie between communities, what will happen if we restrict the negative examples to the same region also. This led us to come up with the concept of under-sampling using communities which is described next.

Community detection method	**Years**		**#POS**	**#Type 1**	**#Type 2**	**#Type 3**
	Learn	**Label**				
LPA (n=50)	1970-1973	1974-1975	16	1	12	3
	1972-1975	1976-1977	49	0	35	14
	1974-1977	1978-1979	93	2	86	5
	1980-1983	1984-1985	426	23	365	38
Walktrap (steps=4)	1970-1973	1974-1975	16	1	9	6
	1972-1975	1976-1977	49	4	30	15
	1974-1977	1978-1979	93	4	68	21
	1980-1983	1984-1985	426	23	313	90

TABLE 6.1: Distribution of positive examples inside and outside communities
#POS: Numbers of positive examples found in the learning graphs
#Type 1, #Type 2, #Type 3: Numbers of positive examples of Type 1, Type 2, Type 3

Community based under-sampling of negative examples: We believe that the information captured in communities can very well be used in filtering of node pairs more relevant for the prediction task. This can allow us to have a safe under-sampling of examples without or less loss of information. Our approach is based on *One sided sampling* [Kubat et al., 1997]. So we keep all the positive examples. Out of negative examples, we propose to select the node pairs where both nodes do not belong to the same community. That means, they lie in the belt of inter-community links. This is the region where we observed the presence of most of the positive examples. Hence the negative examples that are found outside the communities can be more informative than those found inside the communities. These node pairs can better represent the negative class and can help to learn a better classification model.

Algorithm 3 Community based under-sampling

Input: $G =< V, E >$ where V is set of nodes and E is set of edges;
Community detection algorithm A;
(P, N) where P is set of positive examples and N is set of negative examples such that $(x, y) \in N$ where $x, y \in V$ **and** $(x, y) \notin E$
Output: (P, N') where N' is set of sampled negative examples

Apply A on G to find communities $C = C_1, C_2, ..., C_k$
If a node v belongs to a community C_i, then $membership(v, C) = C_i$
Initialize empty list N'
for (x, y) in N **do**
  $check =$ **false**
  if $membership(x, C) == membership(y, C)$ **then**
    $check =$ **true**
  end if
  if $check ==$ **false then**
    $N'.add((x, y))$
  end if
end for
return (P, N')

6.5 Experiments

We implement our concept of under-sampling on the same data of DBLP co-authorship networks formed from data corresponding to year 1970-1979. The information about the three graphs for year 1970-1973, 1972-1975 and 1974-1977 respectively can be found in table 4.1 in Chapter 4. We have used four community detection methods: Louvain [Blondel et al., 2008], Walktrap [Pons and Latapy, 2006], Infomap [Rosvall et al., 2009] and LICOD [Yakoubi and Kanawati, 2014]. The numbers of communities found by each algorithm on different graphs is reported in table 6.2. Also, we compute random samples and distance based samples. For random sampling, we have selected randomly from lot of entire negative examples, n examples keeping n as close as possible to the numbers found by the community based methods. For distance based sampling we have used two values of distance between nodes of negative examples $d \leq 5$ and $d \leq 10$. The number of examples found from all these sampling techniques has been reported in table 6.3.

Graphs	**Louvain**	**LICOD**	**Walktrap**	**Infomap**
1970-1973	9	9	8	16
1972-1975	14	29	17	37
1974-1977	17	25	27	47

Table 6.2: Number of communities found in DPLP co-authorship graphs

Graphs		**# POS**	**# NEG**	**Sampling method**	**# Sampled NEG**
Train	**Label**				
1970-1973	1974-1975	16	1810	*Louvain*	1639
				Walktrap	1553
				LICOD	1507
				Infomap	1737
				Distance $\leq$ *5*	810
				Distance $\leq$ *10*	1767
				Random	1648
1972-1975	1976-1977	49	12141	*Louvain*	11122
				Walktrap	11569
				LICOD	11382
				Infomap	11941
				Distance $\leq$ *5*	3686
				Distance $\leq$ *10*	11030
				Random	11123
1974-1977	1978-1979	93	26223	*Louvain*	24833
				Walktrap	23835
				LICOD	25150
				Infomap	25839
				Distance $\leq$ *5*	7832
				Distance $\leq$ *10*	24222
				Random	24831

Table 6.3: Original and sampled examples found on co-authorship graphs

Using all these examples we do a supervised machine learning based classification using a simple decision tree algorithm. For this we use *TreeLearner* decision tree algorithm of

data mining tool Orange[1] [Demšar et al., 2013] with default parameters. The datasets used for this are listed in table 6.4.

Dataset	Learning year	Test year
Dataset 1	1970-1973	1972-1975
Dataset 2	1972-1975	1974-1977

Table 6.4: Datasets for experiment with supervised machine learning algorithms

Training of the model is done on sampled datasets where as validation of the model is done on the original and complete dataset containing all negative examples. Figure 6.6 shows the performance of the link prediction model in terms of F1-measure and AUC. Results of random sampling correspond to the average result of 10 random samples based models. The lower values for AUC is due to the fact that we test our model on the original raw dataset derived from graphs without any sort of filtering unlike a few state-of-art work. We believe that sampling of test data will not give the true picture of the performance of the method in real scenario and should be avoided.

It can clearly be observed from the figure that learning from a set of examples sampled using communities produces a model that outperforms models trained on original datasets, random samples and distance based samples. One can also notice that the training done on random samples produce better result than that of distance based samples. The failure of distance based sampling may be due to the fact that nodes which are closer are theoretically more similar and have a greater tendency to form an edge. Thus removing the ones with distance greater than a threshold causes loss of some relevant information about negative class. Within community based sampling we see that Louvain method does not give a good result on dataset 1 where as it is the best choice for dataset 2. The other three community algorithms have a comparative result in both datasets in terms of both F1-measure and AUC and their result is almost consistent on both datasets.

Another important point that we notice here is that Walktrap performs quite well in both datasets where as the performance of distance based sampling methods is not as good. The basic idea behind Walktrap is that if we perform random walk in a network, then these walks are likely to stay within same communities as there are very few edges that lead outside a given community. The distance to be covered in random walks can be restricted to a certain length and the best modularity decides which level of partitions to chose form the dendrogram obtained. So, we can say that distance and path length can serve as a good criterion for under-sampling but careful selection of constraints are needed to make it useful for the task.

[1] http://orange.biolab.si/

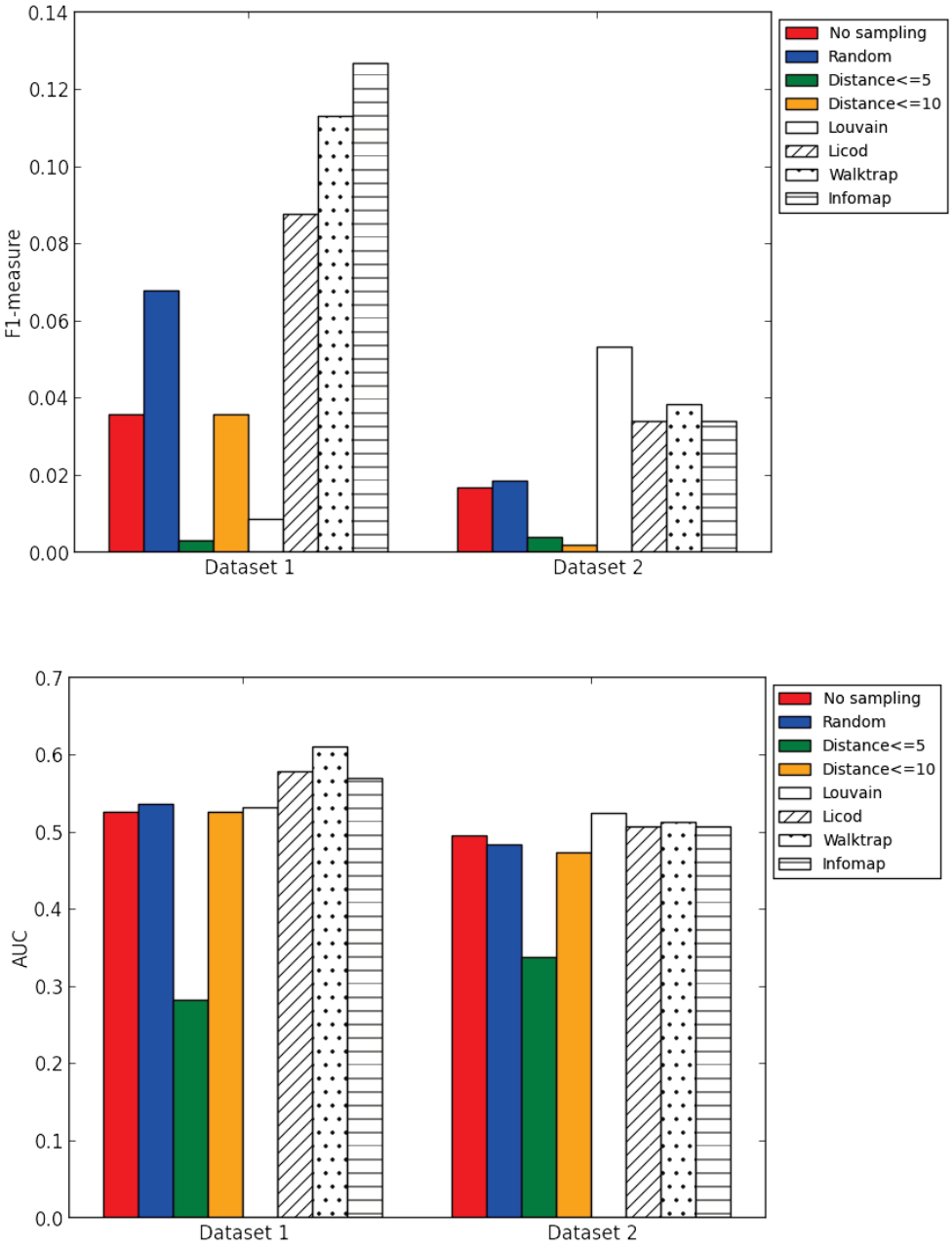


FIGURE 6.6: Results on the two datasets for Decision tree algorithm

6.6 Large network coarsening using communities: A perspective

Communities can be used for another purpose in analysis of complex networks. They can be used for corsening of large graphs. Graph coarsening is the process of grouping nodes together and building condensed and smaller graphs from these groups. It is a technique largely used for multi-level partitioning of huge graphs. It consists of three steps: Coarsening, Initial partitioning and Uncoarsening [Buluç et al., 2013; Karypis and Kumar, 1995]. Main goal of coarsening is to gradually approximate the original problem and input graph with fewer degrees of freedom [Buluç et al., 2013]. This goal is achieved in multilevel partitioning by creating a hierarchy of condensed graphs with decreasing sizes in such a way that cuts in the coarsed graphs can reflect the partition in the original fine graph. Coarsing is usually stopped when the graph is sufficiently small to be initially partitioned using any standard algorithm. After obtaining the initial partitions, uncoarsening is done which is the process of safely recovering the original graph from the condensed form. The coarsed graph is mapped to fine level and the partition is improved using some local improvement method. This process of uncoarsening and local improvement is carried on until finest hierarchy if achieved. This process is diagramatically illustrated in figure 6.7.

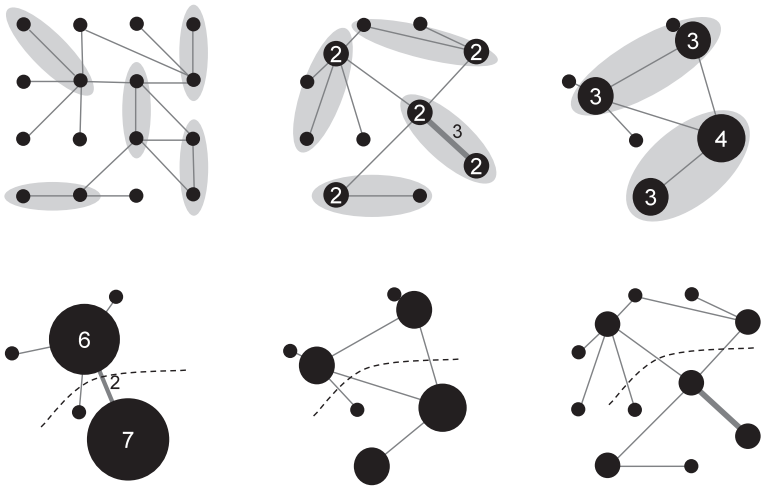


FIGURE 6.7: Coarsening and uncoarsening of graphs (source. `http://www.almob.org/content/9/1/12/figure/F4`). The first row shows the process of coarsening merging a few nodes at each step. The second row shows the process of uncoarsening after partitioning the coarsed graph.

There are many ways of doing coarsening of graphs. In the work by G. Karypis et al. [Karypis and Kumar, 1995], it is done by finding maximal matching and collapsing together the nodes that are incident on each edge of matching. A matching of a graph is a set of edges, in which no two edges are incident on the same node. Thus in this process not more than two edges are grouped together at one time. This matching can be done using many heuristic algorithms like random matching, heavy edge matching, light edge matching etc. [Buluç et al., 2013].

We suggest to use any community detection algorithm for the same purpose i.e. graph coarsening. We are hopeful that due to their more logical base of grouping the nodes in a

complex network, they can produce better condensed graphs. One successful attempt in this direction has been made by L. Wang et al. [Wang et al., 2014]. In this work authors propose a multi-level label propagation method for graph partitioning which is based on the use of label propagation method for community detection (See section 6.2).

In our case, we propose to use graph coarsening using communities for the purpose of link prediction. First any fast and efficient community detection method can be applied to the graph to identify groups of nodes in the form of communities which will form nodes in a compressed graph. The algorithm can have the following steps:

1. Apply a community detection algorithm on the large graph.
2. Merge the nodes in same community to form a single node in compressed graph.
3. Add edges in compressed graph based on the presence of edges between nodes belonging to different communities in the original graphs. These edges will have weights which is equal to total number of inter-community edges in original graph, for communities corresponding to nodes taken into consideration in compressed graphs.

This whole process of coarsening will reduce the size of the graph to work with and it will be easier to implement various standard link analysis methods. For using this graph in the task of link prediction, we have to keep in mind that huge compression of graph may not be very useful as it may lead to unwanted suppression of information. So it is essential to decide a minimum compression rate depending on the kind of network on which the process is to be implemented.

A more generalized approach will be to do graph coarsening using community detection along with diverse linking information available in a complex network in the form of multiplex networks. Our suggested algorithm has the following steps:

1. Apply any community detection algorithm on different layers of graphs representing a multiplex network.
2. Identify which sets of nodes are always together (that means always in the same community) in the different layers.
3. Binding such nodes together, represent them as single nodes to construct a compressed graph.
4. The compressed graph will have weights on edges, which are computed as the total number of edges between two groups of nodes in the original graph (one of the layers), and which represent single nodes in the compressed graph.

Another important point is that we are doing coarsening on only one layer of the network on which prediction of links is to be done (any analysis task) but using information from all three layers. This allows us to do coarsening on any layer of our choice using the same process.

We did a small experiment using the DBLP networks. We used two of community detection methods: Label propagation approach (LPA) [Raghavan et al., 2007] and Walktrap

[Pons and Latapy, 2006]. Following our multiplex coarsening method we found condensed graphs for co-authorship, co-citation and co-venue layers. The obtained result is presented in table 6.5 along with the ratio of nodes and edges in the coarsed and original graphs. It can be clearly seen that LPA has higher compression percentage for larger networks while Walktrap has a higher compression rate for comparatively smaller networks.

Use of coarsening for community detection is straightforward. It is easy to apply steps similar to multilevel graph partitioning. Community detection methods can be applied on the compressed graphs and uncoarsening will allow to obtain communities in the original graphs. The same can be applied to compressed graphs in a multiplex scenario. However the quality of communities found in the end will largely depend on the graph partition technique used in the initial partitioning step.

For application in the link prediction task also the graph coarsening can be very helpful to deal with the problem of computation and application of topological measures in very large networks. However there are many questions that need to be answered. How and where can the compressed graphs be used? A straightforward way is to use them for computation of topological measures on the compressed graphs. Then the question is how to use the values thus obtained to characterize the candidate node pairs belonging to original network, because the topological measures are computed to characterize nodes in compressed graphs (which represents groups of nodes from the original large graphs). Another option is to do a kind of link prediction on the compressed graphs. That means computation of topological measures and implementation of unsupervised or supervised link prediction approach will done on the compressed graph to find links between unlinked nodes in the compressed graph. These are actually probable inter-community links which can further be used to determine actual new or hidden links in the original graph. The interpretation of inter-community links found in the compressed graph to probable links in the original graph is another big question. In other words, it is not yet clear how to do a uncoarsening step taking advantage of the compressed graph for link prediction task. All these require a lot of work and provide an interesting direction for further research.

Community	Original Graph			Year	Coarsed Graph		Compression ratio	
	Layers	#Nodes (V)	# Edges (E)		#Nodes (v)	#Edges (e)	v / V	e / E
LPA	Co-authorship	91	116	1970-1973	63	76	0.6923	0.6552
	Co-venue	91	1256		63	821	0.6923	0.6537
	Co-citing	91	171		63	121	0.6923	0.7076
	Co-authorship	221	319	1972-1975	118	147	0.5339	0.4608
	Co-venue	221	5098		118	2573	0.5339	0.5047
	Co-citing	221	706		118	402	0.5339	0.5694
	Co-authorship	323	451	1974-1977	190	255	0.5882	0.5654
	Co-venue	323	9831		190	6025	0.5882	0.6128
	Co-citing	323	993		190	619	0.5882	0.6234
	Co-authorship	1371	2463	1980-1983	560	1010	0.4085	0.4101
	Co-venue	1371	83849		560	34538	0.4085	0.4119
	Co-citing	1371	13210		560	5305	0.4085	0.4015
Walktrap	Co-authorship	91	116	1970-1973	35	37	0.3846	0.3189
	Co-venue	91	1256		35	261	0.3846	0.2078
	Co-citing	91	171		35	55	0.3846	0.3216
	Co-authorship	221	319	1972-1975	106	126	0.4796	0.3949
	Co-venue	221	5098		106	1830	0.4796	0.3589
	Co-citing	221	706		106	228	0.4796	0.3229
	Co-authorship	323	451	1974-1977	144	172	0.4458	0.3814
	Co-venue	323	9831		144	2825	0.4458	0.2873
	Co-citing	323	993		144	333	0.4458	0.3353
	Co-authorship	1371	2463	1980-1983	844	1224	0.6156	0.4969
	Co-venue	1371	83849		844	38086	0.6156	0.4542
	Co-citing	1371	13210		844	3386	0.6156	0.2563

TABLE 6.5: Coarsening of graphs in different layers of a multiplex network

6.7 Conclusion

In this chapter we present how community detection algorithms can be used for under-sampling of negative examples in the task of link prediction. We provide a global overview of community detection task and different kinds of community detection algorithms available in existing scientific literature. We provide advantages and limitations of a few benchmark community detection methods. We then describe the utility of communities in the context of link prediction. To our knowledge a very limited attempts have been made to include community information for the benefit of link prediction. We describe a few such link prediction approaches in complex network that make use of communities. Then we describe the method of under-sampling which is a well known solution to class imbalance problem especially in supervised machine learning and how it has been used in the context of link prediction in real networks that often face the same problem. We present our proposed concept of under-sampling of negative example set using community detection algorithms. We suggest that two nodes of a negative examples that belong to the same community can be safely remove from the training set of examples resulting in a better learning model for prediction of new links in scientific collaboration network. The experimental results on DBLP co-authorship graphs justify our assumption and provide a strong base for further research in this direction. We would like to remind the readers that this work of using communities for prediction co-authorship links is based on some experimental observations in particularly scientific collaboration network of DBLP data. So we cannot say at this stage if the same concept will work for other kinds of networks like social interaction networks, biological networks, geographical networks etc. The behavior of communities and distribution of links across communities can change with different kinds of networks.

Chapter 7

Conclusion

Link prediction problem is not a new problem in information science and many methods have been proposed time to time to deal with this problem. However, new challenges have emerged with the continuous and rapid growth of complex networks. This report presents our research work on link prediction problem in dynamic large graphs. The main focus of our work is to analyze and predict links in bibliographical networks. We apply our methods to predict co-authorship links. We make detailed studies about the different existing link prediction approaches concentrating mainly on the topology guided approaches. We suggest a two fold classification of link prediction approaches as unsupervised, semi-supervised and supervised methods in one dimension and as dyadic, sub-graph based and global methods in another dimension.

In this work, we have proposed a novel approach based on supervised rank aggregation which has its roots in social choice theory. The approach is motivated by the belief that each attribute can provide some unique information which can be aggregated in the end to make a better prediction of future association between two unconnected entities in a network. First we have come up with a new way of introducing weights in a well known rank aggregation method. And secondly, we have proposed to apply this approach for the purpose of link prediction in complex networks. We have evaluated our approach on a co-authorship network obtained from DBLP database. The experimental results were quite encouraging as our method seemed to perform better than the approaches using classical machine learning algorithms for link prediction especially in terms of precision. The most promising factor of using a Kemeny aggregation based model is its ability to discard noise which has already been proved in the domain of meta-search engines [Dwork et al., 2001].

Next we have attempted to expand the scope of link prediction by adding heterogeneous link information in the form of multiplex networks. Multiplex networks are a subcategory of heterogeneous networks which have a layered structure. Each layer is a graph containing the nodes of the network but the edges in different layers are of different kinds. Simple topological measures are computed on different layers. There is a target layer on which prediction of link is to be made. The topological measures computed on the target layer are direct attributes and the same computed on all layers except the target layer are used as indirect attributes. An aggregate of the values of topological measures on all the layers can also be used as attributes. We have also proposed to use an entropy-based version of the topological measures which would take into account the existence of a non-zero value for a measure in all layers. The last two types of attributes are called

multiplex attributes. The application of these attributes for link prediction in multiplex networks obtained for DBLP dataset showed that the performance of supervised decision tree based model can be improved by including indirect and multiplex attributes.

Last but not the least we have tried to explore the utility of communities and community based information in the context of link prediction. In this report we have given a brief description of basic concept of communities and various community detection methods. We have also provided details about a few works in which attempts have been made to include community information as attributes in supervised link prediction model. We have used communities to filter out the list of candidate node pairs that were considered for prediction of new links but might not be very relevant for building a good model. This approach helped us to build a better prediction model in the presence of high class imbalance. We implemented this concept in a number of DBLP co-authorship networks and showed that the prediction result was better after filtering of instances is done based on communities. Moreover, the result for community based filtering was better than the classical random filtering or distance based filtering.

An additional work includes development of a new topological measure that can be used independently for unsupervised link prediction or as one of the attributes for supervised link prediction. This measure is named *path betweenness centrality*, and it computes the importance of a shortest path between two unlinked nodes in the form of a centrality that is similar to the well known edge betweenness centrality. It tries to advocate the fact that the importance of a shortest path between two nodes inside a network can reflect some information about the linking probability of the two nodes. We experimented this method for the prediction of co-authorship links in networks formed from DBLP data. The performance of this measure was not the best but was better than few other path based measures like truncated Katz and weighted shortest path length. So we believe that it can be useful for the tasks of complex network analysis including link prediction. Its major limitation is its computational complexity as it requires identification of shortest paths between every pair of nodes in the network. More work needs to be done to explore its real utility in the prediction task and other analysis tasks in complex networks.

7.1 Perspectives

We present below a few ideas and future directions for research built on the work presented in this thesis.

- **Reduce complexity of algorithms:** In large and ever growing complex networks, it becomes extremely important to take care of the complexity of the algorithms one is working on. In our case we have made a big attempt of trying to apply rank aggregation algorithms which have their own limitations in terms of time complexity. In order to avoid the limitations of Kemeny optimal aggregation(which is NP-hard), we have focused on the concept of approximate Kemeny aggregation. The use of merge sort has reduced the computational complexity to $O(rn \log n)$ where r is the number of experts and n is the number of candidates in each input ranked list. In order to enhance the performance of our approach and reduce the computational complexity further, we are interested to use the concept of $top-k$ aggregation, where instead of aggregating the complete input lists, only the $top-k$ from each are considered for aggregation. This will greatly reduce the number of

candidates involved in aggregation and thereby ease the computational process. However, this will require a very careful selection of candidates and may require aggregation of partial or even sometimes disjoint lists. In the work provided in [Kumar et al., 2009], the authors have given a vivid description of two well-studied approaches to achieve efficient $top-k$ aggregations namely: early-termination algorithms and pre-aggregation of some input lists. They propose generalized versions of *Threshold algorithm* (TA) and *No random-access algorithm* (NRA) using pre-aggregated intersection lists and they have shown its practical utility on large-scale data of web pages and search engine queries. In our future work, we intend to apply these algorithms to our approach. We are hopeful that it can greatly reduce the computational complexity caused due to application of rank aggregation methods.

- **Large network coarsening using communities:** Another perspective of this research directs towards using communities for coarsening of huge networks. This may provide another solution of find a way to apply traditional network analysis algorithms to large networks. Graph corsening (as discussed in chapter 6, section 6.6), requires grouping of nodes to build a compressed graph. Various criteria and methods can be implemented to find groups of nodes, one of which can be communities.In homogeneous networks, any fast and efficient community detection algorithm can be applied to a large network to identify groups of nodes as communities. These communities are then used as nodes in the coarsed graphs. Edges are added based on the presence of edges between nodes belonging to different communities in the original large graphs. Weights can be added to these edges according to the numbers of inter-community links in original graph. For heterogeneous network, in particular multiplex networks, the nodes of coarsed graph are found by applying community detection algorithm to all layers of original network and then identifying the sets of nodes that always belong to the same communities. Once the coarsed graphs are found, they can be used for various network analysis tasks including link prediction. However, there remains many questions to be answered regarding how these coarsed graphs can be applicable in different tasks including link prediction. All these needs more studies and provides an interesting direction of future research. More discussion on application of coarsed graphs in the context link prediction can be found in section 6.6 of chapter 6.

- **Choice of efficient topological attributes:** The choice of topological measures used as attributes in a supervised link prediction task is very crucial. The individual capability of different measures to identify a true positive link can greatly affect the final performance of the supervised prediction model developed using them. Moreover the individual performance of different topological measures differs with types of network on which it is implemented. Also, not all measures perform highly well all the time [Liben-Nowell and Kleinberg, 2007]. So proper selection of topological measures is very essential. However, it is often very difficult to know which measures are going to perform well without any experimentation. So, it is better to implement some kind of selection procedure in the prediction approach which will filter out the not-so-useful topological measures on the way of building the prediction model. One such implementation has been done in [Bao et al., 2013] using principal component analysis. Correlation between various measures can be used to select the superior ones. We think that if such a selection method can be implemented in our proposed supervised rank aggregation model, it will really lessen the complexity of computation and also help in having a better final prediction list. In the present case, in the supervised Kemeny based method we do

a kind of filtering based on the weights of attributes obtained during the learning phase. We leave all attributes that have zero weight. However, we think more work needs to be done in this direction.

In our work on link prediction in multiplex networks, much work can be done for implementation of better topological measures. Apart from applying the above mentioned way, more different kinds of multiplex attributes can be explored based on the recent advances in analysis of multiplex networks. For example we can take into account the inter-layer links which has not been done in our approach. Also we have applied very naive ways like average of the values of topological measures found in different layers. Instead of that some more complex measures can be used which will include the multiplex information in a much better way. We made an attempt with the entropy based versions of topological measures but they do not perform well always. Other such measures can also be explored in the context of link prediction.

Appendix A

LiPTaR : Link Predicton based Tag Recommendation for Folksonomy

A.1 Introduction

Social tagging sites such as Delicious (for web site sharing), CiteULike and Bibsonomy (two sites for sharing bibliographical data) have become a major tools of sharing resources on the Web. In such sites, called also broad Folksonomies, users annotate resources (new or existing ones) by a set of user-defined words called *tags*. The most important feature of a folksonomy that makes it different from any other resource sharing networks, is the freedom given to the users to select their own tags for annotating resources. This feature gives an advantage of eased cost factor but at the same time, leads to various problems. One key issue to handle is the *tag ambiguity* problem. It refers to the condition of having the same tag being used to index different resources by different users or even by the same the user but at different points of time. It may also refer to a condition where similar resources can be indexed by different tags by different users. This witnessed phenomena limits the utility of tags as a means for sharing new resources. One widely studied approach to cope up with this problem is tag recommendation.

Different approaches for tag recommendation computation has been proposed in the scientific literature. Some make use of resources contents [Mrosek et al., 2009]. Others relay mainly analyzing the topological features of the graph induced from the ternary relation linking users to resources the annotate [Jäschke et al., 2008]. However, according to our knowledge, no prior work has proposed to mine the evolution of the folksonomy graph in order to compute appropriate tags to recommend.

In this work, we describe a new approach for tag recommendation that we call `LipTaR`. The original idea of the approach is to mine the dynamic of the tagging activity in order to compute the most suitable tag for a given user and a given resource. The tagging history of each user is modeled by a temporal sequence of bipartite graphs linking tags to resources. Given a target user and a target resource, we first compute a set of similar users. The tagging history of the identified set of users is merged in one temporal sequence on bipartite graphs. The obtained sequence is used to learn a model of link prediction in bipartite graphs. The learned model is then applied to predict tags to be linked to the target resource and a list of top similar resources.

A.2 Related work

Various approaches have been proposed for tag recommendation which can be broadly categorized as:

- Content-based approaches involve extraction of tags from the content of the resources or titles of the resources. They are efficient in recommending very relevant tags but without considering user's choice. These methods cannot be efficient when the resources do not provide a rich content of information.
- Topology-based approaches find tags for recommendation by analyzing the graphical structures linking users, tags and resources. In this case the recommended tags are mostly those, that has already been used in the system. They can prove to be very efficient in cases without resource contents.

A content-based approach has been proposed in [Mrosek et al., 2009] where recommended tags are generated from the content of the resources. Individual scores are calculated based on different information provided by each resource and then an aggregated global score is found for each tag. Tags with top five highest scores are selected for recommendation. Another content-based approach is proposed in [Lu et al., 2009]. The proposed approach is based on the observation that similar web pages usually have same tags. So, each web page can share tags with similar ones. The propagation of a tag depends on its weight in the originating web page and the similarity between the sending and receiving web pages. The similarity metric between two web pages is defined as a linear combination of four types of cosine similarities, taking into account both tag information and page content. In [Lipczak, 2009] authors propose yet another content-based approach where recommendation computation is made in a three-step process: tags are first extracted from resource titles. The set of potential recommendations is then extended by related tags proposed by a lexicon based on concurrences of tags within the resource posts. In the third and final step tags are filtered by user's *personomy*: a set of tags previously used by the user.

In the category of topology-based approaches, one of the prominent work is given in [Jäschke et al., 2008]. Here, authors compare a number of recommendation techniques like collaborative filtering, *PageRank* and its modified version for folksonomy known as *FolkRank*. They show that the *FolkRank* based recommender outperforms the other two approaches. They propose two tag recommendation algorithms: an adaptation of user-based collaborative filtering and a graph based recommender built on the top of *FolkRank*. Tests were performed on the dense core of folksonomy, so it may not be very representative. Moreover, they do not take into account the dynamic nature of a folksonomy. Another work is given in [Zhang et al., 2009] in which authors propose a tag recommendation algorithm based on an integrated diffusion on user-item-tag tripartite graphs. Authors propose an algorithm using both user-resource relations and the collaborative tagging information. They emphasize on the fact that two resources, sharing many common tags, have greater probability of being closely related in content. They conclude that the use of tag information can significantly improve the accuracy, diversification and novelty of recommendation. Another graph-based method is FolkDiffusion [Liu et al., 2010] which uses the concept of heat diffusion to rank tags. This method can suggest user and resource specific tags without having topic drift. It uses a graph having users, resources and tags with edge weights representing the relatedness between them.

It uses the concept of physical phenomenon of flow of heat from high to low temperature. The user and resource for which tag suggestion is to be made are given a temperature more than zero. All other tags, users and resources are given a temperature equal to zero. The heat is then assumed to flow from target user and target resource to all other nodes according to the edges between them. After a certain number of iterations the heat value on tags show their relatedness to target user and target resource and are accordingly selected for recommendation.

A.3 LiPTaR system

Our system called LiPTaR is based on link prediction in folksonomy graphs. The system takes as input a target user u_t and a target resource r_t. The goal is to compute a list of tags best suited for the user u_t to annotate resource r_t.

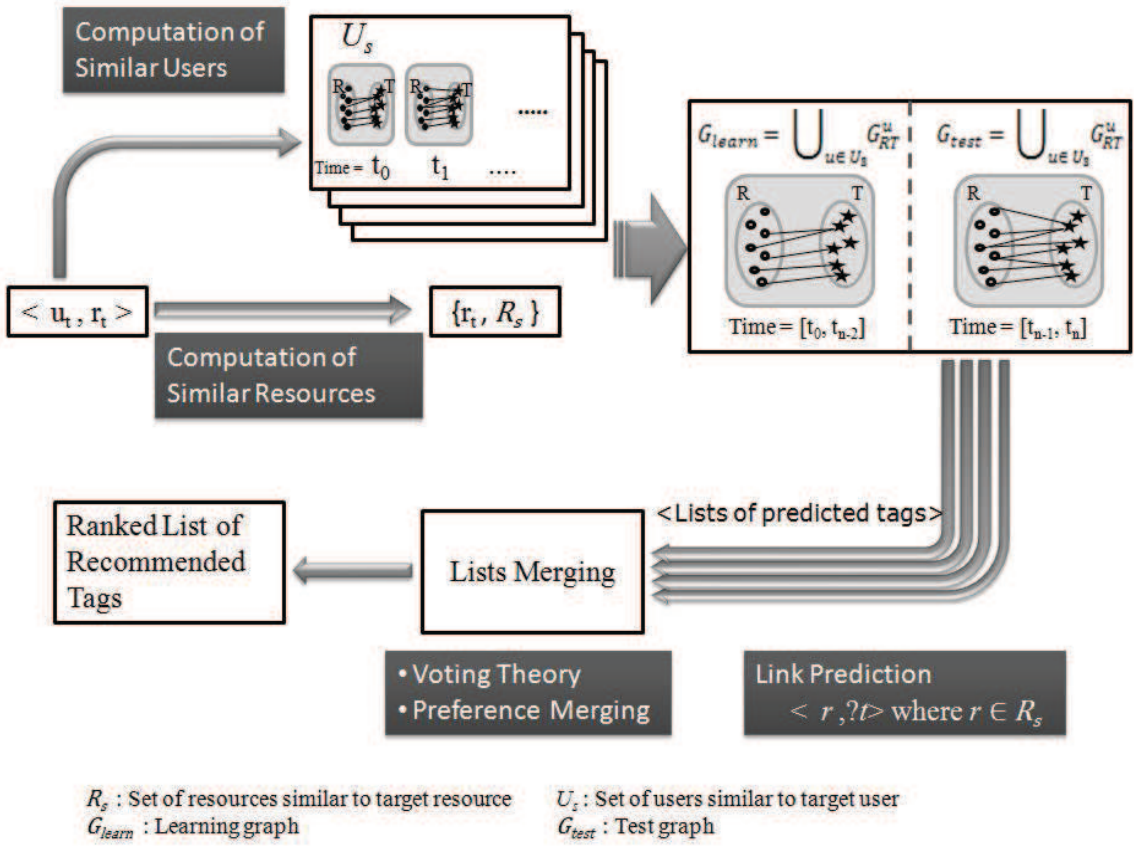


FIGURE A.1: LiPTaR work cycle

Fig. A.1 illustrates the general outlines of the tag recommendation cycle applied in the LiPTaR system. The cycle is structured in three steps :

1. First, the system computes a set of k most similar users $\mathcal{U}_s$ based on their *similarity* to u_t. Many user similarity metrics can be used for this purpose. In the current prototype the top k similar users have been found by application of k-nearest neighbors with a similarity metrics based on both resources and tags used by a

user. Another important aspect of this system is that while computing similarity, it takes into account the users's time of activity. So the similar users found have at least one year of activity time common with the target user u_t. Here we explore the idea that users active during same period of time may have common interests and choices.

2. Each user $u \in \mathcal{U}_s$ is associated with a sequence of temporal bipartite graphs relating resources added by user u to tags used by him at various point of time. These graphs are combined to create a single resource-tag bipartite graph for training(G_{learn}). During this process, only the graphs corresponding to a time within the duration of training period are used. So, $G_{learn} = \bigcup_{u \in \mathcal{U}_s} \bigcup_{i=t_0}^{t_{learn}} G_i$.

   Similarly, G_{label} and G_{test} are also generated to be used for labeling of examples and validation correspondingly. A couple of nodes (resource-tag pair) that are not linked in G_{learn} but both belonging to the same connected component represent an example (in terms of supervised learning convention). For each such couple of nodes, we compute a set of topological attributes that characterize their roles in the network as well as their *similarity*. The class label for them is obtained by checking whether the couple of nodes is indeed connected in G_{label}. If such a connection is found then it is labelled positive in the supervised learning task and if not it is labelled negative. All the training examples thus found are used by a supervised machine learning algorithm to generate a classification model. This model is then used to predict links in the validation graph G_{test} in order to find probable links between target resource r_t and different tags during validation time period. It does the same for each of the similar resources. At this point, we make an assumption that the tags used by the similar users, for resources that are somehow similar to the target resource, can also be useful for recommendation. In the end, we obtain one or more lists of tags for annotating the resource r_t and other similar resources.

3. At the end of step 2, we get one or more ranked lists of tags, obtained for r_t and/or a set of similar resources using the data related to retrieved similar users.These lists include both already used tags and predicted tags. We apply a suitable ranked list aggregation approach [Dwork et al., 2001] to merge these lists.

To sum up, the LiPTaR approach is conceived as a framework offering three main hotspots to be adapted for research: a) the user and resource similarity metrics, b) the link prediction approach to be applied to infer tags for recommendation, from the point of view of each retrieved similar user and c) the rank aggregation method to be applied to merge all obtained list of tags computed in step b).

A.4 Experiment

We experimented our system on data extracted from website CiteULike[1] which is a bibliographic reference sharing website. Like any other folksonomy, users can share their resources with other users and annotate them using their own tags. The dataset covers a time period from year 2004 to year 2010. The total number of data entries are $10,504,915$. After pre-processing we get a tripartite graph with $71,464$ users, $2,402,913$

[1] http://www.citeulike.org/

resources and 489,682 tags. We use only meaningful tags, discarding the system generated ones. We found that there are 397,252 resources without a tag which counts for 16.53% of total resources.

The inputs for our tag recommendation system are a user (target user) and a resource (target resource).We take a combination of Jaccard's similarity coefficient based on both tags and resources for computing similarity between users. As mentioned before, these users also have some common time of activity. We make use of a modified version of link prediction approach proposed in [Benchettara et al., 2010b] for prediction of new links in the bipartite graph linking resources and tags used by top-k similar users. For computation of resource similarity we use the same Jaccard's coefficient but only based on tags.

At present we are using the following topological measures: product of coefficient of clustering, product of degree centrality, preferential attachment [Barabási et al., 2002], an indirect computation of number of common neighbors with respect to tag and with respect to resource, shortest path length, measure of Adamic Adar [Adamic et al., 2003]. Finally we use local Kemeny optimal method [Dwork et al., 2001] for list merging which gives us an optimized aggregation and is computationally efficient.

We use data from period of $2005 - 2007$ for training the link prediction model. We use a boosted decision tree classifier (the Adaboost using $J48$ classifier) in the Orange[2] platform. Validation is done on examples constructed from data of period 2006-2008 to predict the tags used in period 2009. (We discarded the data of 2004 and 2010 as they were not complete.)

The performance of the system is measured in terms of precision, recall and F-measure. We experimented on 31 users and a varying number of resources for each of them. The average precision is found to be approximately 0.01345, average recall is 0.385 and average F-measure is 0.02326. The average precision and F-measure may seem to be very less, but low values are due to the fact that for the moment we have not restricted the number of predicted tags to be used for recommendation. The result is also affected by the data sparsity of the large scale dataset we are using.

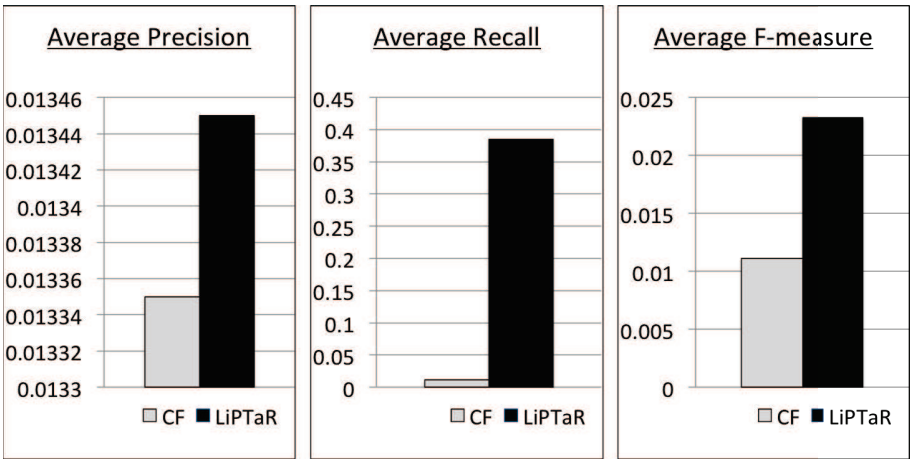


FIGURE A.2: Preliminary Results

[2]http://orange.biolab.si/

To make a comparison with a classical approach, we experimented with a basic method of tag-based collaborative filtering. The input graph is the union of the temporal Resource-Tag graphs for top k similar users.We make a prediction of tags for resources used by the target user in validation period of 2009. This prediction is made on the basis of target user's history and the choice of similar users. Using this approach, for the same number of target users and target resources, the average precision is found to be approximately 0.01335, average recall is 0.011 and average F-measure is 0.01113. Fig. A.2 shows a comparison between the two approaches. Our approach seems to give a better result as compared to collaborative filtering method which encourages us to continue our experiment further.

A.5 Conclusion

In this chapter, we present a new approach of tag recommendation in folksonomy based on a method for link prediction in the bipartite graphs. The approach includes decomposition of a tripartite graph representing a folksonomy, into three bipartite graphs. The proposed approach is implemented as a framework structured around three main hotspots: 1) the similarity metrics to be used for retrieving similar users and similar resources, 2) the link prediction approach to apply and 3) the ranked list merging method to use. Results obtained from applying first implementation of this framework to a real world dataset extracted from a broad folksonomy (where tagging is oriented towards sharing resource within a community) argue for the validity of the approach. Further experiments are required in order to evaluate effects of using more elaborate similarity metrics for retrieving similar users and similar resources. Evaluating different rank aggregation results as well as applying the framework to different types of folksonomies including narrow ones (where tagging is mainly motivated by personal usage).

Appendix B

Path Betweenness Centrality

We propose a new measure named *path betweenness centrality*, which will compute the centrality of a shortest path between any two nodes in a graph with respect to all shortest paths of the graph. That means it counts how many of the shortest paths in the graph contain a shortest path between two nodes under observation. This measure will try to find the importance of a shortest path between two nodes in linking any other nodes in a graph.

We experiment to find the utility of this measure in the context of link prediction. We apply it to datasets derived from co-authorship networks using the real data of DBLP. Next we formally present path betweenness centrality and briefly explain how it can be used to predict new links .

B.0.1 Path betweenness centrality

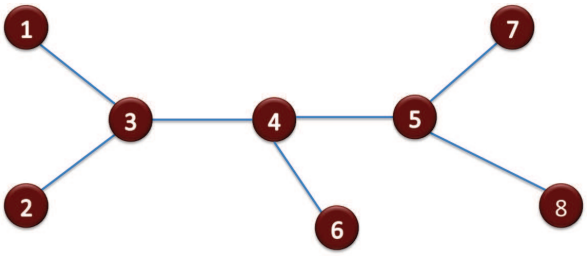
Let $G = (V, E)$ be a network, with V is the set of nodes and E is the set of edges. Let $paths(u, v)$ be the set of shortest paths between nodes u and v. We say that $nsp(u, v) = |paths(u, v)|$ is the number of shortest paths and $dist(u, v)$ is the shortest path length. The betweenness centrality for a path $p \in paths(u, v)$ is defined as :

$$c_B(p) = \sum_{s,t \in V and\ (s,t) \neq (u,v)} \frac{nsp(s, t \mid p)}{nsp(s, t)} \tag{B.1}$$

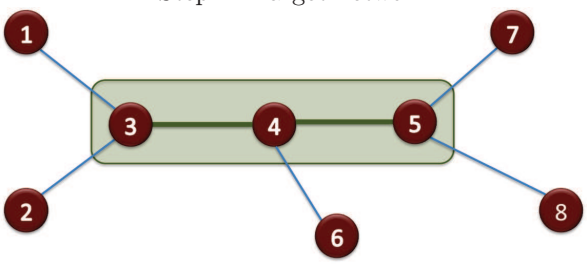
$nsp(s, t \mid p)$ is the number of shortest paths between s and t passing through path p. If the number of shortest paths between two nodes is more than one, then the path betweenness centrality of the pair of nodes is the maximum of the centralities found for all the shortest paths between them. Another way is to apply the average, sum or any other suitable function to aggregate these multiple centrality. But for the time being we will study maximum.

The basic idea underlying this measure is to weight shortest paths in function of there degree of inclusion in other shortest paths in the network. Figure B.1 illustrates computation of path betweenness centrality on a sample graph.

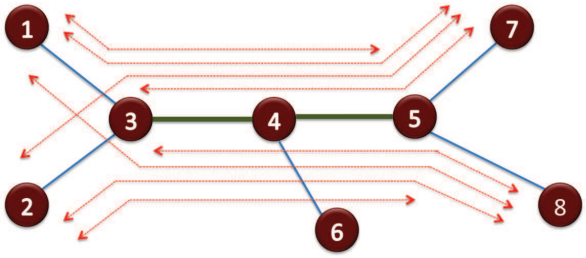
This measure can be used for predicting new links by assuming that the more central a shortest path between two nodes is, the more are the chances of having a direct connection



Step 1: Target network



Step 2: Node pair under consideration are $(3, 5)$. The green color highlights the path for which betweenness centrality is to be found.



Step 3: All shortest paths to be considered to find a path betweenness centrality. Numbers of shortest paths containing the shortest path between node 3 and node 5 is 8 and centrality of the path comes to be PBC(3,5) = 8.0. If normalized by dividing it by number of all shortest paths, the value becomes PBC(3,5) = 0.2963

FIGURE B.1: An example to find the betweenness centrality of a shortest path between a pair of nodes

between them at some point of time in future. Our link prediction approach is similar to that of work of [Liben-Nowell and Kleinberg, 2007]. We rank all pairs of unlinked nodes at a certain time t, based on their path betweenness centrality value. The k-top ranked node pairs are considered to be the predicted new edges at time say $t + 1$. Thereby the performance of path betweenness centrality is measured.

B.0.2 Experiment

We did experiments on networks generated from DBLP[1] data. Various datasets were derived corresponding to different periods of time. Due to its computational complexity and for a justified comparison between node pairs, we decided to use the largest connected components of the networks. Hence, we use data between years 1970-1979 and 1980-1983. We created six different datasets from this and computed the unlinked node pairs (We call them *examples* as in machine learning. See section 3.3.2.1) to be used for validation.

[1] http://www.dblp.org

The different graphs, datasets and numbers of positive and negative examples obtained have been listed in table B.1 and B.2.

Graphs	***V***	***E***	***Density***
1970-1973	91	116	0.0283
1972-1975	221	319	0.0131
1974-1977	323	451	0.0087
1980	147	232	0.0216
1981	241	407	0.0141
1982	202	310	0.0153

TABLE B.1: Co-authorship graphs used for generation of examples

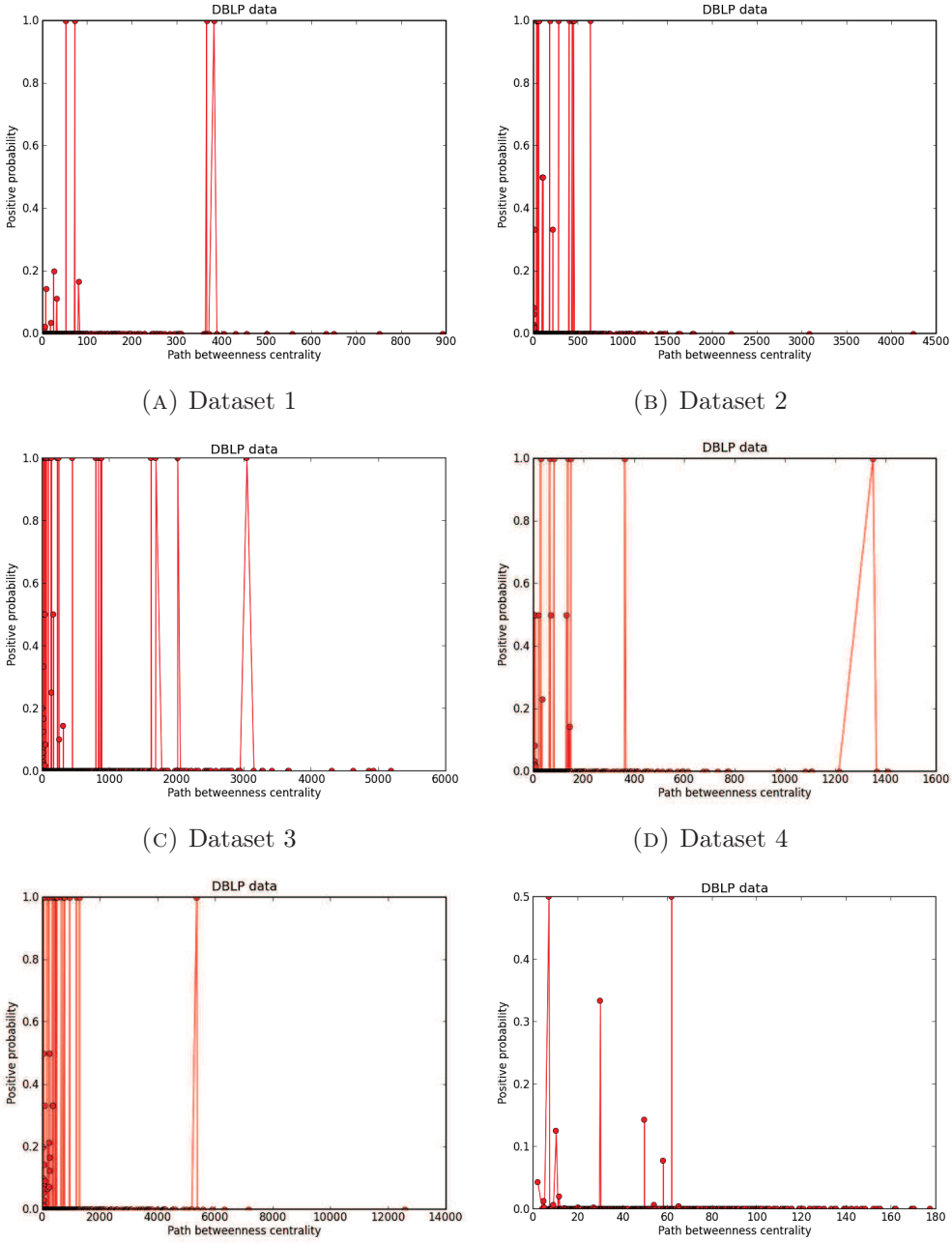
	Learning	**Labeling**	**# Positive**	**# Negative**
Dataset 1	1970-1973	1974-1975	16	1810
Dataset 2	1972-1975	1976-1977	49	12141
Dataset 3	1974-1977	1978-1979	93	26223
Dataset 4	1980	1981	39	5515
Dataset 5	1981	1982	67	13411
Dataset 6	1982	1983	37	9069

TABLE B.2: Examples from largest connected component of co-authorship graphs

We implement an unsupervised prediction method (see section 3.3.1) to find the performance of path betweenness centrality in link prediction. Further we compare it with a few of path based, neighborhood based and node's feature based measures as done in [Liben-Nowell and Kleinberg, 2007] (see section 3.3.1). The different neighborhood based measures used are: Jaccard's coefficient (JC), Neighbor's clustering coefficient(NFC), Common neighbors (CN), Adamic Adar coefficient (AA), Resource allocation (RA); one node feature based measure: Preferential attachment (PA); and path based measures: shortest path length (SPL), truncated Katz coefficient ($TKatz$), and a weighted form of shortest path length ($WSPL$).

Before going to the actual evaluation, we really wanted to understand how exactly path betweenness centrality functions when it comes to prediction of positive links. So we computed the probability of getting positive examples by varying the values of path betweenness centrality in out DBLP datasets in order to verify if our assumption is justified.

For all the datasets we found that the probability of getting positive examples is much higher with lower values of path betweenness centrality. This is contrary to our initial assumption as the lower values of the topological measure are able to identify more positive examples. On explanation for this situation can be that when two unlinked nodes have a shortest path with a high betweeness centrality, it means that they have a shortest path which is more often passed to travel between any other two nodes in the network. This also indicates to the fact that the two nodes may have a higher popularity (in terms of degree or centrality) in their own respective local circles. For example, they may have the highest degree within their respective groups of direct neighbors. And due to which they do not feel any need to connect directly with each other during immediate future. Another example can be that of authors of scientific papers in a co-authorship



(A) Dataset 1

(B) Dataset 2

(C) Dataset 3

(D) Dataset 4

(E) Dataset 5

(F) Dataset 6

FIGURE B.2: Positive probability of path betweenness centrality

network (as it is in our case). If two authors are already highly connected or atleast are highly connected in the local network of their direct neighbors, they can have a shortest path with high centrality. But in spite of that, having high importance in their local circles, they may not collaborate with each other, at least in immediate future. This is a similar scene as shown by preferential attachment also when used for link prediction. Hence here onwards we will assume that a lower path betweenness centrality of shortest path with cause higher linking probability and accordingly ranking of examples is done.

Different topological measures including *path betweenness centrality* (PBC) are computed on the corresponding graphs and examples (unlinked node pairs) are ranked based

on the values. Then the top-k ranked pairs are taken as predicted new links and prediction result is computed in terms of *precision* and *AUC*. AUC is computed using the formula given in section 3.2.1 but the difference is that instead of exact score we compare the ranks of negative and positive examples. So actually, we compute the probability of finding a positive example ranked above a negative example in the list of prediction which is the top-k ranked list. Also we are unable to take into account the equal ranks between negative and positive examples, as we are not treating ties for the moment. Ties are broken randomly whenever they appear in the score and dealing with ties during ranking can be added to future updates of our method. The value of k for each dataset is taken to be the number of actual new links. Tables B.3, B.4,B.5, B.6, B.7 and B.8 summarize the results we obtained in terms of precision.

Looking at result in terms of precision, path betweenness centrality did not seem to be working absolutely well for link prediction, although for some of the datasets it performs better than truncated Katz, neighbor's clustering coefficient, weighted shortest path length and preferential attachment. However, in terms of AUC it is always performing well, thereby justifying its capacity to rank a positive example above all negative examples classified during the prediction process. Hence, this can be a motivation for further experimentation on other networks to see its practical utility. Also, its relevance in other complex network analysis tasks like community detection and influential node identification can be explored.

Topological measures		Precision	AUC
Neighborhood based	JC	0.0625	0.8667
	NCF	0.0	0.0
	CN	0.3125	0.2727
	AA	0.125	0.0714
	RA	0.125	0.0714
Node's feature based	PA	0.0	0.0
Path based	PBC	0.1875	1.0
	$TKatz$	0.0	0.0
	$WSPL$	0.0	0.0
	SPL	0.3125	1.0

Table B.3: Results of prediction on Dataset 1 (k = 16)

Topological measures		Precision	AUC
Neighborhood based	JC	0.1633	0.7104
	NCF	0.1633	0.3323
	CN	0.4898	0.25
	AA	0.1837	0.5611
	RA	0.1633	0.5152
Node's feature based	PA	0.0204	0.5833
Path based	PBC	0.0612	1.0
	$TKatz$	0.1224	0.2597
	$WSPL$	0.0	0.0
	SPL	0.4898	1.0

Table B.4: Results of prediction on Dataset 2 (k = 49)

Topological measures		Precision	AUC
Neighborhood based	JC	0.0645	0.6322
	NCF	0.0323	0.5333
	CN	0.2688	0.4141
	AA	0.0645	0.4962
	RA	0.0538	0.5250
Node's feature based	PA	0.0107	0.5652
Path based	PBC	0.0746	1.0
	$TKatz$	0.0323	0.4778
	$WSPL$	0.0107	0.1848
	SPL	0.2688	1.0

Table B.5: Results of prediction on Dataset 3 (k = 93)

Topological measures		Precision	AUC
Neighborhood based	JC	0.0769	0.6204
	NCF	0.0513	0.4865
	CN	0.3846	0.4167
	AA	0.2051	0.3226
	RA	0.1795	0.7813
Node's feature based	PA	0.0	0.0
Path based	PBC	0.0746	1.0
	$TKatz$	0.0256	0.2632
	$WSPL$	0.0513	0.0
	SPL	0.3846	1.0

TABLE B.6: Results of prediction on Dataset 4 (k = 39)

Topological measures		Precision	AUC
Neighborhood based	JC	0.0298	0.9077
	NCF	0.0448	0.4479
	CN	0.2687	0.0555
	AA	0.0448	0.6198
	RA	0.1045	0.3119
Node's feature based	PA	0.0149	0.6364
Path based	PBC	0.1026	1.0
	$TKatz$	0.0448	0.2135
	$WSPL$	0.0298	0.9385
	SPL	0.2985	1.0

TABLE B.7: Results of prediction on Dataset 5 (k = 67)

Topological measures		Precision	AUC
Neighborhood based	JC	0.1622	0.7742
	NCF	0.0811	0.5294
	CN	0.2973	0.0909
	AA	0.0811	0.7255
	RA	0.0811	0.8824
Node's feature based	PA	0.0270	0.0555
Path based	PBC	0.0746	1.0
	$TKatz$	0.1081	0.2879
	$WSPL$	0.0540	0.7714
	SPL	0.4595	1.0

TABLE B.8: Results of prediction on Dataset 6 (k = 37)

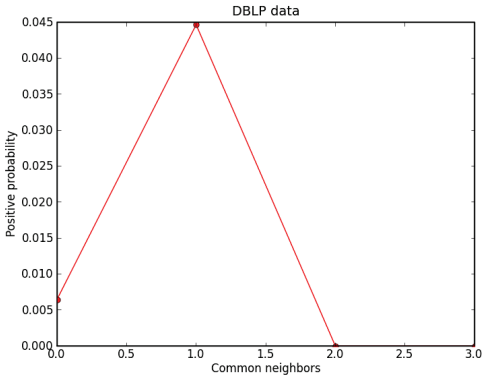
Appendix C

Performance of Topological Measures

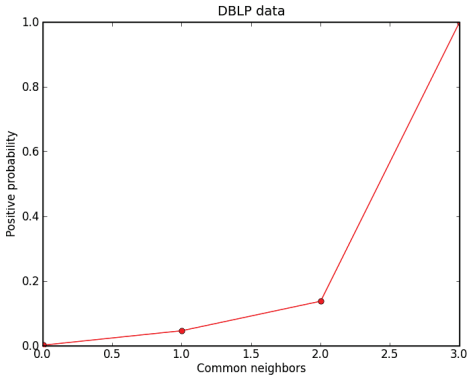
We summarize here the probability of getting a true positive links with increasing values of different topological measures. These values have been computed on six different co-authorship networks created from DBLP data. Table C.1 shows the different datasets used, the corresponding years, and the number of examples in them.

	Learning	**Labeling**	**# Positive**	**# Negative**
Dataset 1	1970-1973	1974-1975	16	1810
Dataset 2	1972-1975	1976-1977	49	12141
Dataset 3	1974-1977	1978-1979	93	26223
Dataset 4	1980	1981	39	5515
Dataset 5	1981	1982	67	13411
Dataset 6	1982	1983	37	9069

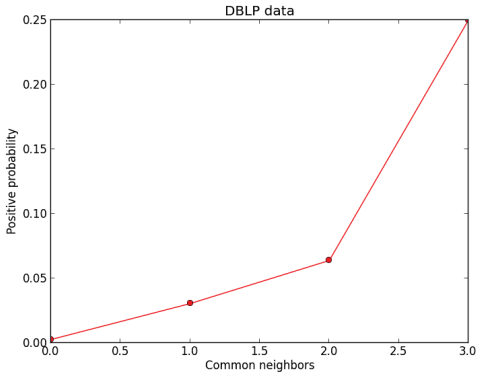
TABLE C.1: Examples from largest connected component of co-authorship graphs



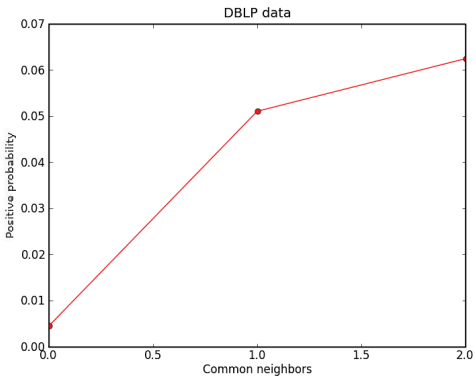
(A) Dataset 1



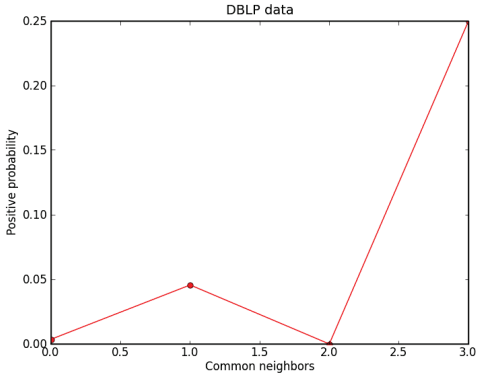
(B) Dataset 2



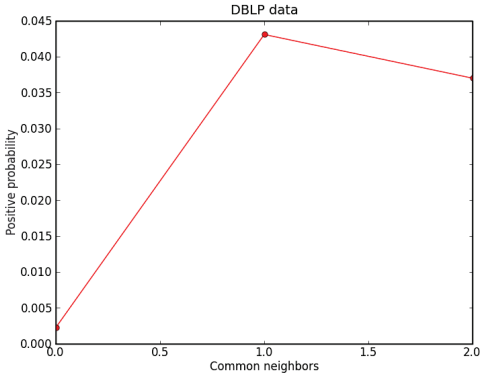
(C) Dataset 3



(D) Dataset 4

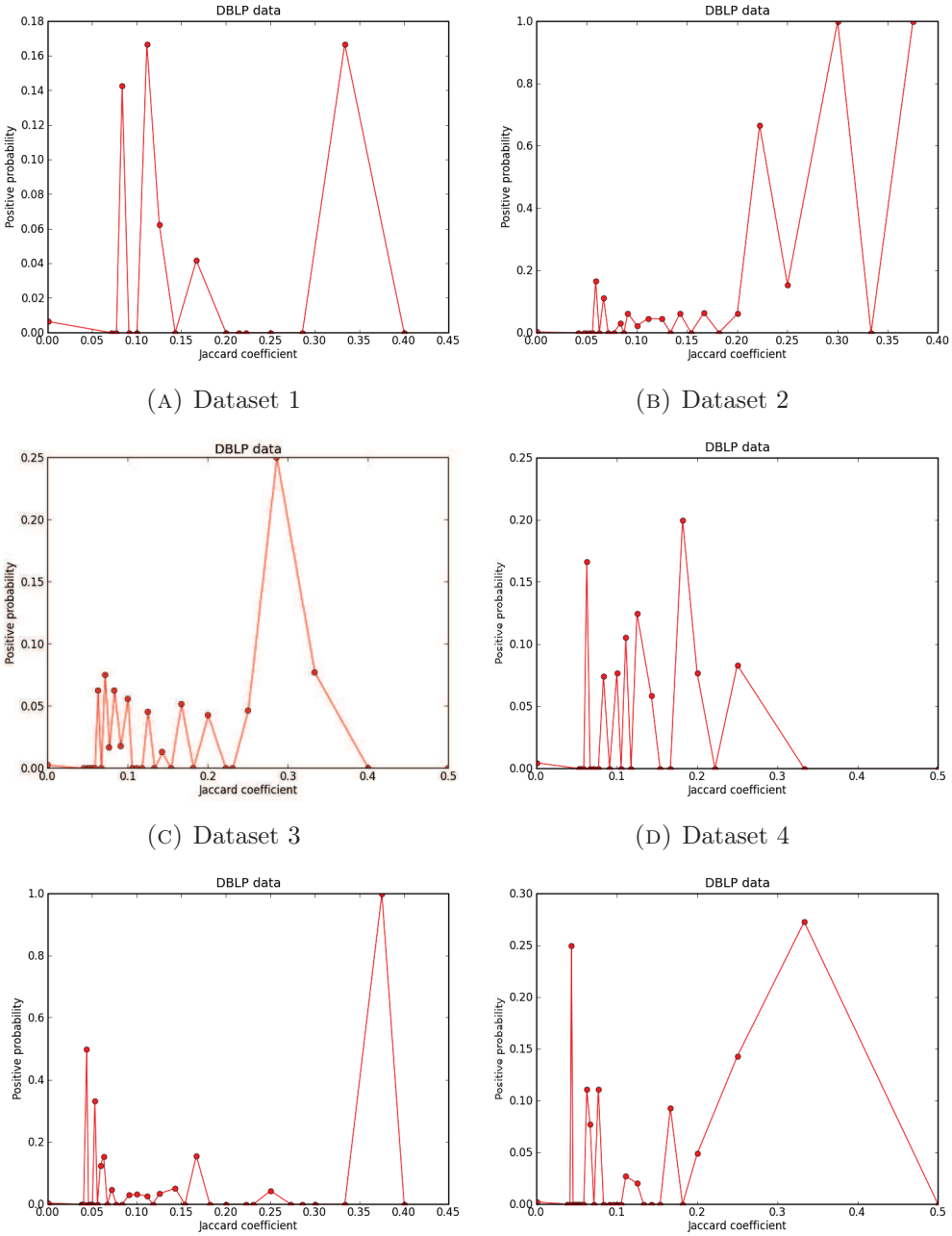


(E) Dataset 5



(F) Dataset 6

FIGURE C.1: Positive probability of number of common neighbors



(A) Dataset 1

(B) Dataset 2

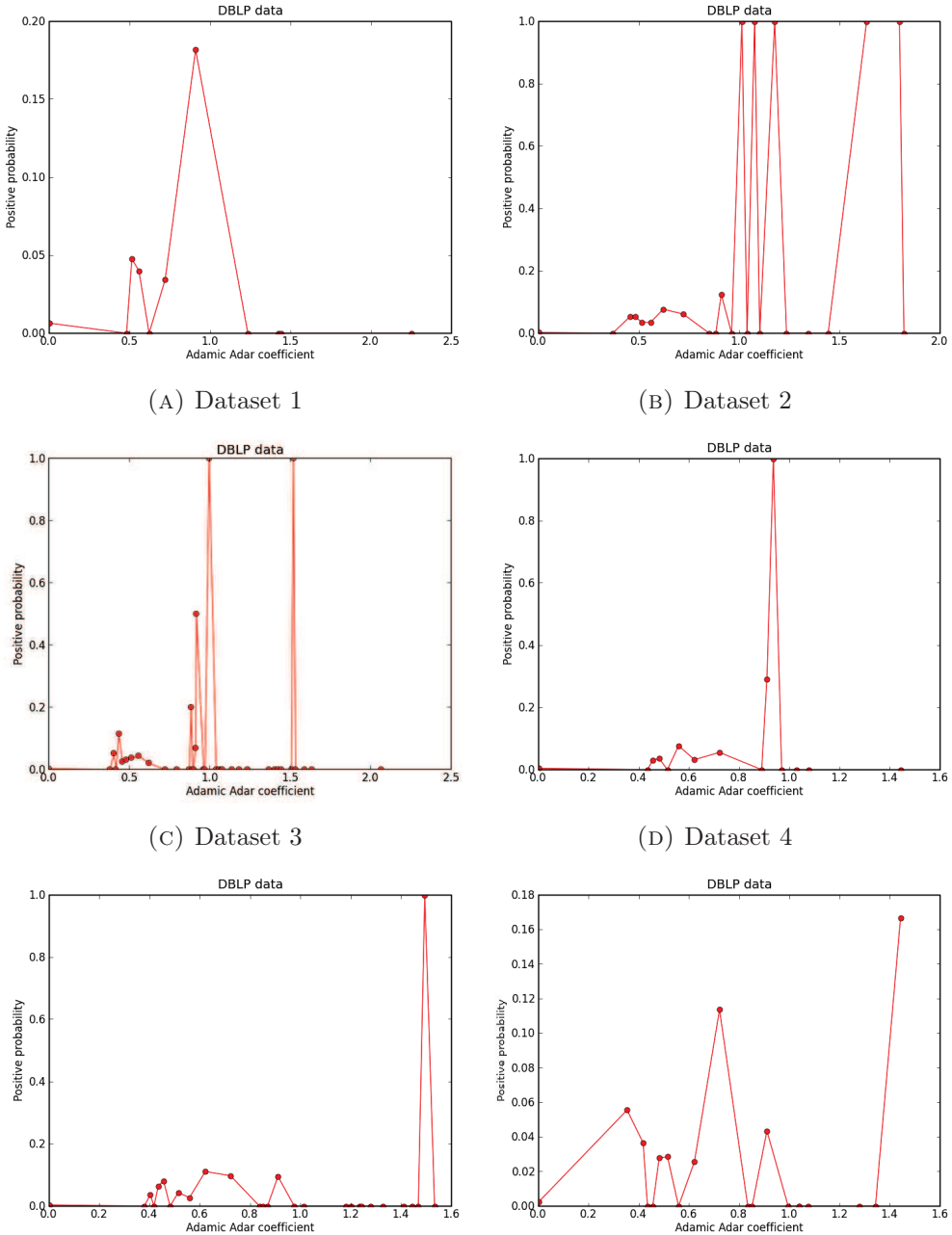
(C) Dataset 3

(D) Dataset 4

(E) Dataset 5

(F) Dataset 6

FIGURE C.2: Positive probability of path Jaccard's coefficient



(A) Dataset 1

(B) Dataset 2

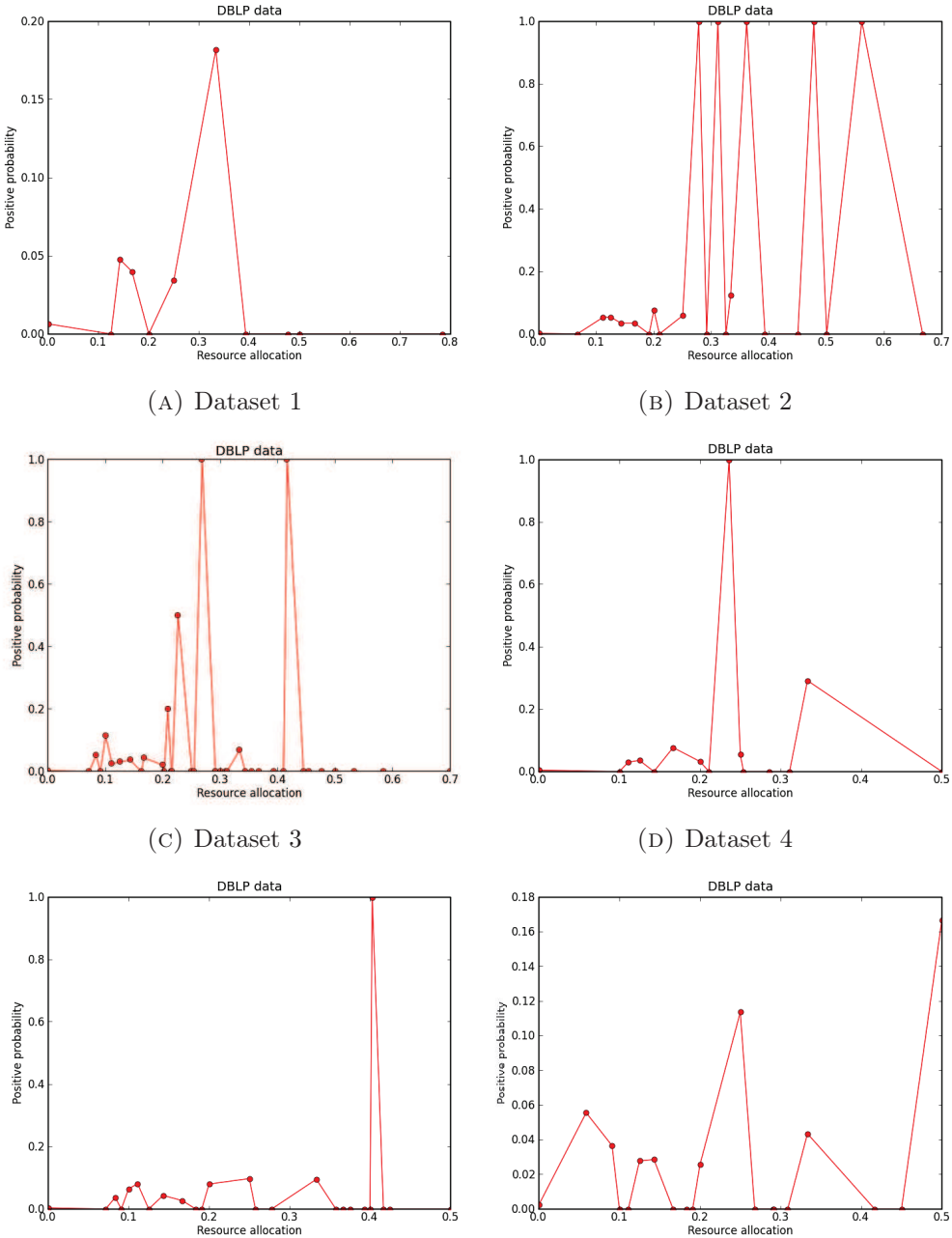
(C) Dataset 3

(D) Dataset 4

(E) Dataset 5

(F) Dataset 6

FIGURE C.3: Positive probability of path Adamic Adar coefficient



(A) Dataset 1

(B) Dataset 2

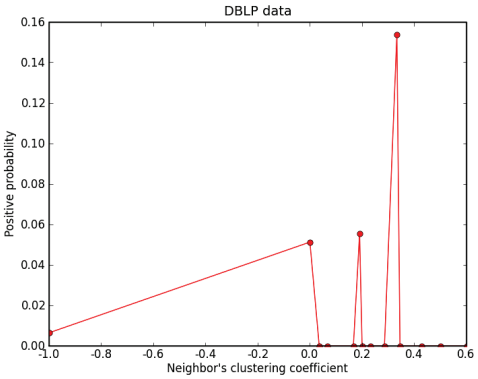
(C) Dataset 3

(D) Dataset 4

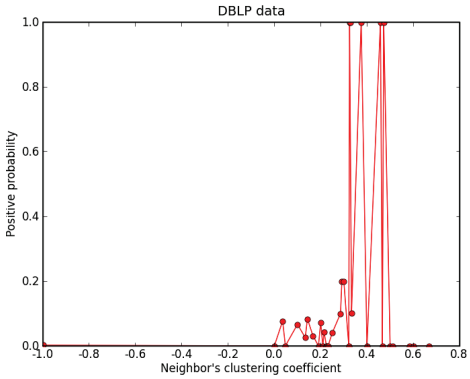
(E) Dataset 5

(F) Dataset 6

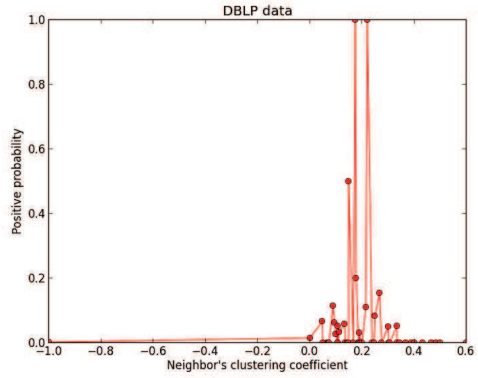
FIGURE C.4: Positive probability of resource allocation



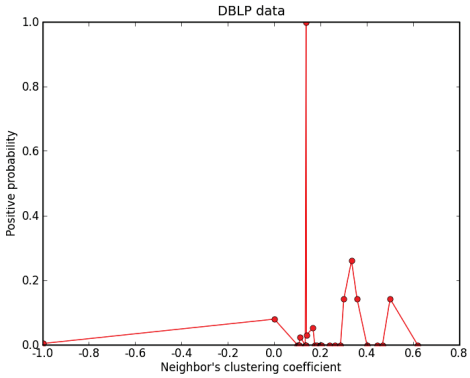
(A) Dataset 1



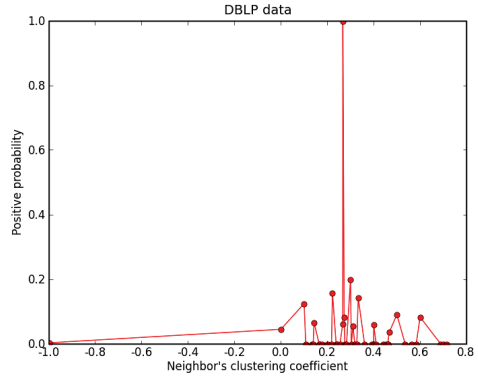
(B) Dataset 2



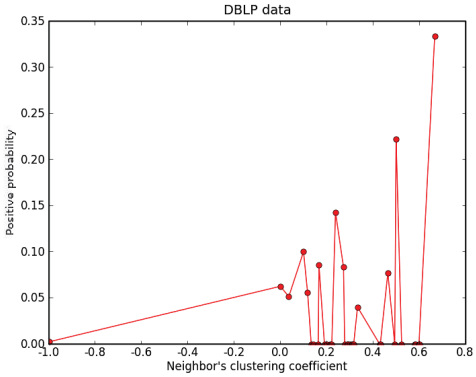
(C) Dataset 3



(D) Dataset 4

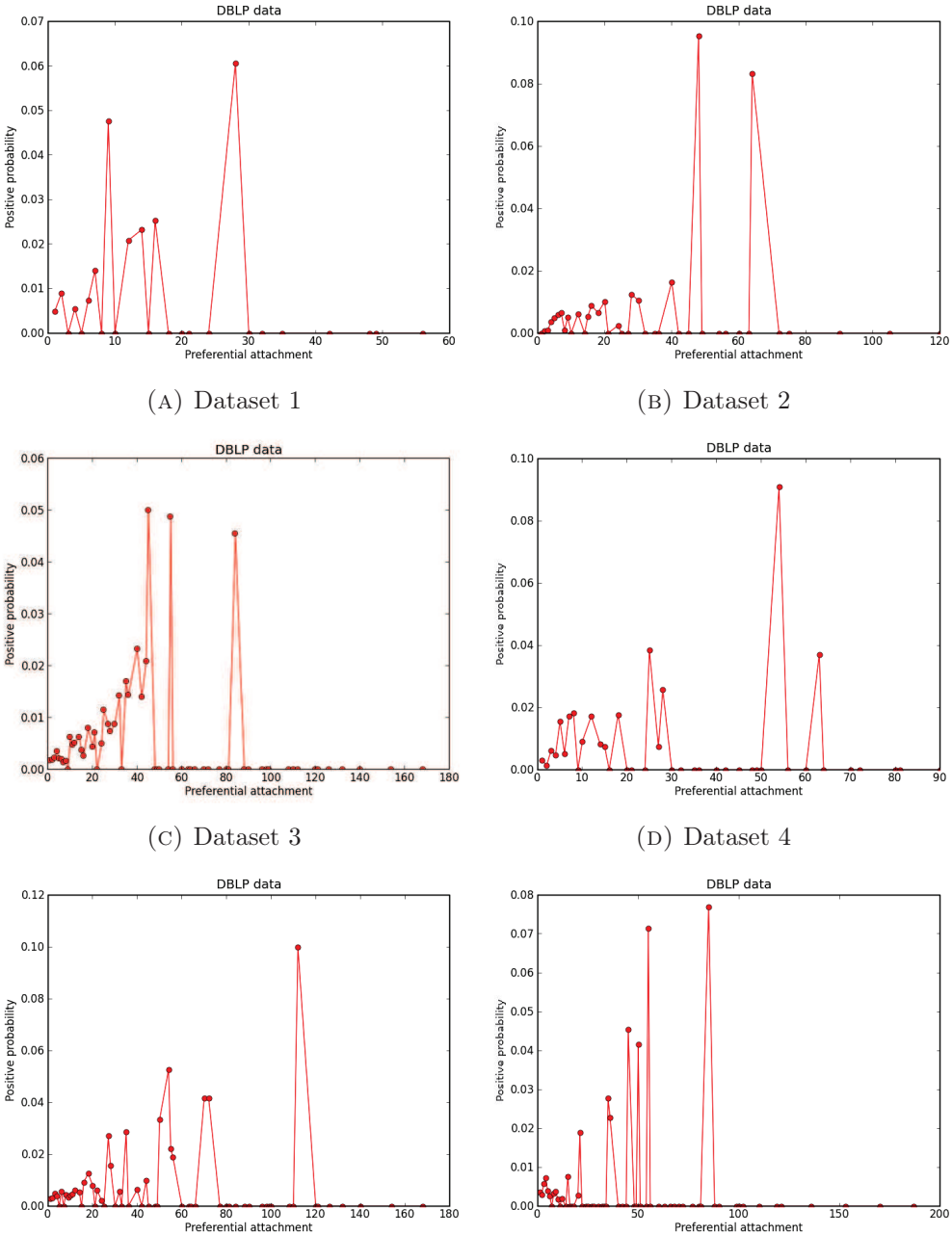


(E) Dataset 5



(F) Dataset 6

FIGURE C.5: Positive probability of neighbor's clustering coefficient



(A) Dataset 1

(B) Dataset 2

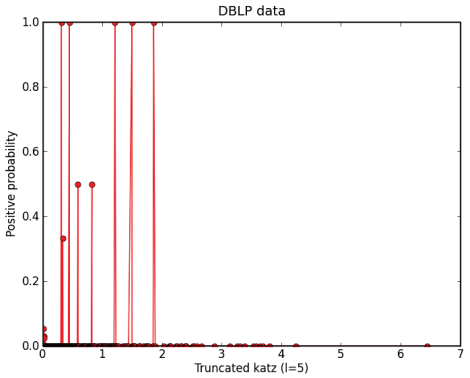
(C) Dataset 3

(D) Dataset 4

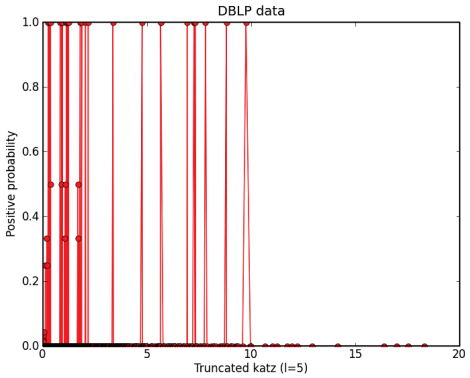
(E) Dataset 5

(F) Dataset 6

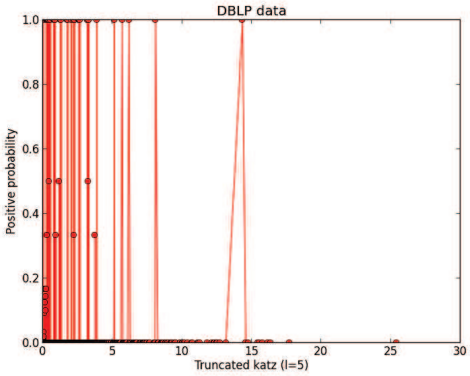
FIGURE C.6: Positive probability of preferential attachment



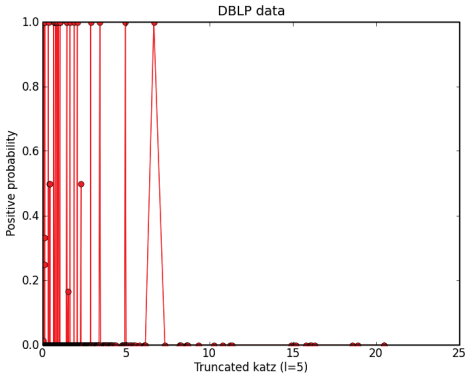
(A) Dataset 1



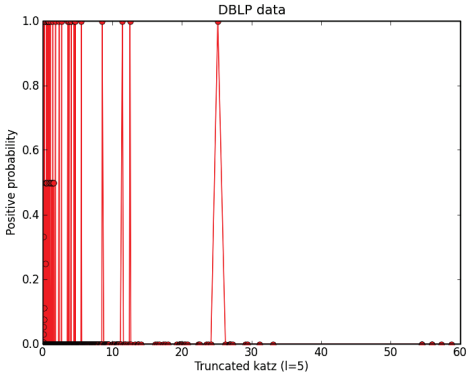
(B) Dataset 2



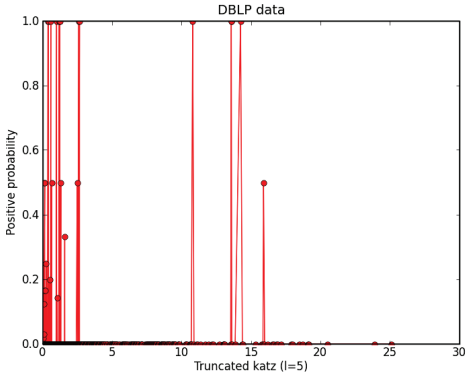
(C) Dataset 3



(D) Dataset 4

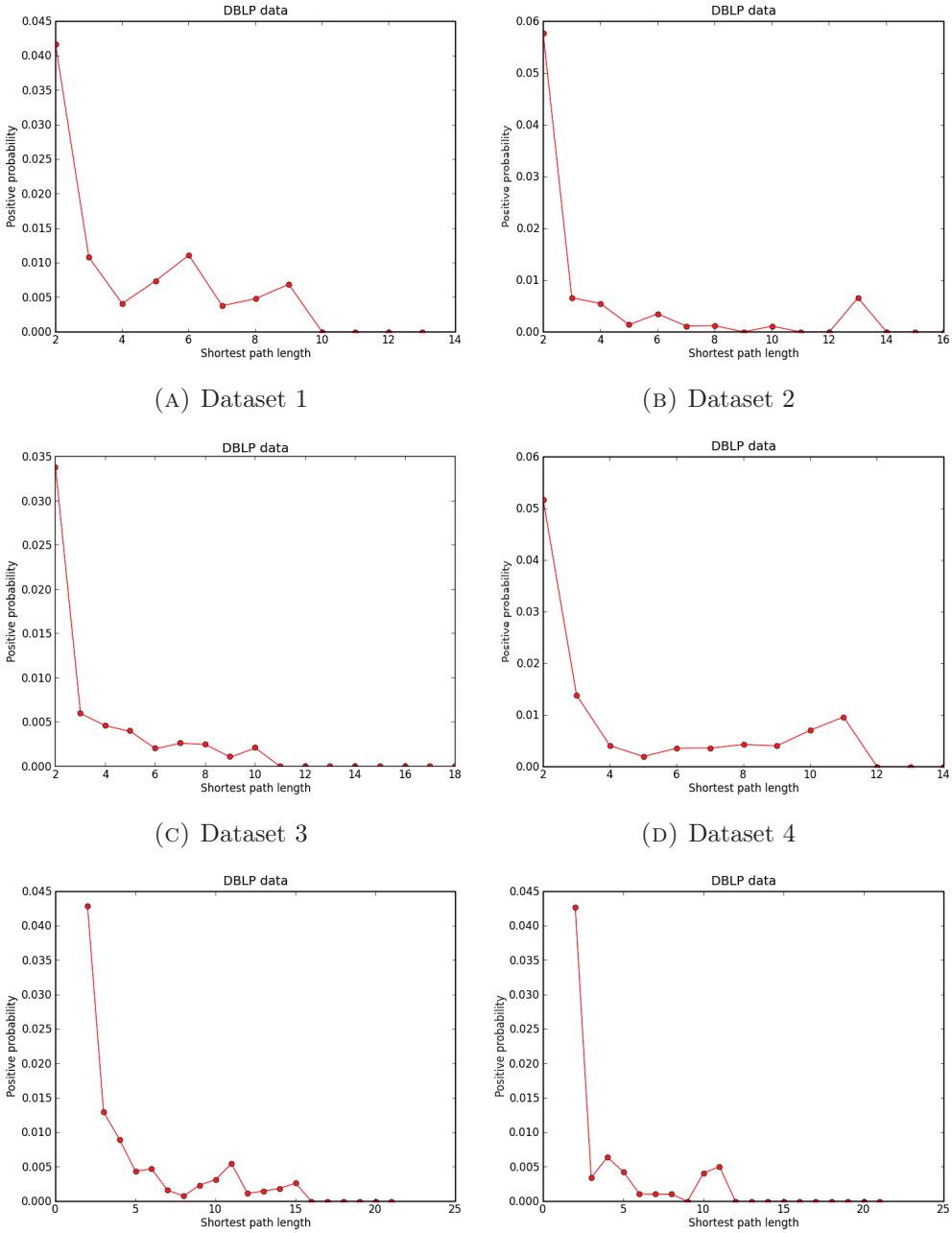


(E) Dataset 5



(F) Dataset 6

FIGURE C.7: Positive probability of truncated Katz centrality



(A) Dataset 1

(B) Dataset 2

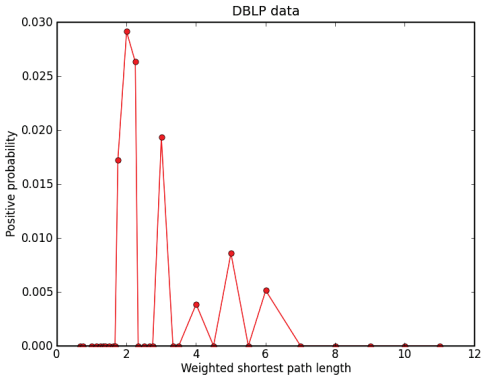
(C) Dataset 3

(D) Dataset 4

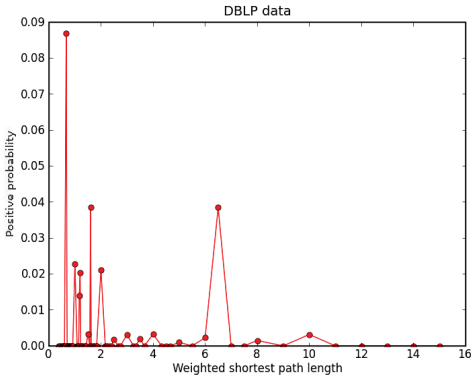
(E) Dataset 5

(F) Dataset 6

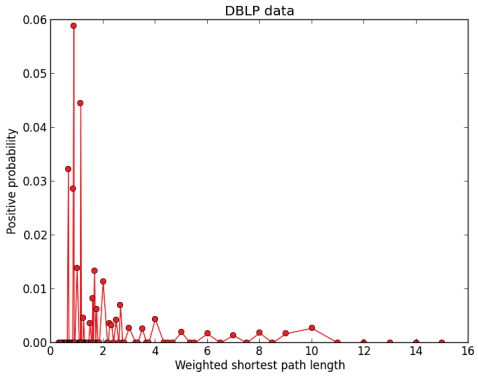
FIGURE C.8: Positive probability of shortest path length



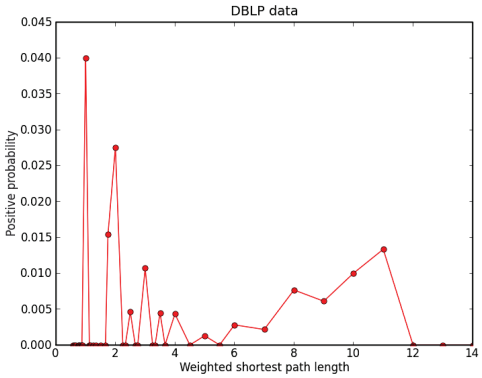
(A) Dataset 1



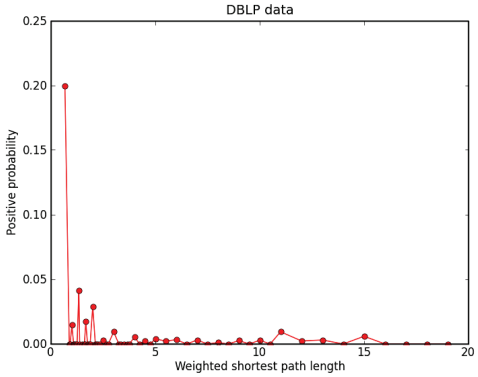
(B) Dataset 2



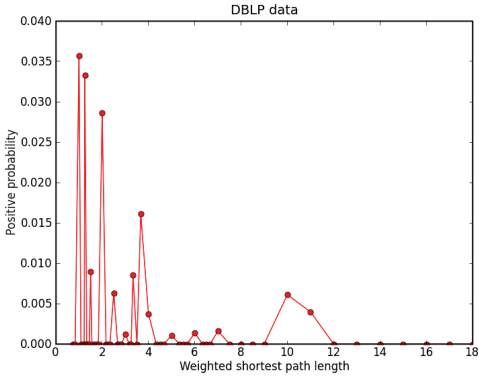
(C) Dataset 3



(D) Dataset 4



(E) Dataset 5



(F) Dataset 6

FIGURE C.9: Positive probability of weighted shortest path length

Appendix D

Publications

Journal

- Manisha Pujari, Rushed Kanawati. *Link prediction in multiplex networks.* Special Issue of Networks and Heterogeneous Media entitled New trends, models and applications in Complex and Multiplex Networks, Volume 10, Number 1, pages. 17 - 35, 2014.
- Manisha Pujari, Rushed Kanawati. *Link prediction in multiplex bibliographical networks.* International Journal of Complex Systems in Science (Proceedings of NET-WORKS 2013, El Escorial),Volume 3, Issue 1, pages. 77-82, December, 2013.

Book chapters

- Manisha Pujari, Rushed Kanawati. *Link prediction in large-scale multiplex networks.* Chapter in Interactions in Complex Systems, Stéphane Cordier, Nicolas Debarsy, Christel Vrain. (Eds.) Cambridge Scholar Publishing, 2014.

International conferences

- Manisha Pujari, Rushed Kanawati. *Link prediction in complex networks by supervised rank Aggregation.* ICTAI 2012: 24th IEEE International Conference on Tools with Artificial Intelligence, pages. 782-789, 7-9 November, 2012, Athens, Greece. (selection rate 41%).
- Manisha Pujari, Rushed Kanawati. *Tag recommendation by link prediction based on supervised machine learning.* Sixth International AAAI Conference on Weblogs and Social Media (ICWSM 2012), 4-7 June 2012, Dublin. (Poster session) (selection rate 26%)
- Manisha Pujari, Rushed Kanawati. *Supervised rank aggregation approach for link prediction in complex networks.* In Proceedings of the 21st international conference companion on World Wide Web (WWW '12 Companion). ACM, New York, NY, USA, pages.1189-1196. (International workshop on Mining Social Network Dynamics (MSND 2012) @WWW'2012, Lyon , 16 April 2012.) (selection rate 56%)

- Manisha Pujari, Rushed Kanawati. *Supervised machine learning link prediction approach for tag recommandation.* 4th International Conference on Online Communities and Social Computing @ HCI International, pages. 336-344, 9-14 July 2011, Hilton Orlando Bonnet Creek, Orlando, Florida, USA, LNCS Springer.

National conferences

- Manisha Pujari, Rushed Kanawati. *Supervised rank aggregation approach for link prediction in complex networks.* 3ième conférences sur les modèles de analyse des réseaux : approches mathématiques et informatiques (MARAMI), 17-19 October 2012 Villetaneuse.

Workshops

- Manisha Pujari. *Link prediction in multiplex networks: application to co-authorship link prediction in bibliographical networks* AFGG@EGC2014, 28 January, 2014, Rennes, France.
- Manisha Pujari. *Path betweenness centrality: A new topological measure for link prediction.* Journée de fouille de grandes graphes (JFGG), MARAMI2013, 16-18 October 2013, Saint-Etienne, France.
- Manisha Pujari, Rushed Kanawati. *Applying supervised rank aggregation to link prediction in large-scale complex networks.* Big Data Mining and Visualization 18-19 June 2012, Tours, France.
- N. Benchettara, R. Kanawati, M. Pujari, C. Rouveirol, G. Santini. *Approches topologiques pour la prévision de liens dans des grands graphes de terrain : application au calcul de recommandation.* Journée moteurs de recommandation au CNAM, 11 June 2012, Paris.
- Manisha Pujari, Rushed Kanawati. *Link prediction approach for tag recommendation in folksonomy.* Actes de la 1er journée de fouille de grands graphes (FGG'10), 13 October 2010, Toulouse.

Invited talks

- Manisha Pujari. *Topological approaches for link prediction in large-scale complex networks*, Franco-Taiwanese Days, INRIA, 8-14 September 2012, Paris, France.

Appendix E

DBLP Network Visualization

E.1 Co-authorship networks

We present below the visualizations of co-authorship graphs created from the DBLP data corresponding to different time period. All visualizations have been done using Gephi [1] [Bastian et al., 2009]. The nodes represent authors of scientific papers and the links are added if two authors have written or published an article together. There is also visualization of the largest connected components (LCC) of a few graphs.

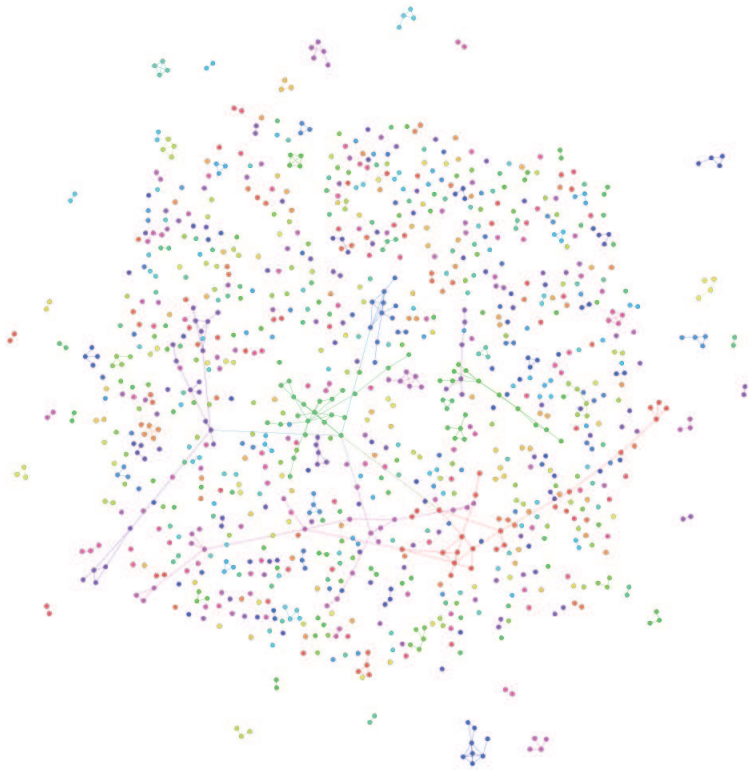


FIGURE E.1: Co-authorship network for year 1970-1973

[1] http://gephi.github.io/

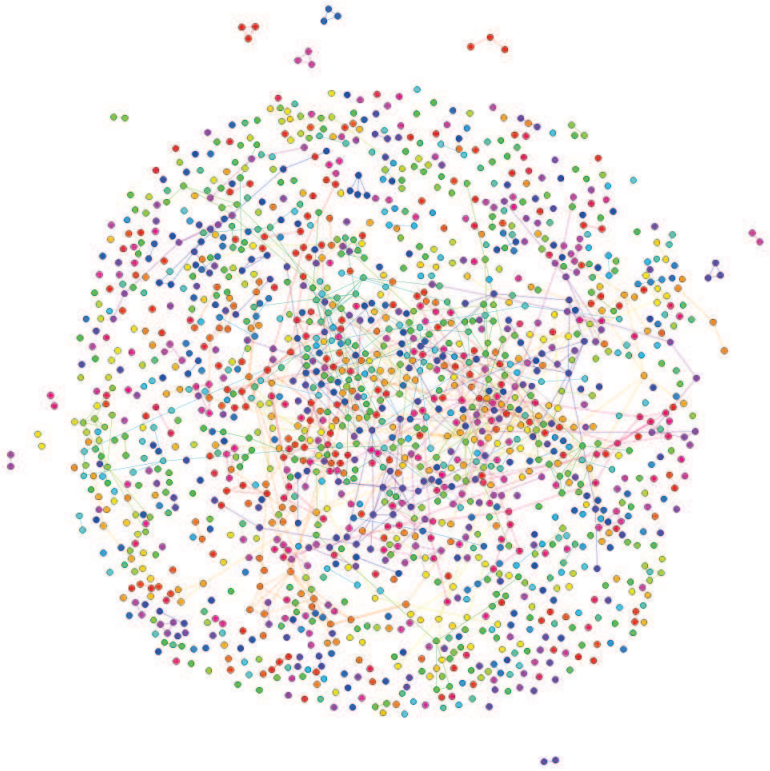


FIGURE E.2: Co-authorship network for year 1972-1975

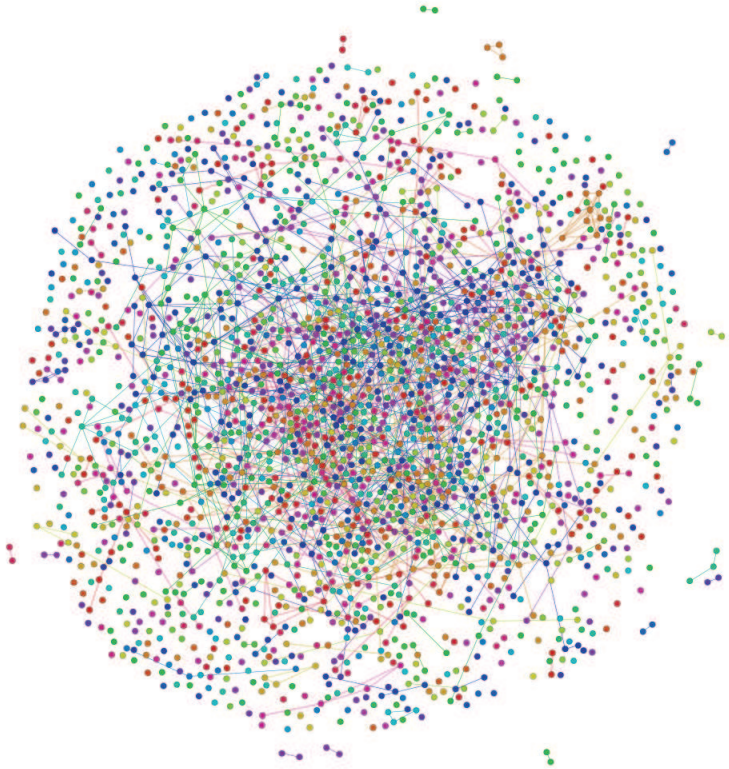


FIGURE E.3: Co-authorship network for year 1974-1977

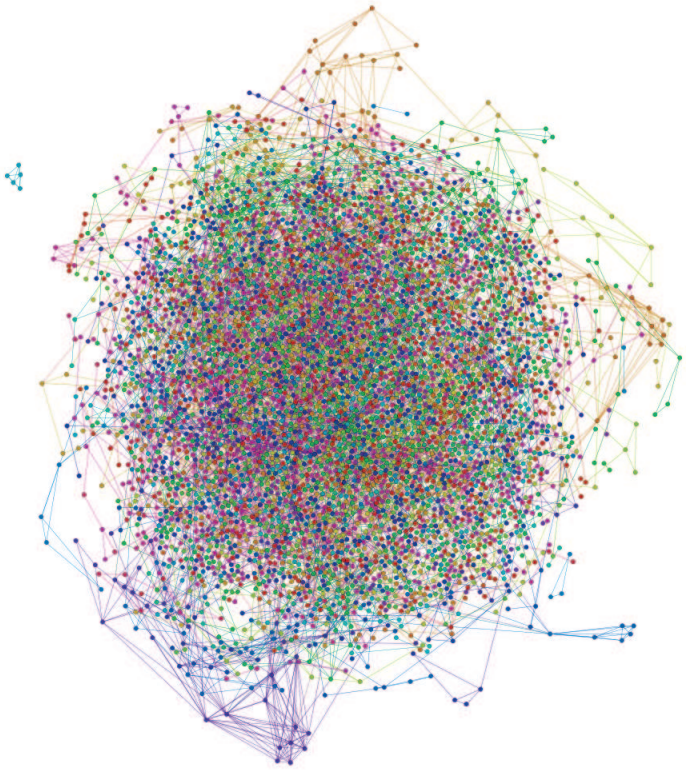


FIGURE E.4: Co-authorship network for year 1980-1983

FIGURE E.5: Co-authorship network for year 1982-1985

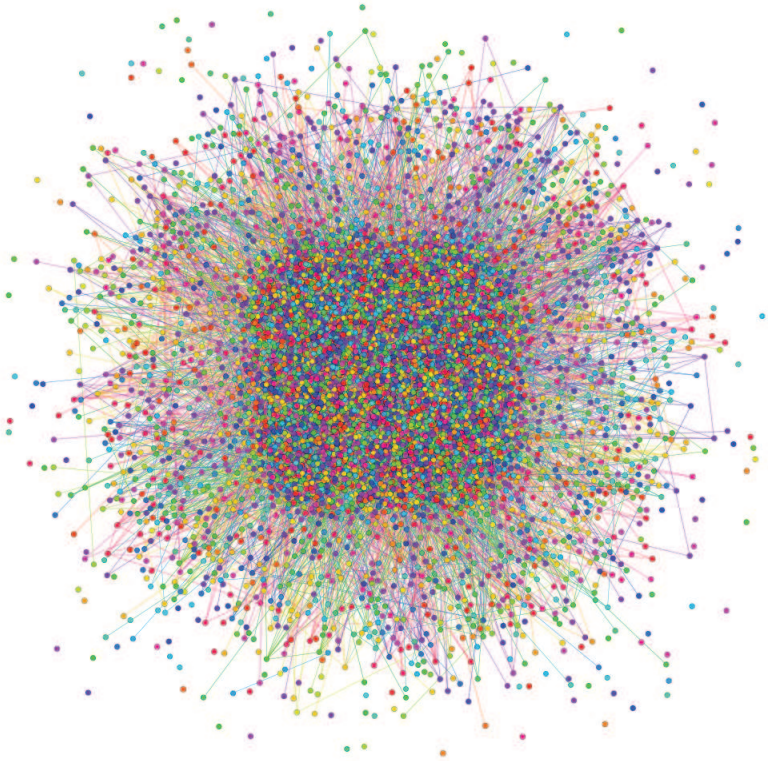


FIGURE E.6: Co-authorship network for year 1984-1987

FIGURE E.7: LCC of co-authorship network for year 1980-1983

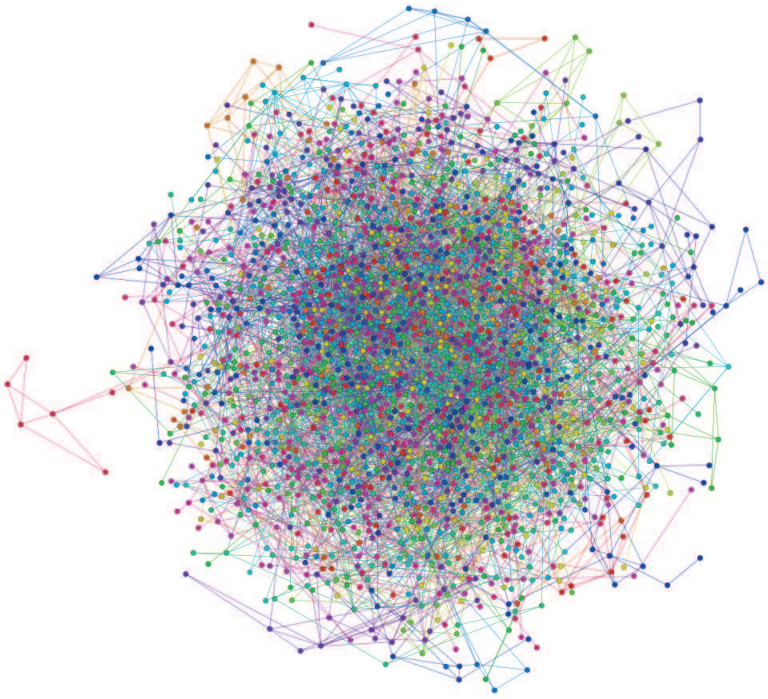


FIGURE E.8: LCC of co-authorship network for year 1982-1985

FIGURE E.9: LCC of co-authorship network for year 1984-1987

E.2 Multiplex networks

This section contains the visualization of multiplex networks created from DBLP data corresponding to different periods of time. These networks represent different kinds of links between authors (nodes) in the form of different layers of graphs. All visualizations have been done using MuxViz[2]. MuxViz is a framework for multilayer network analysis and visualization.

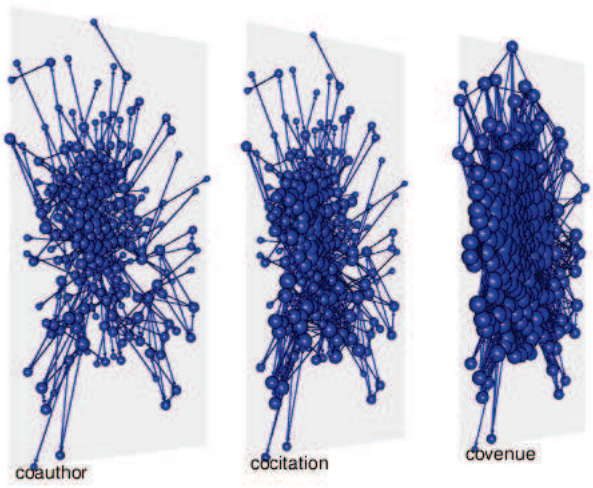


FIGURE E.10: LCC of network for year 1972-1975

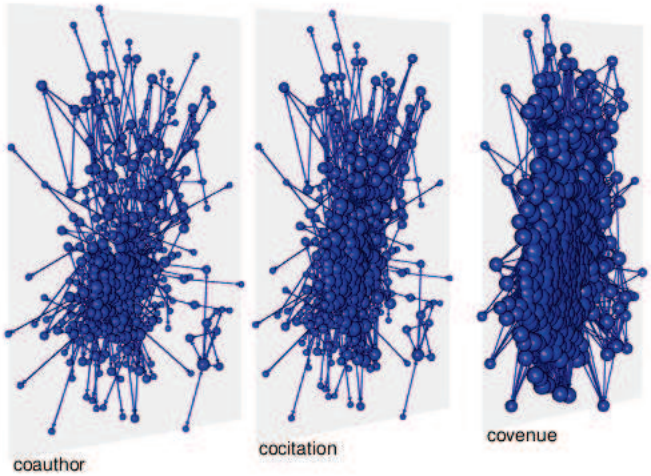


FIGURE E.11: LCC of network for year 1974-1977

[2]http://muxviz.net/

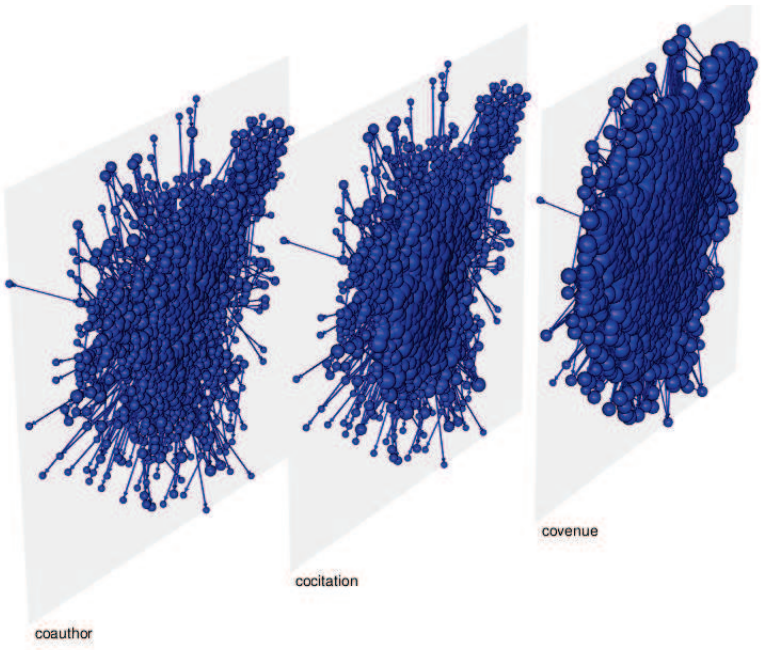


FIGURE E.12: LCC of network for year 1980-1983

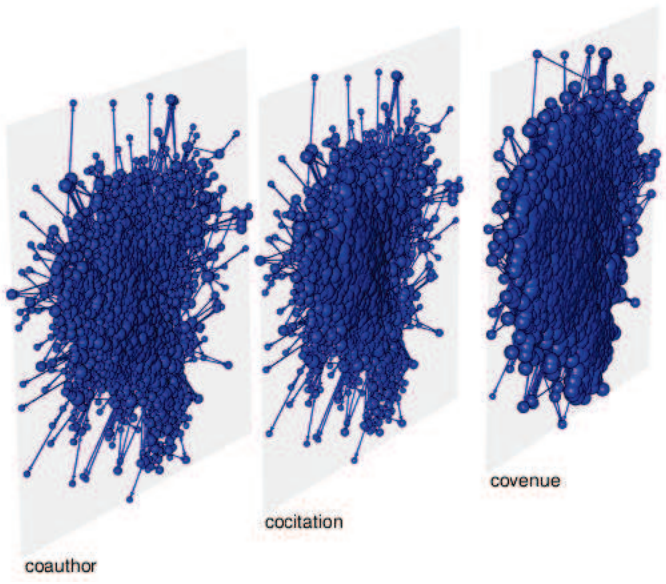


FIGURE E.13: LCC of network for year 1982-1985

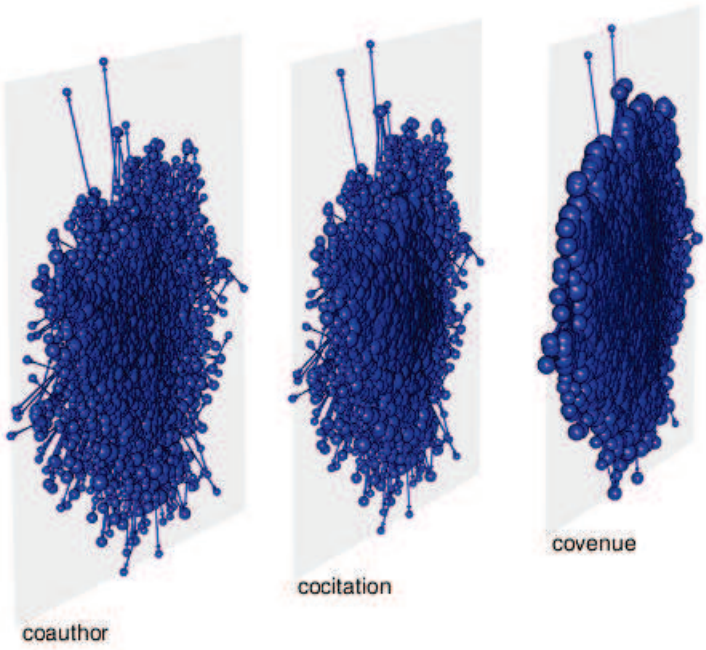


FIGURE E.14: LCC of network for year 1984-1987

Bibliography

Evrim Acar, Daniel M. Dunlavy, and Tamara G. Kolda. Link prediction on evolving data using matrix and tensor factorizations. In *ICDM Workshops*, pages 262–269, 2009. URL http://dblp.uni-trier.de/db/conf/icdm/icdmw2009.html#AcarDK09.

Balázs Adamcsek, Gergely Palla, Illés J. Farkas, Imre Derényi, and Tamás Vicsek. Cfinder: locating cliques and overlapping modules in biological networks. *Bioinformatics*, 22(8):1021–1023, 2006.

Lada Adamic and Eytan Adar. Friends and neighbors on the Web. *Social Networks*, 25 (3):211–230, 2003.

Lada Adamic, Orkut Buyukkokten, and Eytan Adar. A social network caught in the Web. *First Monday*, 8(6), 2003.

Charu C. Aggarwal. *Social Network Data Analytics*, chapter An introduction to social network data analytics. Springer, 2011.

Charu C. Aggarwal, Yan Xie, and S. Yu Philip. On dynamic link inference in heterogeneous networks. In *SDM*, pages 415–426. SIAM, 2012.

Yong-Yeol Ahn, Seungyeop Han, Haewoon Kwak, Sue Moon, and Hawoong Jeong. Analysis of topological characteristics of huge online social networking services. In *Proceedings of the 16th international conference on World Wide Web*, pages 835–844, 2007.

Edoardo M. Airoldi, David M. Blei, Stephen E. Fienberg, Eric P. Xing, and Tommi Jaakkola. Mixed membership stochastic block models for relational data with application to protein-protein interactions. In *Proceedings of the International Biometrics Society Annual Meeting*, 2006.

Mohammad Al Hasan and Mohammed J. Zaki. A survey of link prediction in social networks. In Charu C. Aggarwal, editor, *Social network Data Analysis*, chapter 9, pages 243–275. Springer, 2010.

Javed A. Aslam and Mark Montague. Models for metasearch. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '01, pages 276–284, New York, NY, USA, 2001. ACM. ISBN 1-58113-331-6. doi: http://doi.acm.org/10.1145/383952.384007. URL http://doi.acm.org/10.1145/383952.384007.

Zhifeng Bao, Yong Zeng, and Y. C. Tay. Sonlp: Social network link prediction by principal component regression. In *Advances in Social Networks Analysis and Mining (ASONAM), 2013 IEEE/ACM International Conference on*, pages 364–371. IEEE, 2013.

Albert-László Barabási. Scale-free networks: A decade and beyond. *Science*, 325(5939): 412–413, 2009. ISSN 1095-9203. doi: 10.1126/science.1173299.

Albert-László Barabasi and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999. ISSN 1095-9203. doi: 10.1126/science.286.5439.509.

Albert-László Barabási and Z. N. Oltvai. Network biology: understanding the cell's functional organization. *Nature Reviews Genetics*, 5:101–113, 2004.

Albert-László Barabási, H. Jeong, Z. Néda, E. Ravasz, A. Schubert, and T. Vicsek. Evolution of the social network of scientific collaboration. *Physica A*, 311(3-4):590–614, 2002. URL arXiv:cond-mat/0104162v1.

A. Barrat, M. Barthélemy, and A. Vespigani. Modeling the evolution of weighted networks. *Physical Review E 70:066149*, 2004.

Marc Barthélemy. Spatial networks. *Physics Reports*, 499(1-3):1–101, 2011. ISSN 03701573. doi: 10.1016/j.physrep.2010.11.002.

Mathieu Bastian, Sebastien Heymann, Mathieu Jacomy, et al. Gephi: an open source software for exploring and manipulating networks. *ICWSM*, 8:361–362, 2009.

Federico Battiston, Vincenzo Nicosia, and Vito Latora. Metrics for the analysis of multiplex networks. *CoRR*, abs/1308.3182, August 2013. URL http://dblp.uni-trier.de/db/journals/corr/corr1308.html#BattistonNL13.

Nasserine Benchettara. *Prévision de nouveaux liens dans les réseaux d'interactions bipartis : Application au calcul de recommandationPrévision de nouveaux liens dans les réseaux d'interactions bipartis : Application au calcul de recommandation.* PhD thesis, LIPN, Université Paris Nord, 2011.

Nesserine Benchettara, Rushed Kanawati, and Céline Rouveirol. Supervised machine learning applied to link prediction in bipartite social networks. In *International Conference on Advances in Social Network Analysis and Mining, ASONAM 2010*, pages 326–330, 2010a.

Nesserine Benchettara, Rushed Kanawati, and Céline Rouveirol. A supervised machine learning link prediction approach for academic collaboration recommendation. In *Proceedings of the fourth ACM conference on Recommender systems - RecSys '10*, pages 253–256, New York, New York, USA, 2010b. ACM Press. ISBN 9781605589060. doi: 10.1145/1864708.1864760. URL http://dblp.uni-trier.de/db/conf/recsys/recsys2010.html#BenchettaraKR10.

M. Berlingerio, M. Coscia, F. Giannotti, A. Monreale, and D. Pedreschi. Foundations of Multidimensional Network Analysis. In *Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on*, pages 485–489. IEEE, July 2011a. ISBN 978-1-61284-758-0. doi: 10.1109/asonam.2011.103. URL http://dx.doi.org/10.1109/asonam.2011.103.

Michele Berlingerio, Francesco Bonchi, Björn Bringmann, and Aristides Gionis. Mining graph evolution rules. In Wray L Buntine, Marko Grobelnik, Dunja Mladenic, and John Shawe-Taylor, editors, *ECML/PKDD (1)*, volume 5781 of *Lecture Notes in Computer Science*, pages 115–130. Springer, 2009. ISBN 978-3-642-04179-2.

Michele Berlingerio, Michele Coscia, and Fosca Giannotti. Finding and characterizing communities in multidimensional networks. In *Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on*, pages 490–494. IEEE, 2011b.

Jérémy Besson and Céline Robardet. A new way to aggregate preferences : Application to eurovision song contest. In *Proc. 7th Int. Symp. on Intelligent Data Analysis IDA'07*, LNCS, pages 152–162. Springer, September 2007. doi: 10.1007/978-3-540-74825-0.

Duncan Black, R.A. Newing, Iain McLean, Alistair McMillan, and Burt Monroe. *The Theory of Committees and Elections by Duncan Black, and Revised Second Editions Committee Decisions with Complementary Valuation by Duncan Black*. Kluwer Academic Publishing, 2nd edition, 1998. ISBN 0792381106.

Vincent D. Blondel, Jean-loup Guillaume, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008.

Béla Bollobás. *Random graphs*. Cambridge University Press, 2 edition, 2001. ISBN 0521797225.

Ulrik Brandes. On variants of shortest-path betweenness centrality and their generic computation. *Social Networks*, 30(2):136 – 145, 2008. ISSN 0378-8733. doi: http://dx.doi.org/10.1016/j.socnet.2007.11.001. URL http://www.sciencedirect.com/science/article/pii/S0378873307000731.

Ulrik Brandes and Dorothea Wagner. Analysis and visualization of social networks. In *Graph drawing software*, pages 321–340. Springer, 2004.

Céline Brouard, Florence D'Alché-Buc, and Marie Szafranski. Semi-supervised penalized output kernel regression for link prediction. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 593–600, 2011.

Aydin Buluç, Henning Meyerhenke, Ilya Safro, Peter Sanders, and Christian Schulz. Recent advances in graph partitioning. *CoRR*, abs/1311.3144, 2013.

Guido Caldarelli, Andrea Capocci, Paolo De Los Rios, and Miguel A Muñoz. Scale-free networks from varying vertex intrinsic fitness. *Physical review letters*, 89(25):258702, 2002.

Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16(1):321–357, January 2002.

P. Chebotarev and E. Shamis. The matrix-forest theorem and measuring relations in small social groups. *Automation and Remote Control*, 58(9):1505–1514, 1997.

Yann Chevaleyre, Ulle Endriss, J. Lang, and N. Maudet. A short introduction to computational social choice. *SOFSEM 2007: Theory and Practice of Computer Science*, pages 51–69, 2007. URL http://www.springerlink.com/index/768446470RPLJ120.pdf.

Fan Chung and Linyuan Lu. The diameter of random sparse graphs. *Advances in Applied Math*, 26(4):257–279, 2001.

Aaron Clauset, Cristopher Moore, and M. E. J. Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453(7191):98–101, 2008. ISSN 0028-0836. doi: 10.1038/nature06830.

M. Condorcet. Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix. *Paris, France*, 1785.

Richard J E Cooke. Link prediction and link detection in sequences of large social networks using temporal and local metrics. Master thesis, University of cape Town, 2006.

Gennaro Cordasco and Luisa Gargano. Label propagation algorithm: a semi-synchronous approach. *IJSNM*, 1(1):3–26, 2012.

Carlos D. Correa and Kwan-Liu Ma. Visualizing social networks. In *Social Network Data Analytics*, pages 307–326. Springer, 2011.

Emanuele Cozzo, Mikko Kivelä, Manlio De Domenico, Albert Solé, Alex Arenas, Sergio Gómez, Mason A Porter, and Yamir Moreno. Clustering coefficients in multiplex networks. *CoRR*, 2013.

Darcy Davis, Ryan Lichtenwalter, and Nitesh V. Chawla. Multi-relational link prediction in heterogeneous information networks. In *2011 International Conference on Advances in Social Networks Analysis and Mining*, pages 281–288. IEEE, July 2011. ISBN 978-1-61284-758-0. doi: 10.1109/ASONAM.2011.107. URL http://dblp.uni-trier.de/db/conf/asunam/asonam2011.html#DavisLC11.

Darcy Davis, Ryan Lichtenwalter, and Nitesh V Chawla. Supervised methods for multi-relational link prediction. *Social Network Analysis and Mining*, pages 1–15, 2013.

Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240. ACM, 2006.

Jean de Borda. Mémoire sur les elections au scrutin. 1781.

Manlio De Domenico, Albert Solé, Sergio Gómez, and Alex Arenas. Random walks on multiplex networks. *CoRR*, 2013a.

Manlio De Domenico, Albert Solé-Ribalta, Elisa Omodei, Sergio Gómez, and Alex Arenas. Centrality in interconnected multilayer networks. *CoRR*, abs/1311.2906, 2013b.

Janez Demšar, Tomaž Curk, Aleš Erjavec, Črt Gorup, Tomaž Hočevar, Mitar Milutinovič, Martin Možina, Matija Polajnar, Marko Toplak, Anže Starič, et al. Orange: data mining toolbox in python. *The Journal of Machine Learning Research*, 14(1): 2349–2353, 2013.

Yon Dourisboure, Filippo Geraci, and Marco Pellegrini. Extraction and classification of dense communities in the web. In *Proceedings of the 16th international conference on World Wide Web*, pages 461–470. ACM, 2007.

Nan Du, Bai Wang, Bin Wu, and Yi Wang. Overlapping community detection in bipartite networks. In *Web Intelligence*, pages 176–179. IEEE, 2008.

Daniel M. Dunlavy, Tamara G. Kolda, and Evrim Acar. Temporal link prediction using matrix and tensor factorizations. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 5(2):10, 2011.

C. Dwork, R. Kumar, Naor M., and D. Sivakumar. Rank aggregation methods for web. In *Proceedings of the 10th international conference on World Wide Web*, WWW '01, pages 613–622, Hong Kong, 2001. ACM. doi: http://doi.acm.org/10.1145/371920.372165. URL http://doi.acm.org/10.1145/371920.372165.

Paul Erdős and Alfréd Rényi. On random graphs. *Publicationes Mathematicae Debrecen*, 6:290–297, 1959.

Lauri Eronen and Hannu Toivonen. Biomine: predicting links between biological entities using network models of heterogeneous databases. *BMC bioinformatics*, 13(1):119, January 2012.

Ernesto Estrada. *The Structure of Complex Networks: Theory and Applications*. Oxford University Press, 2011.

Ronald Fagin, Ravi Kumar, and D. Sivakumar. Efficient similarity search and classification via rank aggregation. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, SIGMOD '03, pages 301–312, New York, NY, USA, 2003. ACM. ISBN 1-58113-634-X. doi: http://doi.acm.org/10.1145/872757.872795. URL http://doi.acm.org/10.1145/872757.872795.

Katherine Faust and Stanley Wasserman. Centrality and prestige: A review and synthesis. *Journal of Quantitative Anthropology*, 4(1):23–78, 1992.

Michael Fire, Lena Tenenboim, Ofrit Lesser, Rami Puzis, Lior Rokach, and Yuval Elovici. Link prediction in social networks using computationally efficient topological features. *Proceedings of the 3rd IEEE Int. Conference on Social Computing - SocialCom-11*, 2011. URL http://www.ise.bgu.ac.il/faculty/liorr/CONF5.pdf.

Michael Fire, Rami Puzis, and Yuval Elovici. Link prediction in highly fractional data sets. In *Handbook of Computational Approaches to Counterterrorism*, pages 283–300. Springer New York, 2013.

Gary William Flake, Steve Lawrence, C. Lee Giles, and Frans Coetzee. Self-organization and identification of web communities. *IEEE Computer*, 35(3):66–71, 2002.

S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75–174, 2010.

S. Fortunato and M. Barthélemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences,*, 104(1):36, 2007.

Francois Fouss, Luh Yen, Alain Pirotte, and Marco Saerens. An experimental investigation of graph kernels on a collaborative recommendation task. In *Sixth International Conference on Data Mining (ICDM'06)*, pages 863–868. IEEE, December 2006. ISBN 0-7695-2701-7. doi: 10.1109/icdm.2006.18. URL http://dx.doi.org/10.1109/icdm.2006.18.

L.C. Freeman. A set of measures of centrality based upon betweenness. In *Sociometry 40*, pages 35–41, 1977.

Linton C. Freeman. Visualizing social networks. *Journal of social structure*, 1(1):4, 2000.

Linton C. Freeman. *The development of social network analysis: A study in the sociology of science*, volume 1. Empirical Press Vancouver, 2004.

Antonino Freno, Gemma C. Garriga, and Mikaela Keller. Learning to recommend links using graph structure and node content. In *Neural Information Processing Systems Workshop on Choice Models and Preference Learning*, Granada, Spain, December 2011.

Yupeng Fu, Rongjing Xiang, Yiqun Liu, Min Zhang, and Shaoping Ma. Finding Experts Using Social Network Analysis. In *Web Intelligence*, pages 77–80. IEEE Computer Society, 2007. ISBN 0-7695-3026-5.

Sheng Gao, Ludovic Denoyer, and Patrick Gallinari. Temporal link prediction by integrating content and structure information. In *Proceedings of the 20th ACM international conference on Information and knowledge management - CIKM '11*, page 1169, New York, New York, USA, 2011. ACM Press. ISBN 9781450307178. doi: 10.1145/2063576.2063744. URL http://dblp.uni-trier.de/db/conf/cikm/cikm2011.html#GaoDG11.

Lisa Getoor, Nir Friedman, Daphne Koller, and Benjamin Taskar. Learning probabilistic models of link structure. *The Journal of Machine Learning Research*, 3:679–707, 2003.

B. H. Good, Y.-A. de Montjoye, and A. Clauset. The performance of modularity maximization in practical contexts. *Physical Review*, E(81):046106, 2010.

Roger Guimera and Luis A. Nunes Amaral. Functional cartography of complex metabolic networks. *Nature*, 433(7028):895–900, 02 2005. URL http://dx.doi.org/10.1038/nature03288.

Roger Guimerà, Stefano Mossa, Adrian Turtschi, and Luis A. Nunes Amaral. The worldwide air transportation network: Anomalous centrality, community structure, and cities' global roles. *Proceedings of the National Academy of Sciences*, 102(22):7794–7799, 2005.

Roger Guimerà, Marta Sales-Pardo, and Luis A Nunes Amaral. A network-based method for target selection in metabolic networks. *Bioinformatics*, 23(13):1616–1622, 2007.

A. Halu, R. J. Mondragon, P. Pansaraza, and G. Bianconi. Multiplex pagerank. *arXiv preprint arXiv:1306.3576*, 2013.

Franck Harary. *Graph Theory*. Addison–Wesley, 1969.

Mohammad Al Hasan, Vineet Chaoji, Saeed Salem, and Mohammed Zaki. Link prediction using supervised learning. In *Workshop on link analysis, Counter-terrorism and security, SIAM Data Mining Conference*, Bethesda, MD, 2006.

Jian Huang, Ziming Zhuang, Jia Li, and C. Lee Giles. Collaboration over time: characterizing and modeling network evolution. In Marc Najork, Andrei Z Broder, and Soumen Chakrabarti, editors, *WSDM*, pages 107–116. ACM, 2008.

Zan Huang. Link prediction based on graph topology: The predictive value of the generalized clustering coeffecient. In *Proceedings of LinkKDD'06*, Philadelphia, Pennsylvania, 2006.

Zan Huang and Dennis K. J. Lin. The time-series link prediction problem with applications in communication surveillance. *INFORMS JOURNAL ON COMPUTING*, published, 2008.

Zan Huang, Xin Li, and Hsinchun Chen. Link prediction approach to collaborative filtering. In Mary Marlino, Tamara Sumner, and Frank M Shipman III, editors, *JCDL*, pages 141–142. ACM, 2005. ISBN 1-58113-876-8.

P. Jaccard. Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bulletin de la Société Vaudoise des Sciences Naturelles*, 37:547–579, 1901.

Robert Jäschke, Leandro Balby Marinho, Andreas Hotho, Lars Schmidt-Thieme, and Gerd Stumme. Tag recommendations in social bookmarking systems. *AI Commun.*, 21(4):231–247, 2008.

George Karypis and Vipin Kumar. Analysis of multilevel graph partitioning. In *Proceedings of the 1995 ACM/IEEE conference on Supercomputing*, page 29. ACM, 1995.

Hisashi Kashima, Tsuyoshi Kato, Yoshihiro Yamanishi, Masashi Sugiyama, and Koji Tsuda. Link propagation: A fast semi-supervised learning algorithm for link prediction. In *SDM*, volume 9, pages 1099–1110. SIAM, 2009.

L. Katz. A new status index derived from sociometric analysis. *Psychmetrika, 18(1)*, 18 (1):39–43, 1953.

Reihaneh Rabbany Khorasgani, Jiyang Chen, and Osmar R. Zaiane. Top leaders community detection approach in information networks. In *4th SNA-KDD Workshop on Social Network Mining and Analysis*, Washington D.C., 2010.

Mikko Kivelä, Alexandre Arenas, Marc Barthelemy, James P. Gleeson, Yamir Moreno, and Mason A. Porter. Multilayer networks. *Journal of Complex Networks*, 2(3):203–271, July 2013.

G. Kossinets. Effects of missing data in social networks. *Social Networks*, 28(3):247–268, 2006. ISSN 03788733. doi: 10.1016/j.socnet.2005.07.002.

Miroslav Kubat, Stan Matwin, et al. Addressing the curse of imbalanced training sets: one-sided selection. In *ICML*, volume 97, pages 179–186. Nashville, USA, 1997.

Ravi Kumar, Kunal Punera, Torsten Suel, and Sergei Vassilvitskii. Top-k aggregation using intersections of ranked inputs. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, WSDM '09, pages 222–231, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-390-7. doi: 10.1145/1498759.1498830. URL http://doi.acm.org/10.1145/1498759.1498830.

Jérôme Kunegis, Ernesto W De Luca, and Sahin Albayrak. The Link Prediction Problem in Bipartite Networks. *Computational Intelligence for KnowledgeBased Systems Design*, page 10, 2010. URL http://arxiv.org/abs/1006.5367.

Mayank Lahiri and Tanya Y Berger-Wolf. Structure Prediction in Temporal Networks using Frequent Subgraphs. In *CIDM*, pages 35–42. IEEE, 2007.

Renaud Lambiotte. Multi-scale modularity in complex networks. In *Modeling and optimization in mobile, ad hoc and wireless networks (WiOpt), 2010 Proceedings of the 8th International Symposium on*, pages 546–553. IEEE, 2010.

Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Trans. Knowl. Discov. Data*, 1(1):2, 2007. ISSN 1556-4681. doi: 10.1145/1217299.1217301.

Jure Leskovec, Kevin J. Lang, and Michael W. Mahoney. Empirical comparison of algorithms for network community detection. In Michael Rappa, Paul Jones, Juliana Freire, and Soumen Chakrabarti, editors, *WWW*, pages 631–640. ACM, 2010. ISBN 978-1-60558-799-8.

Lin Li, Bao-Yan Gu, and Li Chen. The topological characteristics and community structure in consumer-service bipartite graph. In Jie Zhou, editor, *Complex (1)*, volume 4 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 640–650. Springer, 2009. ISBN 978-3-642-02465-8.

David Liben-Nowell. *An algorithmic approach to social networks*. PhD thesis, M.I.T., 2005.

David Liben-Nowell and Jon M Kleinberg. The link-prediction problem for social networks. *JASIST*, 58(7):1019–1031, 2007.

R. Lichtenwalter and N.V. Chawla. Link prediction: Fair and effective evaluation. In *Advances in Social Networks Analysis and Mining (ASONAM), 2012 IEEE/ACM International Conference on*, pages 376–383, 2012. doi: 10.1109/ASONAM.2012.68.

Ryan N. Lichtenwalter, Notre Dame, Jake T Lussier, and Nitesh V Chawla. New perspectives and methods in link prediction. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 243–252. ACM, 2010. ISBN 9781450300551. doi: 10.1145/1835804.1835837. URL http://portal.acm.org/citation.cfm?id=1835804.1835837.

Marel Lipczak. Tag recommendation for folksonomies oriented towards individual users. In *ECML PKDD Discovery Challenge 2009, CEUR Workshop Proceedings Vol. 497*, pages 189–199, 2009.

Nick Littlestone and Manfred K. Warmuth. Weighted majority algorithm. *IEEE Symposium on Foundations of Computer Science*, 1989.

Xin Liu and Tsuyoshi Murata. Community detection in large-scale bipartite networks. In *Web Intelligence*, pages 50–57. IEEE, 2009.

Yu-Ting Liu, Tie-Yan Liu, Tao Qin, Zhi-Ming Ma, and Hang Li. Supervised rank aggregation. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 481–490, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-654-7. doi: http://doi.acm.org/10.1145/1242572.1242638. URL http://doi.acm.org/10.1145/1242572.1242638.

Zhen Liu, Qian-Ming Zhang, Linyuan Lü, and Tao Zhou. Link prediction in complex networks: A local naïve bayes model. *EPL (Europhysics Letters)*, 96(4):48007, 2011.

Zhen Liu, Jia-Lin He, and Jaideep Srivastava. Cliques in complex networks reveal link formation and community evolution. *CoRR*, 2013.

Zhiyuan Liu, Chuan Chi, and Maosong Sun. Folkdiffusion: A graph-based tag suggestion method for folksonomies. In *Information Retrieval Technology*, pages 231–240. Springer Berlin / Heidelberg, 2010.

Linyuan Lü and Tao Zhou. Link prediction in complex networks: A survey. *Physica A: Statistical Mechanics and its Applications*, 390(6):1150–1170, March 2011. ISSN 03784371. doi: 10.1016/j.physa.2010.11.027. URL http://dx.doi.org/10.1016/j.physa.2010.11.027.

Yu-Ta Lu, Shoou-I Yu, Tsung-Chieh Chang, and Jane Yung jen Hsu. A content-based method to enhance tag recommendation. In Craig Boutilier, editor, *IJCAI*, pages 2064–2069, 2009.

Matteo Magnani and Luca Rossi. Pareto distance for multi-layer network analysis. In *Social Computing, Behavioral-Cultural Modeling and Prediction*, pages 249–256. Springer, 2013.

Matteo Magnani, Barbora Micenkova, and Luca Rossi. Combinatorial analysis of multiple networks. *CoRR*, 2013.

Mary McGlohon, Leman Akoglu, and Christos Faloutsos. Statistical properties of social networks. In *Social Network Data Analytics*, pages 17–42. Springer, 2011.

Aditya Krishna Menon and Charles Eklan. Link prediction via matrix factorization. In Dimitrios Gunopulos, Thomas Hofmann, Donato Malerba, and Michalis Vazirgiannis, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 6912 of *Lecture Notes in Computer Science*, pages 437–452. Springer Berlin Heidelberg, 2011.

Mark Montague and Javed A. Aslam. Condorcet fusion for improved retrieval. In *Proceedings of the eleventh international conference on Information and knowledge management*, CIKM '02, pages 538–548, New York, NY, USA, 2002. ACM. ISBN 1-58113-492-4. doi: 10.1145/584792.584881. URL http://doi.acm.org/10.1145/584792.584881.

J. Mrosek, S. Bussmann, H. Albers, K. Posdziech, B. Hengefeld, N. Opperman, S. Robert, and G. Spira. Content-and graph-based tag recommendation: Two variations. In *ECML PKDD Discovery Challenge 2009, CEUR Workshop Proceedings Vol. 497*, pages 189–199, Bled, Slovenia, September 2009.

Tsuyoshi Murata. Detecting communities from bipartite networks based on bipartite modularities. In *CSE (4)*, pages 50–57. IEEE Computer Society, 2009a.

Tsuyoshi Murata. Modularities for bipartite networks. In Ciro Cattuto, Giancarlo Ruffo, and Filippo Menczer, editors, *Hypertext*, pages 245–250. ACM, 2009b. ISBN 978-1-60558-486-7.

Tsuyoshi Murata. Detecting communities from tripartite networks. In Michael Rappa, Paul Jones, Juliana Freire, and Soumen Chakrabarti, editors, *WWW*, pages 1159–1160. ACM, 2010. ISBN 978-1-60558-799-8.

Nicolas Neubauer and Klaus Obermayer. Towards community detection in k-partite k-uniform hypergraphs. In *Workshop on Analyzing Networks and Learning with Graphs (NIPS 2009)*, Whistler, BC, Canada, 2009.

M. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 74(3), 2006.

M. E. J. Newman. Coauthorship networks and patterns of scientific collaboration. *Proceedings of the National Academy of Science of the United States (PNAS)*, 101:5200–5205, 2004a.

M. E. J. Newman. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69(6):066133, 2004b.

M. E. J. Newman. Communities, modules and large-scale structure in networks. *Nature Physics*, 8:25–31, 2012.

Mark Newman. Random graphs as models of networks. *Handbook of Graphs and Networks*, pages 35–68, 2005. doi: 10.1002/3527602755.ch2.

M.E. J. Newman and M M. Girvan. Finding and evaluating community structure in networks. *Physics review E*, 69:026113:1–022613:15, 2004.

Vincenzo Nicosia, Ginestra Bianconi, Vito Latora, and Marc Barthelemy. Growing multiplex networks. *arXiv preprint arXiv:1302.7126*, 2013.

Q. Ou, Y. D. Jin, T. Zhou, B. H. Wang, and B. Q. Yin. Power-law strength-degree correlation from resource-allocation dynamics on weighted networks. *Phys. Rev. E*, 75:021102, 2007. doi: 10.1103/PhysRevE.75.021102. URL http://dx.doi.org/10.1103/PhysRevE.75.021102.

Vladimir Ouzienko, Yuhong Guo, and Zoran Obradovic. Prediction of Attributes and Links in Temporal Social Networks. In *ECAI*, pages 1121–1122, 2010. doi: 10.3233/978-1-60750-606-5-1121. URL http://dblp.uni-trier.de/db/conf/ecai/ecai2010.html#OuzienkoGO10.

G. Palla, I. Derônyi, I. Farkas, and T. Vicsek. Uncovering the overlapping modular structure of protein interaction networks. *FEBS Journal*, 272:434, 2005.

Symeon Papadopoulos, Yiannis Kompatsiaris, Athena Vakali, and Ploutarchos Spyridonos. Community detection in social media - performance and application considerations. *Data Min. Knowl. Discov.*, 24(3):515–554, 2012.

Chengbin Peng, Tamara G Kolda, and Ali Pinar. Accelerating community detection by using k-core subgraphs. *CoRR*, 1403.2226, March 2014.

Pascal Pons and Matthieu Latapy. Computing communities in large networks using random walks. *J. Graph Algorithms Appl.*, 10(2):191–218, 2006.

Alexandrin Popescul and Lyle H Ungar. Cluster-based concept invention for statistical relational learning. In Won Kim, Ron Kohavi, Johannes Gehrke, and William DuMouchel, editors, *KDD*, pages 665–670. ACM, 2004. ISBN 1-58113-888-1.

Manisha Pujari and Rushed Kanawati. Link prediction in multiplex bibliographical networks. *International Journal of Complex Systems in Science proceedings of NETWORKS 2013*, 2013.

Usha N. Raghavan, Roka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76:1–12, September 2007.

Matthew J Rattigan and David Jensen. The case for anomalous link discovery. *ACM SIGKDD Explorations Newsletter*, 7(2):41–47, 2005.

M. Rosvall, D. Axelsson, and C. T. Bergstrom. The map equation. *Eur. Phys. J. Special Topics*, 13:178, 2009.

Camille Roth, Soong Kang, Michael Batty, and Marc Barthelemy. A long-time limit for world subway networks. *Journal of The Royal Society Interface*, 2012. ISSN 1742-5662. doi: 10.1098/rsif.2012.0259.

D. Sculley. Rank aggregation for similar items. In *Proceedings of the Seventh SIAM International Conference on Data Mining (SDM)*, April 2007.

D. Shah and T. Zaman. Community detection in networks: The leader-follower algorithm. In *Workshop on Networks Across Disciplines in Theory and Applications, NIPS*, 2010.

Han Hee Song, Tae Won Cho, Vacha Dave, Yin Zhang, and Lili Qiu. Scalable proximity estimation and link prediction in online social networks. In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, pages 322–335. ACM, 2009.

Sucheta Soundarajan and John Hopcroft. Using community information to improve the precision of link prediction methods. In *Proceedings of the 21st international conference companion on World Wide Web*, pages 607–608. ACM, 2012.

Myra Spiliopoulou. Evolution in social networks: A survey. In *Social network data analytics*, pages 149–175. Springer, 2011.

K. Subbian and P. Melville. Supervised rank aggregation for predicting influence in networks. In *Proceedings of the IEEE Conference on Social Computing (SocialCom-2011).*, Boston, October 2011.

Peng-Gang Sun and Lin Gao. A fast iterative-clique percolation method for identifying functional modules in protein interaction networks. *Frontiers of Computer Science in China*, 3(3):405–411, 2009.

Dan Suthers, Judi Fusco, Patricia Schank, Kar-Hai Chu, and Mark Schlager. Discovery of community structures in a heterogeneous professional online network. In *System Sciences (HICSS), 2013 46th Hawaii International Conference on*, pages 3262–3271. IEEE, 2013.

Lionel Tabourier, Anne-Sophie Libert, and Renaud Lambiotte. Rankmerging: Learning to rank in large-scale social networks. In *Proceedings of the 2nd International Workshop on Dynamic Networks and Knowledge Discovery co-located with the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2014)*, volume 1229, 2014.

Fei Tan, Yongxiang Xia, and Boyao Zhu. Link prediction in complex networks: A mutual information perspective. *PLoS ONE*, 9(9):1–8, 2014.

Lei Tang and Huan Liu. *Community detection and mining in social media.* Synthesis Lectures on Data Mining and Knowledge Discovery. Morgan & Claypool Publishers, 2010.

Benjamin Taskar, Ming Fai Wong, Pieter Abbeel, and Daphne Koller. Link prediction in relational data. In Sebastian Thrun, Lawrence K Saul, and Bernhard Schölkopf, editors, *NIPS*. MIT Press, 2003. ISBN 0-262-20152-6.

Jeffrey Travers and Stanley Milgram. An experimental study of the small world problem. *Sociometry*, 32(4):425–443, 1969. ISSN 00380431. doi: 10.2307/2786545.

Michel Truchon. An extension of the condorcet criterion and kemeny orders. *Cahier*, 9813, 1998.

Anurag Verma and Sergiy Butenko. Network clustering via clique relaxations: A community based approach. *Graph Partitioning and Graph Clustering*, 588:129, 2012.

Chao Wang, Venu Satuluri, and Srinivasan Parthasarathy. Local probabilistic models for link prediction. In Yong Shi and Christopher W Clifton, editors, *Seventh IEEE International Conference on Data Mining (ICDM)*, pages 322–331. IEEE, October 2007.

Lu Wang, Yanghua Xiao, Bin Shao, and Haixun Wang. How to partition a billion-node graph. In *Data Engineering (ICDE), 2014 IEEE 30th International Conference on*, pages 568–579. IEEE, 2014.

Xi Wang and Gita Sukthankar. Link prediction in multi-relational collaboration networks. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 1445–1447. ACM, 2013.

Stanley Wasserman and Katherine Faust. *Social Network Analysis: Methods and Applications*, chapter Social Network Analysis in the Social and Behavioral Sciences, pages 3–27. Number 8. Cambridge University Press, 1994.

D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, 1998.

Scott White and Padhraic Smyth. A spectral clustering approach to finding communities in graph. In *SDM*, 2005.

Zied Yakoubi and Rushed Kanawati. Licod: A leader-driven algorithm for community detection in complex networks. *Vietnam Journal of Computer Science*, 1 (4):241–256, 2014.

Bowen Yan and Steve Gregory. Detecting communities in networks by merging cliques. In *Intelligent Computing and Intelligent Systems, 2009. ICIS 2009. IEEE International Conference on*, volume 1, pages 832–836. IEEE, 2009.

Dawei Yin, Liangjie Hong, and Brian D. Davison. Structural link analysis and prediction in microblogs. In *Proceedings of the 20th ACM international conference on Information and knowledge management - CIKM '11*, page 1163, New York, New York, USA, 2011. ACM Press. ISBN 9781450307178. doi: 10.1145/2063576.2063743. URL http://dblp.uni-trier.de/db/conf/cikm/cikm2011.html#YinHD11.

Yizhou Sun, Rick Barber, Manish Gupta, Charu C. Aggarwal, and Jiawei Han. Co-author relationship prediction in heterogeneous bibliographic networks. In *Advances on social network Analysis and mining (ASONAM)*, Kaohsiung, Taiwan, 2011.

H.P. Young and A. Levenglick. A consistent extension of condorcet's election principle. *SIAM Journal on Applied Mathematics*, 35(2), 1978. ISSN 00361399. doi: 10.2307/2100667.

Xiao Yu, Quanquan Gu, Mianwei Zhou, and Jiawei Han. Citation prediction in heterogeneous bibliographic networks. In *SDM*, pages 1119–1130, 2012.

Jiawei Zhang and S. Yu Philip. Link prediction across heterogeneous social networks: A survey. Technical report, University of Illinois, Chicago, 2014. URL http://www.cs.uic.edu/~jzhang2/files/2014_survey_paper.pdf.

Zi-Ke Zhang, Tao Zhou, and Yi-Cheng Zhang. Personalized recommendation via integrated diffusion on user-item-tag tripartite graphs. *CoRR*, abs/0904.1989, 2009.

Tao Zhou, Linyuan Lu, and Yi-Cheng Zhang. Predicting missing links via local information. *The European Physical Journal B*, 71(4):623–630, October 2009.

Xiaojin Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.