Exploiting Tie-strength and Structure Towards Link Prediction in Social Networks

Niladri Sett

Exploiting Tie-strength and Structure Towards Link Prediction in Social Networks

Thesis submitted in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

by

Niladri Sett

Under the supervision of

Dr. Sanasam Ranbir Singh and Prof. Sukumar Nandi

Department of Computer Science and Engineering

INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

Guwahati 781039, India

March 21, 2017

Declaration

I certify that

a. The work contained in this thesis is original, and has been done by myself under the general supervision of my supervisors.

b. The work has not been submitted to any other institute for any degree or diploma.

c. Whenever I have used materials (data, theoretical analysis, results) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references.

d. Whenever I have quoted written materials from other sources, I have put them under quotation marks and given due credit to the sources by citing them and giving required details in the references.

Place: IIT Guwahati
Date:

Niladri Sett
Research Scholar
Department of Computer Science
and Engineering,
Indian Institute of Technology Guwahati,
Guwahati 781039, India

CERTIFICATE

This is to certify that this thesis entitled "**Exploiting Tie-strength and Structure Towards Link Prediction in Social Networks**" being submitted by Mr. Niladri Sett to the Department of Computer Science and Engineering, Indian Institute of Technology Guwahati, is a record of bona fide research work under my supervision and is worthy of consideration for the award of the degree of Doctor of Philosophy of the Institute.

The results contained in this thesis have not been submitted in part or full to any other university or institute for the award of any degree or diploma.

Dr. Sanasam Ranbir Singh
Dept. of Computer Sc. and Engg.
Indian Institute of Technology Guwahati
Guwahati-781039, Assam, India
Place: IIT Guwahati
Date:

Prof. Sukumar Nandi
Dept. of Computer Sc. and Engg.
Indian Institute of Technology Guwahati
Guwahati-781039, Assam, India
Place: IIT Guwahati
Date:

Acknowledgements

First I would like to convey my deepest gratitude to my parents, whose love, support, encouragement, inspiration, and sacrifice have made my success possible. Their simple but respectable life-style has provided me liberty to be fearless and explore new things in life. Probably, I would have never decided to pursue a PhD, and able to submit my thesis, if they were otherwise. I also thank my other family members, especially my cousins, for their unconditional love and support, which they have been sharing with me throughout my life.

This thesis is a culmination of a perfect working relationship with my supervisors, Dr. Sanasam Ranbir Singh and Prof. Sukumar Nandi, to whom I am very grateful. They provided me immense support during my journey towards PhD, and generously paved the way for my development as a research scientist. I thank my supervisors for sharing their knowledge and novel ideas with me, which have helped me throughout this process. I am also highly grateful to my Doctoral Committee members, Dr. V. Vijaya Saradhi, Dr. Ashish Anand and Dr. Amit Awekar for sharing their insightful comments and suggestions on my research work. I also thank my friends Deepak Mangal, Akash Anil, Saptarshi Basu, Subhrendu Chattopadhyay and Devesh for collaborating with me in research during my PhD tenure. I express my sincere thanks to Prof. P. Bhaduri, Prof. G. Sajith, Prof. S. V. Rao, Prof. S. B. Nair, the former Heads, and Prof. Diganta Goswami, the present Head of the Department of Computer Science and Engineering, for providing a nice research environment in the department, and support my research works in many ways. I also thank department's scientific staffs Mr. Bhriguraj Bora and Mr. Nanu A. Kachari for extending their helping hands in solving many technical issues which I faced during my PhD.

I cherished my stay in the institute largely due to my friends. I thank Dibyendu,

Kartick, Sandip, Soumyadip, Subhendu, Aswini, Kaushik, Santu, Himadri, Barun, Kalyan (Manna and Sinha), Purnendu-da, Mandar, Murali, Arnab, Sayantan, Rajib, Subhrangsu, Abhijit, Suddhasil-da, Shounak, Subhrendu, Biswajit-da, Ranajit, Rahul, Sanjit, Rohit, Mrityunjay (Singh and Kunwar), Sirshendu, Nilkanta, Sibaji, Shuvendu, Satish, Awnish, Durgesh, Rajesh, Mayank, Shilpa, Pravati, Debanjan, Sashi, Amrita-madam, Ashok-sir, Shrinivasa-sir, Tripathi-sir and many others for their encouragement and delightful company. I also thank my friends of OSINT lab for their support. I am deeply indebted to my college friends and seniors: Bodhisattwa, Tanmoy, Debasis, Saptarshi, Biplab, Rasbihari, Chinmoy, Arka, Samir-da, Soumen-da and others, who always inspired me during my PhD.

The development of this thesis would not have been possible without financial support from the MHRD, Govt. of India. I would also like to thank Department of Computer Science and Engineering, and Department of Science and Technology, Govt. of India sponsored project ISEA phase II for providing me financial support after the tenure of my MHRD fellowship was over.

Place: IIT Guwahati

Date: **Niladri Sett**

Abstract

Analysis of complex network has emerged as a booming research area since last decade. Social network, a type of complex network, has gained attention from the contemporary researchers due to the abundance of social network data in the Web in recent times. Rapid increase in the number of subscribers to the social platforms (such as blogs, dating sites, friends making sites) provided by the Web has revealed unseen human relationships, and motivated the researchers to make good use of this. This thesis deals with an important problem of social network analysis (also of complex network analysis): *link prediction*.

Given a social network, the link prediction problem predicts new relationships which will appear in future. Homophily, i.e., similarity between two individuals influences new connections. This work models homophily by combining link strength and structure of the network towards link prediction. Link strength is encoded in link weight in several ways, which is derived from pattern of dyadic interaction between two nodes. Structural homophily is captured by traditional proximity based link prediction methods like common neighbor, Jaccard's co-efficient, Adamic/Adar, etc. Several social network properties, such as, reciprocative nature of relationships, temporal change of homophily between actors, information flow between actors through heterogeneous paths, etc., are investigated in this regard, and new link prediction methods are proposed by exploiting dyadic interactions and structural similarity. This thesis also proposes a time aware method to predict broken relationships, and analyzes their effect in link prediction. It further devises methods to deal with sparsity problem of dyadic time-series towards effective link prediction. Using unsupervised and supervised methods, rigorous experiments are performed over various real and longitudinal social network datasets to demonstrate the effectiveness of the proposed methods over the existing ones.

Contents

List of Figures

List of Symbols

$\Gamma(x)$	Set of neighbors of node x in a network
$s(x)$	Weighted degree or strength of node x in a network
$CN(x, y)$	Common neighbor score of node pair x and y
$JC(x, y)$	Jaccard's coefficient score of node pair x and y
$AA(x, y)$	Adamic/Adar score of node pair x and y
$RA(x, y)$	Resource allocation score of node pair x and y
$PA(x, y)$	Preferential attachment score of node pair x and y
$ARIMA(p, d, q)$	ARIMA model, where p = degree of auto-regressive polynomial, q = degree of moving average polynomial, d = stationarity parameter
$ETS(A, N, N)$	Simple exponential smoothing model
$ETS(A, A, N)$	Exponential smoothing model with additive trend
Δ	Span of time window, when a dynamic network is divided into parts
$\mathbb{N}$	Set of natural numbers
$\mathbb{R}$	Set of real numbers
$\mathscr{D}(x, y)$	Dyadic time-series of link (x, y)
$\mathscr{S}(x)$	Dyadic time-series of node x
G^t	Graph at time t
$G^{t-\Delta i}$	Snapshot of G^t at $t - \Delta i$

Chapter 1

Introduction

1.1 Complex network

With advancement of computational capability, the study of complex network [1] has gained substantial attention from large group of researchers in exploring and understanding the characteristics of real world networks. Complex networks mostly refer to real world networks, which are often characterized by non-trivial attributes such as large number of interacting entities, self organization, heterogeneity, evolution, etc. Real world networks are large, and contain complex structural relationships between their entities. The complex structure refers to the presence of emergence properties in the network. These properties are the consequences of interactions of different attributes of the network. One such property is the presence of *small world* characteristics in social communities [2]. Some of the popular real networks (complex networks) are *biological networks*, *social networks*, *semantic networks*, *World Wide Web (WWW)*, etc. Biological networks maintain relationships among biological objects, such as, protein, enzyme, etc.; social networks maintain social relationships among human beings; semantic networks connect related concepts; and WWW connects web pages through hyper-links. A network is represented by a set of entities (known as vertices or nodes) and a set of connections between the entities (known as edges or ties).

In recent past, several studies on complex network analysis have been reported in

literature. Seminal discoveries of complex networks reveal that real world networks follow *power law degree distribution* [3], *community structure* [4], *high clustering coefficient* [2], *low expected geodesic distance among node pairs* [2], *existence of motifs* [5], etc. Such discoveries have motivated researchers from diverse areas to apply the concept of complex network analysis to solve real world problems in various domains. Some of the popular application areas in the complex network analysis are; (i) `epidemic`: understanding spreading phenomenon of infectious diseases [6], (ii) `community detection`: identifying functional groups in metabolic networks [7], (iii)`hazard warning`: exploiting food-web for environmental warnings [8], (iv)`planning and policy`: urban planning [9], (v)`fault detection`: detecting vulnerabilities in power grid [10], (vi)`spam detection`: e-mail spam detection [11], (vii)`link mining`: link prediction in social network [12], etc.

1.2 Social network

A social network is often characterized by a network, where nodes are social elements or actors such as humans, and ties are social relationship between two actors. Analysis and discovery of relationship patterns in social networks is termed as *social network analysis (SNA)*. Study on social network can be traced back in 1956 [13], decades before the birth of Web based social platforms. Earlier, SNA researchers used to collect dataset manually either through survey questionnaires or personal interactions [14, 15]. With the increase in popularity and usage of Web by general public, availability of social network data has increased rapidly. The most primitive form of social network in the Web is e-mail network, which is formed by e-mail exchanges among people. Due to technological advancement in big data technologies, network protocols, and computer hardware, etc., the Web has been able to accommodate the mass population. It has given access of hand-held electronic gadgets to the people, through which they access *On-line social networking sites (OSNs)*. OSNs allow people to make friendships (in social networking site like `https://www.facebook.com/`), express their opinion (in on-line blogging and micro-blogging sites like `https://twitter.com/`), date with selected partners (in on-line dating sites like `http://in.match.com/`), etc. Apart from OSNs and e-mail networks,

researchers have also considered other forms of social networks, such as co-authorship networks [16], bibliographic citation network [17], criminal networks [18], etc.

1.2.1 General characteristics of social networks

Social network inherits the fundamental properties of complex networks. However, distinctive feature of SNA is characterized by temporal dynamics and complex nature of human relationships like friendship, enmity, kinship, co-operation, collaboration, etc. Few of the distinctive properties of social network are briefed below. Some of these properties fit for complex networks also, in general.

- **Community formation [4]:** Humans form groups within their social circles. Groups are closely knit, with dense connection setup, where inter-group links are few as compared to intra-group links.

- **Strength of tie [19]:** Strength of ties in social network vary with degree of attachment between the end nodes. Granovetter [19] has found that node-pairs inside a social community connect with *strong* ties, and inter-community node-pairs connect with *weak* links. He has referred the inter-community links as *bridges*.

- **Triadic closure property [19]:** It says that if two nodes have a common connection, then there is a high chance that they are also connected.

- **Small world phenomenon [2]:** Expected geodesic distance between two nodes in a social network is less, typically is proportional to *logarithm* of the number of nodes in the network.

- **Preferential attachment [3]:** Preferential attachment property of social network tells that in evolving networks new nodes are attracted towards highly connected nodes, making them richer in terms of number of connections. It results in *power law degree distribution* for the nodes in social networks, where some nodes play as *central* nodes with very high number of connections.

- **Dense neighborhood [2]:** Neighborhood of nodes in social networks are dense, i.e., neighbors of a node are highly interconnected among themselves. It is measured by *clustering co-efficient* index.

- **Temporal evolution [20]:** Social networks evolve with time. Nodes move between communities as they change their affiliations. New nodes and links appear with time, and some nodes and links are withdrawn.

- **Information diffusion [21]:** Information such as rumor, campaign, etc., are generated from a small set of nodes, and is spread to a large population through social interactions.

- **Heterogeneity [22]:** Human beings participate in various social circles, where they make several kinds of ties. Moreover, in each social circle they play diverse roles.

1.2.2 Commonly Studied Social Network Analysis Problems

Followings are some of the commonly studied research problems in social network analysis.

- **Finding central nodes [23]:** Finding central nodes in a social network is an age old problem in SNA. Central nodes play important role in the connectivity of network. They control information dissemination and the spread of influence inside a network [24]. Some popular centrality measures are degree centrality, betweenness centrality, eigenvector centrality, etc.

- **Community detection [4]:** The problem of finding functional groups or community is extensively studied [25]. People with similar interests form community. In a nutshell, the community detection task aims at finding such social communities, their evolution, and interplay of different communities.

- **Link prediction [12]:** Link prediction problem in social network aims at determining hidden or future relationships between actors. Hasan et al. [26] have compiled a survey on link prediction in social network.

- **Modeling information diffusion [27]:** Modeling the process of information diffusion in social network helps in understanding how information, ideas, influence, etc., propagates in social networks.

- **Modeling evolution [28]:** Modeling evolution of social network has attracted researchers in recent times. It has been studied in different granularities, starting from microscopic level [29] (temporal arrival of nodes and links) up-to evolution of groups [30].

These tasks are not independent from each other, rather results obtained from each task are used to facilitate in understanding others. One such example is: finding central nodes may help in understanding the information diffusion process in social network. This thesis focuses on *link prediction problem* (formally defined in Section 1.3). Among the wide spectrum of real world networks, this thesis focuses on social network: *interaction network*, *email network* and *co-authorship network*, in particular.

1.3 Link prediction

Link prediction is an important task in network science because of its wide range of real world applications in a variety of fields. The task of link prediction is applicable for any social, biological or information system, which can be naturally described as a network. It detects hidden links or future links from the observed part of the network. In social network, link prediction methods can effectively and efficiently help individuals to find companions, assistants, or friends [12, 31]. It helps in retrieving relevant documents of a given query in information retrieval systems [32]. It can assist (homeland) security agencies to precisely focus their efforts on probable relationships in malicious (terrorist) networks [33, 34]. In medicine and biology, link prediction can be used to effectively find non-trivial relationships and associations between biological entities [35]. Considering such a wide spectrum of application domains, link prediction problem has attracted attention from a large community of researchers. Though link prediction is an old research problem, new challenges arise due to the emergence of new forms of real world data (from

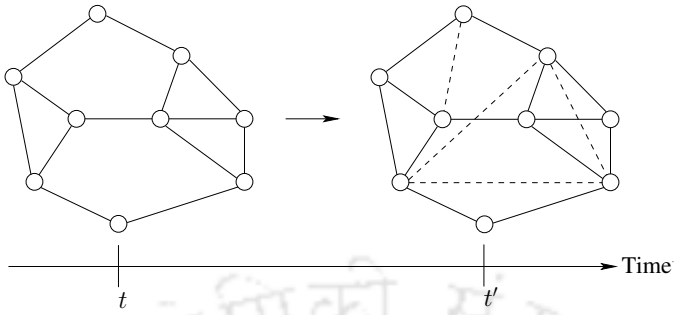


Figure 1.1: Graphical illustration of the link prediction problem

homogeneous to heterogeneous data, from social to biological, from static to evolving, etc.). Such diversity has motivated us to explore link prediction as the research problem in this thesis.

1.3.1 Problem Definitions

In literature, link prediction problem has been explored from two perspectives; (i) given a network, identifying the missing relationships between two nodes [32, 36], and (ii) given a network, predicting the relationship that will emerge in future [12, 34]. The second perspective is appropriate when the network is *longitudinal*, i.e., it evolves with time. Social networks are longitudinal in nature, and in this thesis we consider the link prediction in social network in particular. So, in this thesis, the link prediction problem is considered as follows: *given a network of individuals, it predicts the links which will appear in future among them.* Figure 1.1 provides a graphical illustration of the link prediction problem in social networks. It shows two snapshots of an evolving network at time t and t' respectively, where t' follows t. The links which newly appear in the network at time t' are shown in dotted lines. The link prediction task is to predict those links correctly.

Because of the diversified nature of social networks, link prediction problem in social network needs to be studied from various perspectives. This thesis focuses on the following five important aspects.

- **Link weight:** Link weight often encodes *strength of relationship* between two actors.

This thesis analyzes the influence of link weight on link prediction in several ways.

- **Reciprocal link:** Relationships may be one-sided or reciprocative. We exploit this behavior in link prediction.
- **Multi-relational:** People belong to multiple social circles. They engage themselves in various kind of relationships in those circles. Information regarding one kind of relationships may help in predicting relationships of other kinds, which this thesis explores.
- **Disappearing links:** This thesis explores the effect of disappearing links, i.e., breaking down of relationships, in link prediction.
- **Temporal:** Temporal dynamics of network is exploited in this thesis to help link prediction. History of dyadic interaction between two nodes and change of network topology with time are considered to represent temporal dynamics.

1.4 Motivation and scope of the thesis

The motivation of this thesis buds from a fundamental social network phenomenon, *homophily*. In the classic review titled "Birds of a feather: Homophily in social networks", McPherson et al. [37] write:

"Similarity breeds connection."

Homophily, the frequently used term by sociologists to represent similarity between two individuals in a social structure, is the fundamental cause of building relationships or social ties. Similarity may be in terms of their profession, geographical location, age, mother tongue, their geodesic distance in the *small world*, etc. Social networks are dynamic in nature, where new nodes and links appear with time. Communication of any type usually indicates link formation. Probability of building new relationships (i.e., formation of a link) is positively correlated with the similarity between two individuals or nodes, i.e., similarity influences link prediction. Exploiting underlying topology of network towards

link prediction is prevalent in literature. Most of such methods focus on shortest paths between the target nodes, and shortest paths with smaller path lengths influence future connectivity more than the larger ones. Therefore, the paths of length two influence link prediction most. The simplest two-length path based link prediction method is *common neighbor* (CN) [12], which counts the number of two-length paths, i.e., number of common neighbors between the target nodes[1], and higher the count implies higher chance of getting connected in future. *Jaccard's coefficient (JC), Adamic/Adar (AA), resource allocation (RA)* [12,38], etc., are variants of CN. *Triadic closure property* [19] of social network is the building block of these methods. It says that, given three nodes x, y and z in a network, if there exists links between x—z and y—z, then there is a chance that x and y also will be connected. In other ways, if nodes x and y have a common friend z, z may introduce them with each other to form a new connection x—y. The chance of establishment of that connection increases as *strength* of the ties x—z and y—z increases, according to *strong triadic closure property* [19]. Total number of communications (synonymously, frequency of interaction) between the end nodes is the state-of-the-art technique for quantifying strength of ties [39]. The strong triadic closure property has inspired researchers [40, 41] to propose weighted versions of the link prediction methods mentioned above, where link weight represents strength of the tie. Lichtenwalter et al. [42] have proposed a supervised learning framework for link prediction, which effectively utilizes common neighbor based link prediction methods as features. Throughout the thesis, we shall use the term "strength of tie" (and the link weight derived from it) to refer the degree of association between two connected nodes derived from their dyadic activities.

There exist conflicting results on the effect of link weight on link prediction. Murata et al. [40] have observed that weighted link prediction methods perform better than their unweighted counterparts, where Lü et al. [41] have observed otherwise. Both of them have considered frequency of interaction as the link-weight, and performed experiments on different sets of networks. It raises following questions.

[1]Target nodes are the two nodes, whose likelihood to be connected in future is being investigated

- Does the effect of link weight on link prediction depends on the network properties, and vary over different classes of networks?
- Is it possible to devise link-weighting techniques, which make the weighted prediction methods perform better than the unweighted ones in every network?
- Is it possible to embed social network properties like reciprocity, heterogeneity, dynamics, etc. in link weight and prediction methods to enhance the performance of the link prediction?

We critically analyze these questions next.

Reciprocity: In reality, social networks are diverse in nature. Obviously, nodes are represented by humans, but an individual may be included in various social circles, which results in a wide range of relationship types. Each relationship type defines a social network. Some relationships like *friendship, collaboration, etc.* are inherently *reciprocative*, where they build undirected networks. On the other hand, networks like *e-mail*, which are constructed through e-mail exchanges, are directed. In some directed networks, which are called *information networks*, the nodes may have hierarchical features. In this case, the flow of information usually follows a hierarchy. The *follower-followe* relationship of Twitter on-line social network (OSN) is one such example. The difference in link prediction performance over various network types needs to be analyzed and justified, as far as weighted prediction methods are concerned. Moreover, as the intuitive meaning and structural nature of links vary over types of relationship, the representation of strength of ties also should be different over them. Therefore, tuning the existing weighted prediction methods, and devising new link-weighting techniques are required.

Heterogeneity: A social network can also be built by combining heterogeneous types of relationships. We choose to refer them as *multi-relational*. Let us explain multi-relationality with an example of network created by researchers in scientific communities. When multiple researchers write a paper collaboratively, they become connected with *co-authorship* links. Similarly, two researchers may also be connected through another type of link, say *conference* when they attend same conference. So, in this network,

link prediction problem may be defined over either co-authorship or conference link-type. To predict one type of link, flow of information through links of other types should be taken into consideration to enhance the prediction performance. For an example, if two researchers have not published any paper together, but meeting at a conference venue may inspire them to collaborate in future. Recently, researchers have started exploring in this direction, which we call *multi-relational link prediction*. Most of the popular studies in this direction [43–48] have devised prediction models by exploiting graph topological features. Thorough analysis of the effect of incorporating dyadic strength in prediction models remains unexplored.

Temporal information: Initial studies on link prediction focus on static networks. These studies construct a static graph by ignoring *temporal* information embedded in it, and apply several unsupervised and supervised techniques to predict future links. Lü et al. [49] have presented a survey on such studies. However, social networks are dynamic in nature. Nodes move between different social circles. They sometimes abandon a social circle, and dissolve their relationships with others residing in that circle, or sometimes their relationships with other nodes degrade with time. On the other hand, some nodes increase their strength of participation (by adding links and increasing communications) in a circle. Incorporating temporal dynamics in the dyadic strength and topological similarity through time-series models seems logical. Recent studies [50–53] consider temporal dynamics of the underlying graph in order to predict future links. Tylenda et al. [50] have exploited time-agnostic functions to model a relationship, and used it to modify the state-of-the-art link prediction methods. Potgieter et al. [51] have used recency of a node in communicating with others as a parameter for link prediction. Tensor factorization based temporal link prediction method has been proposed in [52], where the third dimension of a tensor has been used to embed temporal information. In [53], Richard et al.have assumed that social networks evolve with stationary dynamics, and modeled the history of graph properties using auto-regressive models for dynamic link prediction. These studies discretely build their temporal models with a particular assumption on the dynamics. They do not explore the applicability of individual models over networks of varying characteristics. Neither do

they explore the incentives of dynamics, if any, in dyadic strength. Moreover, most of the existing studies (except very few like Yang et al. [44]) do not exploit heterogeneity of social networks in temporal link prediction.

This thesis focuses on developing new methods for link prediction by exploiting several network properties like reciprocity, heterogeneity and temporal dynamics. Historical communication pattern is modeled to form link weight, and combined with topological methods towards it. To incorporate temporal dynamics, state-of-the-art time-series forecasting methods are used. It also studies the effect of degrading or dismissing nature of nodes and links in a social circle on link prediction. Aiming to overcome the difficulty in modeling network evolution due to sparse nature of datasets, this thesis proposes robust and efficient methods for temporal link prediction. Proposed prediction methods are combined to produce several predictive models towards supervised link prediction.

1.5 Contribution of the thesis

Major contributions of the thesis are point-wise discussed below.

1.5.1 Influence of link-weight on link prediction methods

This work empirically analyzes the influence of link weight (frequency of interaction) on baseline link prediction methods like common neighbor (CN), Jaccard's coefficient (JC), Adamic/Adar (AA) and resource allocation (RA) [40, 41]. Experiments over ten real datasets reveal that the weighted versions of the baseline methods [40, 41] are not able to enhance prediction performance for some datasets. So, we propose new weighted methods based on different influence models, and tune existing weighted methods. Although, few of those datasets respond well to the new methods and after tuning is applied, others (specifically, the directed network datasets) do not show any significant improvements. We further perform analysis on dataset properties to justify the findings.

1.5.2 Exploiting reciprocity towards link prediction

This work exploits reciprocative nature of human relationship to enhance link prediction performance in directed networks. It first provides empirical evidence supporting "existence of high number (maximum 3) of reciprocal links in triads" in real networks using a null model. This evidence supports *strong triadic closure property* [19] under an assumption that reciprocal links are stronger than the one-way links. As triangle closing is the building block of the baseline link prediction methods, we exploit it in three ways. (a) We introduce reciprocity-aware link weighting mechanism. (b) We propose three new link prediction methods for directed networks, which exploit directions of the edges. (c) We consider proposed methods as features, and prepare several models combining them towards supervised prediction. All experiments are carried out on two real directed network datasets.

1.5.3 Effect of degraded relationships on link prediction

Existing studies on evolution of social network largely focus on addition of new nodes and links in the network. However, as network evolves, existing relationships degrade and break down, and some nodes go to hibernation or decide not to participate in any kind of activities in the network where it belongs. Such nodes and links, which we refer as "dull", may affect analysis and prediction tasks in networks. This work formally defines the problem of predicting dull nodes and links at an early stage, and proposes a novel time agnostic method to solve it. Pruning of such nodes and links is framed as "network data cleaning" task. As the definitions of dull node and link are non-trivial and subjective, a novel scheme to label such nodes and links is also proposed here. Experimental results on two real network datasets demonstrate that the proposed method accurately predicts potential dull nodes and links. This work further experimentally validates the need for data cleaning by investigating its effect on link prediction.

1.5.4 Temporal link prediction in multi-relational networks

Though initial studies consider only static snapshot of a network, importance of temporal dimension has been observed and cultivated subsequently. In recent times, multi-domain relationships between node-pairs embedded in social networks have been used to boost link prediction performance. In this work, we combine multi-domain topological features as well as temporal dimension, and propose a robust and efficient feature set called TMLP (Time-aware Multi-relational Link Prediction) for link prediction in dynamic heterogeneous networks. It combines dynamics of graph topology and history of interactions at the dyadic level, and exploits time-series model (*simple exponential smoothing* time-series forecasting method [54] is used) in the feature extraction process. Several experiments on two networks prepared from DBLP[2] bibliographic dataset show that the proposed framework outperforms the existing ones significantly in predicting future links. It also demonstrates the necessity of combining heterogeneous information with temporal dynamics of graph topology and dyadic history in order to predict future links. Empirical results find that the proposed feature set is robust against longitudinal bias. The results obtained is also compared with recent works [42, 44], which show that our method outperforms the existing ones substantially.

1.5.5 Addressing sparsity problem in dyadic time-series

In Subsections 1.5.3 and 1.5.4, we have used simple exponential smoothing model to model several time-series extracted from node and link properties of networks. Reason behind the selection of exponential smoothing has been its simplicity and efficiency in estimating model parameter, and an assumption that network dynamics is driven by its recent behavior, which is inherent in exponential smoothing. Due to inherent sparseness in time-series data extracted from network dynamics, complex time-series models sometimes behave abnormally while modeling such data, which results in over-fitting. Here we device methods to prepare a single aggregate time-series with sufficient and consistent data to represent the dynamics of all dyadic time-series, which is used to estimate the model

[2] `dblp.uni-trier.de/xml/`

parameters. Here we propose two such aggregation methods. Finally, new link prediction methods are proposed and evaluated by exploiting the aggregation methods. Unsupervised and supervised prediction is performed to demonstrate that proposed methods enhance link prediction performance substantially. Four real datasets (two directed and two undirected) are used for the experiments, and the effect of sparsity on individual forecasting methods are analyzed from the perspective of network characteristics.

1.6 Organization of the thesis

The thesis is organized as follows. **Chapter 2** provides a brief overview of the existing literature on state-of-the-art link prediction methods. It also discusses about preliminary concepts and models used in the thesis. **Chapter 3** analyzes influence of link-weight on local similarity based link prediction methods. **Chapter 4** exploits reciprocity towards link prediction to boost the performance. **Chapter 5** devices method for predicting dull nodes and links early, and analyzes the effect of pruning them over link prediction. **Chapter 6** proposes TMLP feature set by combining multi-relational and temporal dynamics towards link prediction. **Chapter 7** deals with data-sparsity in time-series data and proposes aggregation methods to propose robust and efficient feature set for link prediction. **Chapter 8** draws conclusion of the work done, and indicates future directions.

Chapter 2

Background

Majority of the contributions of this thesis reported in subsequent chapters are built around two themes; (i) general link prediction method, and (ii) future link prediction using time series modeling. Each chapter attempts to address a very specific and independent research problem in link prediction task. The literature survey related to specific research problem corresponding to different chapters is discussed with the chapter itself. Considering the diverse nature of the link prediction tasks reported in different chapters, no separate section on literature survey is provided in the thesis. This chapter aims at providing background studies on link prediction and time series modeling in general to help the readers better connect to the subsequent chapters. It limits to a basic introductory discussion on link prediction methods and times series modeling methods.

2.1 Link Prediction

We witness link prediction task to be applicable in various real world applications such as queries and documents relationship in information retrieval [32], user and product relationship in recommendation system [36], building structure of terrorist networks [33], monitoring and controlling computer viruses [55], engineering surveillance system in communication network [34], finding friendship ties in friendship networks [31], predicting future collaborators in scientific research community [12], etc. Studies in literature have

considered several ranges of approaches, such as, similarity based methods, statistical methods, spectral methods, classification based methods, etc. A brief tour on these approaches is presented in this section.

2.1.1 Similarity based approaches

These methods assign a similarity value or score to each of the node pairs (target nodes) which are potential candidates to be connected by a link in future. The node pairs with high score defined by a threshold value are chosen as the predicted future links. All of these methods follow unsupervised approach, where no training phase is involved. The similarity based link prediction methods can be classified in two: *local methods* and *global methods*. Local methods exploit localized properties of the target nodes, whereas global methods consider the whole network structure to get the similarity score. We discuss few of the popular local and global similarity based link prediction methods in this section. An exhaustive list of the similarity based link prediction methods may be found in [56]. These methods have been extended in many of the modern link prediction approaches [42, 57–60]. We give more emphasis on the similarity based approaches than the other approaches, because the link prediction methods introduced in this thesis are built upon these approaches.

A. Local methods

Local methods require the nodes of a target node pair to lie close to each other in the network (usually in two hop distance). Fortunately, in social networks, probability of two nodes being connected in future decreases rapidly with their distance, and most of the future links appear when the target nodes are in two hop distance [61]. The simplest and widely used method of this category is *common neighbor (CN)* [12]. It assigns the number of two length paths (i.e., the number of common neighbors) between two nodes as their similarity. Some variants of CN are *Jaccard's coefficient (JC), Adamic/Adar (AA), Resource allocation (RA)* [12, 38], etc. All of these methods are defined for target nodes which are in two-hop distance. Another popular local method is *Preferential attachment*

(PA) [12], which have been inspired by the *preferential attachment phenomenon* [3] of social network. Although the preferential attachment method can be applied on any pair of nodes, it is most effective when the nodes are in each other's proximity. These methods are very effective in predicting future links in spite of their simplicity. Low computational cost also makes local methods popular.

Common neighbor (CN): Idea behind the common neighbor index [12] in a social network graph is that - if the target nodes x and y have many friends in common, they are more likely to form a link in future. If $\Gamma(x)$ and $\Gamma(y)$ denote the set of neighbors of x and y respectively, the CN score is defined as below:

$$CN(x, y) = |\Gamma(x) \cap \Gamma(y)|.$$

This score is equivalent to counting the number of two length paths between x and y.

Jaccard's coefficient (JC): Concept of Jaccard's coefficient index has been borrowed from the classic paper written by P. Jaccard [62], way back in 1901. It is based on an intuition that similarity of two objects decreases if each of the objects are similar to many other objects. In networks, neighbors of a node are trivially considered as most similar to that node. Subsequently, the score of JC is given by adding a normalization factor to the CN score:

$$JC(x, y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|}.$$

Adamic/Adar (AA): Adamic/Adar measure treats each common neighbor differently. A common neighbor having less number of connections associates more closely with the target nodes than those having large number of connections. So, in AA index for link prediction, the common neighbors having less number of connections contribute more into the score. This idea of AA has been borrowed from [63], where Adamic and Adar have proposed a similarity measure between two homepages in World Wide Web. The Adamic/Adar score for link prediction is defined as:

$$AA(x, y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{log|\Gamma(z)|}.$$

Resource allocation (RA): Resource Allocation method is much like Adamic/Adar. Baseline intuition of this method has been borrowed from the resource allocation

process [64] happening inside complex networks. The RA score [38] is given by:

$$RA(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{|\Gamma(z)|}.$$

Preferential attachment (PA): Preferential attachment property of social networks says that: high degree nodes tend to get new connections with higher probability than lower degree nodes. Experimental results presented in [65] empirically validate this. The preferential attachment index for link prediction is due to Barabasi et al. [20], which has shown that if both of the target nodes have high degree, they tend to be connected in future with high probability. Hence the preferential attachment score is measured as:

$$PA(x,y) = |\Gamma(x)| \times |\Gamma(y)|.$$

B. Global methods

Global methods can be applied to any node pair in the network, and are computationally expensive. Popular global methods for link prediction are *Katz, Rooted PageRank, SimRank, etc.* [12]. Inspired by the seminal paper of Leo Katz [66], the Katz index measures a weighted sum of all paths between the target nodes, where the importance of higher length paths decay exponentially. Rooted PageRank is an application of PageRank algorithm [67], and SimRank index has been inspired by SimRank similarity measure [68]. These two global methods rely on the random walk process. Here we discuss two of the most popular global methods for link prediction namely *Katz, Rooted PageRank*.

Katz: Katz index [66] exploits all paths between target nodes, and gives exponentially higher importance to the shorter distant paths. The measure is given by:

$$Katz(x,y) := \sum_{l=1}^{\infty} \beta^l \times |paths_{x,y}^{<l>}|,$$

where $paths_{x,y}^{<l>}$ represents the set of all paths of length l between nodes x and y. $\beta > 0$ is a constant that regulates the amount of importance given to higher length paths. As $\beta \to 0$, Katz index starts behaving like common neighbor.

Rooted PageRank: Rooted PageRank link prediction algorithm has been proposed by Nowell and Kleinberg [12] that exploits PageRank algorithm [67], which is originally used

to rank web-pages according to their importance. The similarity score between the target nodes x and y is defined as the stationary probability of y under the random walk: with probability α, jump to x, and with probability $1 - \alpha$, go to random neighbor of current node.

2.1.2 Statistical and stochastic model based methods

These methods fit the given network with some statistical or stochastic models, and estimate model parameters in order to find the probability of existence of links between the target nodes. A line of research in this context has modeled the network structure by exploiting some widely accepted network properties, and then used *maximum likelihood estimation method* to learn the model parameters that are most likely to fit the observed data. The learned parameters are used to assign the likelihood of the target node pairs getting connected. Clauset et al. [69] and Guimera et al. [70] are two popular studies in this direction. Both of these studies exploit community structure of networks, where Clauset et al.exploits hierarchical structure of group formation, and Guimera et al.exploits *stochastic block models* for social network [71]. Another line of study has built probabilistic models over the observed network, and optimized some target function based on a collection of network parameters [72–74]. These models are stochastic variants of *relational models* of database theory, and assign a (conditional) probability value to each target-node pair. Other than the parametric methods mentioned above, several nonparametric methods [75–77] also have been proposed for link prediction. Miller et al. [75] have proposed a Bayesian nonparametric approach named nonparametric latent feature relational model for link prediction, which infer latent features, and simultaneously learn the links having those features. Zhou [76] also has used Bayesian nonparametric approach for link prediction, but their approach relies on class variables rather than feature variables. Sarkar et al. [77] have built nonparametric model for growth and shrinking of node neighborhood towards link prediction. The methods mentioned in this subsection are computationally expensive, because of the complex nature of the optimization functions.

2.1.3 Spectral methods

Spectral methods of link prediction [52, 78–80] deal with *graph spectrum, i.e.,* the eigenvalues of laplacian or adjacency matrix of the graph representing a network. Kunegis et al. [78] have used matrix decomposition based low rank approximation, and applied *graph kernels* [81] to produce another matrix in latent space for recommending hidden links in network. This algorithm has been extended for bipartite and signed networks in [79, 80]. Dunlavy et al. [52] have used tensor decomposition along with matrix decomposition to predict future links, where evolution of network has been encoded in the third dimension of tensor. Although matrix (and tensor) factorization is expensive, spectral link prediction methods scales well due to availability of faster algorithms (such as, Arnoldi algorithm [82]) for low rank factorization.

2.1.4 As classification problem

This class of link prediction methods [42, 44, 83–87] consider simple similarity based link prediction methods as features, and build several supervised models by combining them. These models are trained using state-of-the-art classifiers, which classify whether a node pair will be linked in future or not. This class of methods have observed substantial boost in prediction performance than the individual unsupervised methods, when the similarity measures are properly chosen. Class imbalance is an inherent issue in the link prediction problem (very few possible future links as compared to the number of links which may never appear). So, modern classification based approaches [42, 44, 87] use ensemble of classifiers (such as *random forest*) to train the examples. A trained supervised model may not always meet the performance benchmark due to the bias induced by the longitudinal nature of dynamic networks. Lichtenwalter et al. [42] have proposed an approach (Appendix A details the approach) to reduce the effect of longitudinal bias in link prediction.

In this thesis, we propose several local similarity based link prediction methods. These methods are combined to form several models, and supervised learning is used to classify future links and the links which may never appear. A large part of the thesis deals with network dynamics. We model the network dynamics with time-series forecasting methods.

We briefly describe time-series forecasting methods in the next section.

2.2 Time-series forecasting methods

In this thesis, we model temporal evolution of dyad and topological properties of social networks using time-series forecasting methods. Here we discuss about those methods in brief. The problem of time-series forecasting is formally defined as: given a sequence of time-series data $< x_1, x_2, ..., x_{T-1}, x_T >$ corresponding to discrete time instants $1, 2, ..., T-1, T$, the task is to predict x_{T+h} for $h = 1, 2, 3, ...$ This thesis adapts auto-regressive integrated moving average (ARIMA) model and exponential smoothing model, because these have been frequently applied in several real-time forecasting tasks [88], and their versatility in modeling a wide range of time-series datasets.

2.2.1 ARIMA (auto-regressive integrated moving average) model

ARIMA Model [89, 90] is the combination of auto-regressive (AR) model (with a drifting behavior) and moving average model (MA). It is of the form

$$\phi(B)(1-B)^d x_t = c + \theta(B)e_t, \tag{2.1}$$

where B is the backshift operator: $B^l x_t = x_{t-l}$; $\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - ... - \phi_p B^p$ is the auto-regressive polynomial of B; $\theta(B) = 1 - \theta_1 B - \theta_2 B^2 - ... - \theta_q B^q$ is the moving average polynomial of B; c is a constant; and e_t is white noise, interpreted as the residual of the regression-like moving average model at time instant t. This form of ARIMA model is said to be $ARIMA(p, d, q)$ model. $ARIMA(p, d, q)$ models are able to fit a wide range of classes of time series data. It can be used to model stationary[1] as well as non-stationary data, governed by the parameter d. $ARIMA(p, 0, q)$ is the stationary version of ARIMA model, and is called ARMA model ($ARMA(p, q)$). $ARIMA(p, 0, 0)$ and $ARIMA(0, 0, q)$ are equivalent to auto-regressive and moving average models, and denoted as $AR(p)$ and $MA(q)$ respectively. Random walk models can be represented by $ARIMA(0, 1, 0)$. A limitation of ARIMA models is that, they fail to represent the time-series data having

[1] with constant mean and variance

exponential properties. Here we do not discuss seasonal version of ARIMA model, because datasets used in this work do not exhibit seasonal behavior.

ARIMA model selection

If the underlying data shows stationary property, subject to some stationarity test, d is set to 0. For non-stationary data, the data is differenced[2] repeatedly till stationarity is achieved. Number of times the difference operator is applied, gives the value of d. p and q are selected by investigating the auto-correlation function (ACF) and partial auto-correlation function (PACF) of the differenced data. Once the model order has been selected, the appropriate model (coefficients of the two polynomials) is chosen by minimizing the sum of the squared errors (SSE). SSE is defined as:

$$SSE = \sum_{j=1}^{t} (x_j - x'_j)^2 = \sum_{j=1}^{t} e_j^2 = e^2, \tag{2.2}$$

where e_j gives the error in time instant j, and

$$x'_j = c + \sum_{i=1}^{p} \phi_i x_{j-i} - \sum_{i=1}^{q} \theta_i e_{j-i}. \tag{2.3}$$

Minimizing SSE is a nonlinear optimization task. Least square method is one such optimization method, which is used in this thesis.

Forecasting using ARIMA

After an appropriate ARIMA model is chosen for a given time-series, forecasting is a straight forward task. Given a time-series upto time instant T, x'_{T+1} is calculated from Equation (2.3) by substituting j with $T+1$. This value is the forecast value for the time instant $T+1$. x'_{T+h} is obtained similarly by substituting j by $T+h$ in the Equation (2.3), and using the corresponding forecast values in place of x_{T+n} for $n = 1, 2, ..., h-1$.

[2]Difference operator d is defined as: $d(x_t) = x_t - x_{t-1}$.

2.2.2 Exponential smoothing

Exponential smoothing [54] class of models have evolved from the Exponentially weighted moving average (EWMA) model, which is equivalent to $ARIMA(0,1,1)$ without the constant term in Equation (2.1). $ARIMA(0,1,1)$ with $-1 < \theta < 1$ is a non-stationary version of the moving average model mentioned earlier. When the constant term is removed, it can be written as, $x_t = (1-\theta)x_{t-1} + \theta(1-\theta)x_{t-2} + \theta^2(1-\theta)x_{t-3} + ... + e_t$, where the dependence of x_t on the past values $x_{t-1}, x_{t-2}, ...$ increase in an exponential manner from less recent to more recent observations. It is called the EWMA model, and the exponential smoothing models are variations of this model. Exponential smoothing models linearly or exponentially combine the past observations to forecast, where the values of the coefficients increase in an exponential manner from less recent to more recent observations (like EWMA). Exponential smoothing class of models can be described with three parameters: error (E), trend (T), and seasonality (S), and a particular model is called $ETS(-,-,-)$, where the first parameter stands for the error type, the second one represents the trend type, and the third tells whether seasonality is taken into consideration or not. Error can take values: additive (A) or multiplicative (M); trend can be absent (N), additive (A) or multiplicative (M); seasonality may be absent (N), additive (A) or multiplicative (M). Here we discuss two non-seasonal and linear versions of the exponential smoothing models as they are used to model network properties in the thesis.

Simple exponential smoothing model (ETS(A,N,N))

The forecast equation of the simple exponential smoothing model is given by a simple recurrence equation :

$$x'_{t+h} = \alpha x_t + (1-\alpha)x'_t \tag{2.4}$$

for $h = 1, 2, 3, ...$ Breaking the recurrence, it is simplified to:

$$x'_{t+h} = \sum_{i=0}^{t-1} \alpha(1-\alpha)^i x_{t-i} + (1-\alpha)^t x'_1. \tag{2.5}$$

$0 < \alpha \leq 1$ is called smoothing parameter. Like ARIMA, α is estimated by minimizing SSE, which considers Least square method as the optimization method.

Holt's linear trend model(ETS(A,A,N))

This model adds an additive trend parameter to simple exponential smoothing model. The forecast equation of the same can be given as :

$$x'_{t+h} = \alpha x_t + (1-\alpha)(x'_{t-1} + \tau_{t-1}) + h\tau_t \tag{2.6}$$

for $h = 1, 2, 3, ...$, where $\tau_t = \beta(x'_{t+1} - x'_t) + (1-\beta)\tau_{t-1}$ describes the trend. Like simple exponential smoothing, the parameters of Holt's linear trend model also is estimated by minimizing SSE.

2.3 Summary

This section categorically presented a brief overview of the existing link methods, and the fundamentals of time-series models, which we use in this thesis to model network dynamics. Our contributions start from the next chapter.

Chapter 3

Influence of link-weight on link prediction methods

3.1 Overview

Throughout this thesis we exploit link weight to facilitate link prediction in several ways. We start with static link weight, and move towards temporal link weight gradually. This chapter analyzes the influence of frequency of interaction based static link weight on state-of-the-art link prediction methods.

3.1.1 Background literature and motivation

Initial studies [12,38,91] on link prediction methods, such as, *Common Neighbor(CN), Jaccard's coefficient(JC), Adamic/Adar(AA), Resource Allocation(RA)*, etc., have explored topological characteristics of graphs by performing local analysis on node proximity. Majority of studies on the local analysis based link prediction methods consider unweighted graphs. However, links in typical social networks like message passing, co-authorship, friendship networks, etc., are weighted in nature. Traditionally, link weights represent association between the end nodes, i.e., strength of the tie. In the classical study [19], Granovetter has characterized links by *strong* and *weak*. Later works on social network analysis have quantified link weights with discrete variable, which is referred as *frequency*

of interaction between two actors. The meaning of frequency of interaction varies over networks. Newman [39] has characterized co-authorship links as the number of research papers written by two authors. In mobile phone communication networks, average duration of historical calls and total number of calls have been used as tie-strength [92, 93]. Ogata et al. [94] have used frequency of e-mail exchange between two users to quantify tie-strength. Frequency of interactions has also been used to predict the strength of ties in social media [95, 96]. Therefore, it seems very natural to take tie weights into consideration for the link prediction problem. However, influence of tie weights on the local analysis based link prediction methods is not clearly understood. Few studies have been reported in literature in this regard. First of such study has been reported in [40], where the authors have observed positive influence of weight on the prediction methods. However, later in some studies [41,86], conflicting results are observed, where Lü and Zhou [41] have observed a negative influence, and De Sá and Prudêncio [86] have observed a positive influence. These studies tune each of the link prediction methods like CN, AA, JC, RA to incorporate link weight in it. In all these studies, an *additive* influence model has been used for the weighted prediction methods, where influence of each common neighbor of the target nodes is quantified by summation of the weights of links, which connect the common neighbor with the target nodes. It has not been thoroughly analyzed whether the reported influence model in its current form is the best estimate of incorporating weights to the local prediction methods. It has motivated us to propose new influence models to incorporate weight in baseline link prediction methods, and investigate and analyze their performance over datasets. Like [40, 41, 86], this study focuses only on the local analysis based link prediction methods.

3.1.2 Contributions

New weighted link prediction methods are proposed here which exploit *min-flow* and *multiplicative* influence model. These weighted methods are tuned version of baseline unweighted methods mentioned earlier. From several experiments using ten datasets constructed from eight social networks, it is observed that the three models, additive,

multiplicative and min-flow respond differently over prediction methods and datasets. To study the influence of link weight on baseline prediction methods, we systematically explore it in three levels. First, the weighted prediction methods are applied over datasets. The datasets get divided in three classes according to the performance of weighted prediction methods. The classes are formed due to three distinct behavior: weighted methods consistently outperform their unweighted counterparts, unweighted methods consistently outperform their weighted counterparts, and comparative performances of unweighted and weighted methods are only marginally different. Second, we modify the weighted methods by applying tuning parameters. With a proper selection of tuning parameters, a significant boost in the performance of the weighted methods is observed. After tuning, the weighted models perform better than their unweighted counterparts even for some of the datasets, which has been observed otherwise before tuning. Third, we present a neighborhood based analysis on datasets to find out the reason behind the diversified effect of tie weight on the node proximity based link prediction methods. We propose an index that is useful in predicting the nature of influence of link weight over link prediction methods. We further extend the analysis based on density in target nodes' neighborhood[1], by introducing *odd ratio over node degree*[2]. An interesting observation follows: for the target nodes with low average odd ratio, weighted methods are suitable, and for the nodes with high average odd ratio, unweighted methods are suitable. In short, we can summarize our contributions as follows.

- Tune local proximity based link prediction methods to produce their weighted versions using min-flow and multiplicative influence models.
- Investigate the effect of three influence models: additive, min-flow and multiplicative over individual prediction methods and datasets.
- Systematically study the effect of weight on prediction methods by introducing weighted links iteratively.

[1]Nodes connected by strong ties are considered to belong to same region or community and nodes connected by weak ties are considered to belong to different regions or communities.

[2]Ratio between unweighted and weighted.

- Analyze the effect of two weight tuning methods applied over RA.
- Present a node proximity based analysis of underlying graph, and propose an index to predict the nature of influence of weight on link prediction methods for a particular dataset.
- Define degree odd-ratio and use it to propose a directive model for effective prediction.

3.2 Prediction methods: weighted and unweighted

The prediction methods CN, JC, AA and RA explore the local proximity of two nodes to estimate a predicted score. A classical comparative study of various prediction methods (including the first three) has been reported by Nowell and Kleinberg in [12]. Later in 2009, the resource allocation measure has been introduced by Zhou et al. [38]. All these measures assign a positive score to a node pair, if and only if there is at least one 2-length path between the participating nodes, i.e., the participating nodes have at least one common neighbor. Among these four methods, RA is reported to perform better in several studies [38, 41, 56], and all these studies except [41] have considered only unweighted networks.

Study on the effect of tie weights over the local analysis based node proximity measures is still not explored much. The first such study has been presented by Murata et al.in [40]. Authors have investigated the effect on three measures: CN, JC and AA using Yahoo! Chiebukuro social graph. Their results indicate positive influence of tie weights on the link prediction. However Lu et al. [41] have revisited the problem and observed conflicting results i.e., the performance of weighted measures of almost all proximity measures (CN, AA and RA) perform badly in all of the three datasets: USAir (US air transportation network), C.elegans (neural network of the nematode worms) and CGScience (co-authorship network of computational geometry). Lü et al.have further extended the study to investigate the role of weak ties, and concluded that their results have been influenced by M. S. Granovetter's weak tie theory [19], i.e., weak ties play important role in the information dissemination in social networks. In [86], the

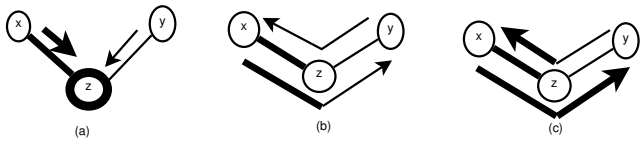


Figure 3.1: Graphical representation of different weighting models

authors have not found significant improvement in performance, while experimenting with weighted co-authorship networks. However, the performance improved when they have applied supervised approach to the weighted measures. In another recent study [97], the authors have explored face to face interaction network among researchers, and have observed that the weighted methods outperform their unweighted counterparts.

In all these studies, only an additive (linear summation) model has been used to incorporate weights. However, the additive model may not have equal effect on different prediction methods. Like existing studies, we also focus on CN, JC, AA and RA, but investigate the responses of three influence models: (i) additive, (ii) min-flow and (iii) multiplicative.

3.2.1 Three influence models

If x and y are the target nodes, the *additive* strength between x and y is bound by a common neighbor z, which is defined by a linear model $w(x,z) + w(z,y)$, where $w(-,-)$ is the symmetric edge weight connecting two nodes. If we assume that two nodes x and y have infinite supply of information through channels connecting them, $w(x,z) + w(z,y)$ represents the aggregate information received by node z from nodes x and y. Higher the volume of information z receives, tighter is the bond that z holds between x and y. Figure 3.1.(a) shows graphical representation of the additive model, where z acts as an information aggregator. Thickness of the edges represents strength of the tie.

Unlike additive model, *min-flow* defines the bonding between x and y by the channel capacity, $\min\Big(w(x,z), w(z,y)\Big)$, through z. Considering channels of different capacities, the information received by one node from another node is defined by the channel of lower

capacity. Figure 3.1.(b) shows graphical representation of min-flow model, where z acts as a flow control node between x and y.

In *multiplicative* model, node z acts as a flow booster. The incoming flow is exaggerated by many folds defined by the outflow channel capacity, i.e., $w(x,z) \times w(z,y)$. In the following subsections, we incorporate the above three influence models with each of the prediction methods and define the weighted versions.

3.2.2 Weighted prediction methods

Weighted common neighbor

Murata et al. [40] have defined the weighted common neighbor with additive influence as follows:

$$CN_A(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} (w(x,z) + w(z,y)).$$

In this, the binding strength between x and y is defined by the collective information received by all common neighbors. Similarly, we define the min-flow and multiplicative model of CN respectively, with the collective amount of information received by one node from another through all common neighbors, as follows:

$$CN_{MF}(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \min\left(w(x,z), w(z,y)\right),$$

where $\min\left(w(x,z), w(z,y)\right)$ defines the channel capacity passing through z, and

$$CN_M(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} (w(x,z) \times w(z,y)).$$

In multiplicative, z acts as a booster node. The incoming strength towards z is either boosted or filtered by the outgoing channel capacity.

Weighted Adamic/Adar

Murata et al. [40] have defined the additive weighted measure of AA as follows,

$$AA_A(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{w(x,z) + w(z,y)}{log(1 + s(z))},$$

where $s(z) = \sum_{z' \in \Gamma(z)} w(z, z')$ is the additive strength or weighted degree of node z. Like CN, we define the min-flow and multiplicative version of weighted AA as below.

$$AA_{MF}(x, y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{\min\left(w(x, z), w(z, y)\right)}{log(1 + s(z))},$$

and

$$AA_{M}(x, y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{w(x, z) \times w(z, y)}{log(1 + s(z))}.$$

In the equations of weighted versions of AA, one is added with $s(z)$ to avoid negative score.

Weighted resource allocation

Lü et al. [41] have defined the additive weighted measure of RA as follows:

$$RA_{A}(x, y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{w(x, z) + w(z, y)}{s(z)}.$$

We introduce the min-flow and multiplicative weighted methods as:

$$RA_{MF}(x, y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{\min\left(w(x, z), w(z, y)\right)}{s(z)},$$

and

$$RA_{M}(x, y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{w(x, z) \times w(z, y)}{s(z)}.$$

Weighted Jaccard's coefficient

We formulate additive, min-flow and multiplicative weighted methods of JC as follows:

$$JC_{A}(x, y) = \frac{\sum_{z \in \Gamma(x) \cap \Gamma(y)} \left(w(x, z) + w(z, y)\right)}{s(x) + s(y) - \sum_{z \in \Gamma(x) \cap \Gamma(y)} \min\left(w(x, z), w(z, y)\right)},$$

$$JC_{MF}(x, y) = \frac{\sum_{z \in \Gamma(x) \cap \Gamma(y)} \min\left(w(x, z), w(z, y)\right)}{s(x) + s(y) - \sum_{z \in \Gamma(x) \cap \Gamma(y)} \min\left(w(x, z), w(z, y)\right)},$$

and

$$JC_{M}(x, y) = \frac{\sum_{z \in \Gamma(x) \cap \Gamma(y)} \left(w(x, z) \times w(z, y)\right)}{s(x) + s(y) - \sum_{z \in \Gamma(x) \cap \Gamma(y)} \min\left(w(x, z), w(z, y)\right)}.$$

The denominator $s(x) + s(y) - \sum_{z \in \Gamma(x) \cap \Gamma(y)} \min\left(w(x, z), w(z, y)\right)$ represents the weighted equivalent of $|\Gamma(x) \cup \Gamma(y)|$ as $|\Gamma(x) \cup \Gamma(y)| = |\Gamma(x)| + |\Gamma(y)| - |\Gamma(x) \cap \Gamma(y)|$. Here we consider min-flow weighted version for the subtraction.

Table 3.1: Characteristics of the datasets

Datasets	#Nodes	#Links	Avg CC	Avg Degree
dblp	339223	969287	0.643	5.715
enron	76548	297224	0.153	7.766
newman	16264	47594	0.562	5.853
oclinks	1899	13838	0.109	14.574
openflights	2939	15677	0.453	10.668
astro	16046	121251	0.665	15.113
hep-th	7610	15751	0.486	4.140
netscience	1461	2742	0.694	3.754

3.3 Datasets

We consider eight datasets of diverse characteristics in our experiments. A brief characteristics of the datasets are presented in Table 3.1. `dblp` and `enron` networks are constructed locally from raw data that includes time-stamps. So, for these two networks, actual future links are considered to test the performance of the prediction methods. For other datasets, ten-cross validation is used for testing. Considering the nature of the networks, the datasets are divided into three groups. A brief discussion is presented below.

1. Collaboration networks

   - `dblp`: It is a co-authorship network of computer scientists. Raw data has been downloaded from the web-link `dblp.uni-trier.de/xml/` in .xml format. It contains the publication information in the field of computer science from the year of 1936 upto the starting of 2014. Each publication is maintained using tags like $< inproceedings >$ (for conference publications), $< article >$ (for journal publications), etc. Inside such tags, a number of $< author >$ tags

are present, which provide the set of authors who have collaborated in that paper. In order to build our datasets from the raw data, we have considered the publications over five years (between 2001 and 2005) for constructing the co-authorship graph. Publications of next two years are used to collect true future links. Based on previous studies found in the literature, we create two datasets using two different tie weighting methods: 1) traditional number of co-authorships and 2) Newman's method, where for each collaboration, $1/(n-1)$ is contributed to the link weight, n being the number of authors present in the collaboration [39]. We name them as `dblp-1` and `dblp-2` respectively.

- `newman`, `astro`, `hep-th`, and `netscience`: `newman`, `astro` and `hep-th` [39] are datasets built from collaboration networks of Condensed matter, Astrophysics and High-energy theory of arXiv E-Print Archive between 1995 and 1999. `netscience` [98] is made from collaboration network of network scientists. `newman` collaboration network dataset is available with both the weighting schemes like `dblp`, and we name them as `newman-1` and `newman-2`, respectively. `astro`, `hep-th`, and `netscience` use Newman's method as the tie weighting measure.

2. Communication networks

- `enron`: This dataset is constructed from the e-mails exchanged between employees of an organization available at `http://www.cs.cmu.edu/~enron/`. From each e-mail, the fields *From*, *To*, *Cc* and *Bcc* are extracted to form edges between the sender (mail id present in the field From) and receiver(s) (mail id(s) present in the fields To, Cc and Bcc) of that e-mail. Directions are ignored to keep the graph undirected. The number of emails exchanged between two individuals represent the weight of the link. Downloaded data contains e-mails spanning January, 1997 – December, 2002. E-mails till November, 2001 is considered to build the graph, and rest are used for generating future links.
- `oclinks` :`oclinks` [99] is a communication network dataset collected from

a Facebook like social network. This network is built upon the messages exchanged among the users. Weight of a link indicates the number of messages exchanged between two users.

3. Other

   - `openflights`[3]: This is a network of airports. The number of routes between two airports gives the weight of the link between them.

The above datasets are further grouped into four subsets based on the nature of the network and the type of edge weighting method, which are referred as (i) `Col-1`: `dblp-1` and `newman-1`; (ii) `Col-2`: `dblp-2`, `newman-2`, `astro`, `hep-th` and `netscience`; (iii) `Com`: `enron` and `oclinks`; and (iv) `oth`: `openflights`.

3.4 Experimental observations

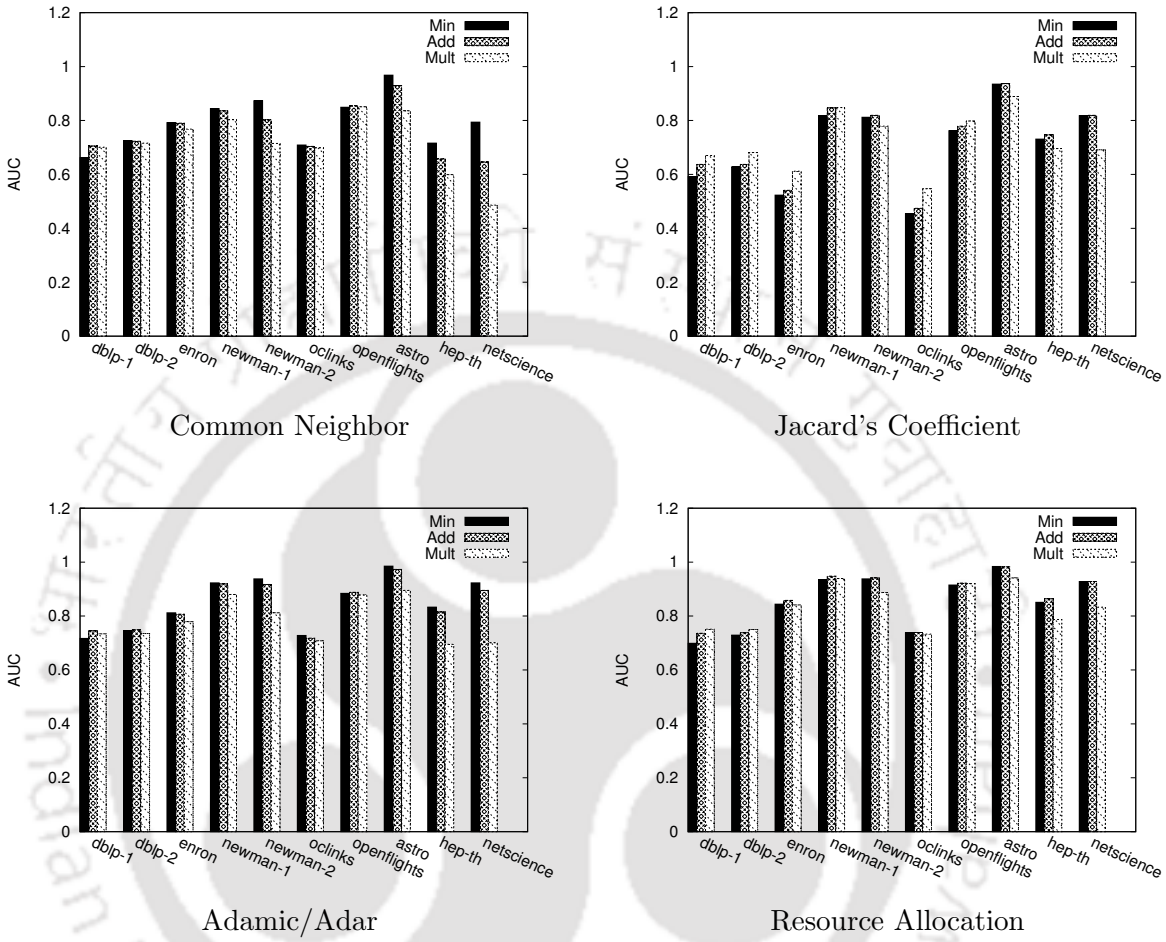
This section discusses the experimental responses of edge weights on four local proximity based link prediction methods namely JC, CN, AA and RA. Previous studies [40, 41, 86, 97, 100] have discussed responses of the additive model. This work introduces two other weighting models (as discussed in Section 2), and compare their response upon a larger collection of datasets with the response of the methods reported in [40, 41, 86].

3.4.1 Response of different influence models and networks

Figure 3.2 compares the AUC scores of three influence models namely min-flow, additive and multiplicative for CN, JC, AA and RA. There are 40 independent experimental cases (four local proximity based link prediction methods and 10 datasets), and following observations are evident.

- Characteristics of the relative response of the three models vary from one dataset to another. Out of the 40 experimental cases, min-flow responds better than its

[3]Opsahl, T., 2011. Why Anchorage is not (that) important: Binary ties and Sample selection



Common Neighbor

Jacard's Coefficient

Adamic/Adar

Resource Allocation

Figure 3.2: Response of influence models on prediction methods and datasets

counterparts in 42.5% of the cases, whereas additive and multiplicative models come first in 37.5% and 17.5% of the cases respectively (see Table 3.2).

- For the CN and AA, min-flow outperforms its counterparts in 90% and 70% cases respectively. For RA, additive responds better (in 60% of the cases), and for JC, there is a tie between multiplicative and additive (both 50% of the cases).

- From the perspective of edge weighting method, the following is observed: min-flow weighting model responds positively on Newman's edge weight (i.e., min-flow responded better in 50% of the cases, additive in 35% and multiplicative in 10% on

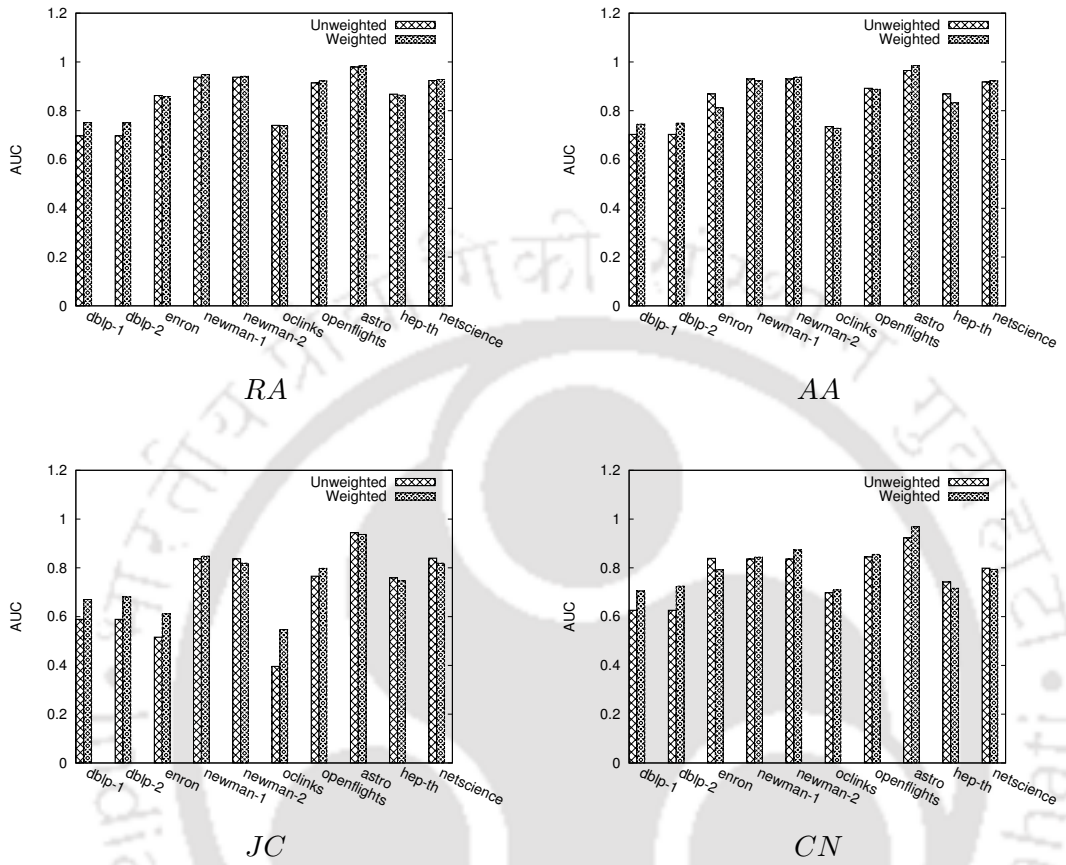


Figure 3.3: Comparing the AUC score of weighted and unweighted link prediction methods.

`Col-2` datasets), and additive performs best in case of frequency based edge weight. Of all the cases over `Col-1, Com` and `oth` datasets, additive performs best in 40% of the cases, min-flow in 35% of the cases and multiplicative in 25%.

Above observations clearly indicate that the performance of weighting models depends on the characteristics of the dataset and the prediction method used. Therefore, subject to the dataset and prediction method, weighting model should be carefully chosen.

Figure 3.3 further compares the performance of weighted and unweighted versions of the baseline prediction methods. Table 3.3 compares the performance of unweighted and weighted methods over datasets. For each prediction method and dataset pair, the best

Table 3.2: Detailed experimental cases, a comparative visualization.

	Min-flow (f)				Additive (a)				Multiplicative (m)			
	Col-1	Col-2	Com	oth	Col-1	Col-2	Com	oth	Col-1	Col-2	Com	oth
CN	$\frac{1}{2}$	$\frac{5}{5}$	$\frac{2}{2}$	$\frac{1}{1}$	$\frac{1}{2}$	$\frac{0}{5}$	$\frac{0}{2}$	$\frac{0}{1}$	$\frac{0}{2}$	$\frac{0}{5}$	$\frac{0}{2}$	$\frac{0}{1}$
	90%				10%				0%			
JC	$\frac{0}{2}$	$\frac{0}{5}$	$\frac{0}{2}$	$\frac{0}{1}$	$\frac{1}{2}$	$\frac{4}{5}$	$\frac{0}{2}$	$\frac{0}{2}$	$\frac{1}{2}$	$\frac{1}{5}$	$\frac{2}{2}$	$\frac{1}{1}$
	0%				50%				50%			
AA	$\frac{1}{2}$	$\frac{4}{5}$	$\frac{2}{2}$	$\frac{0}{1}$	$\frac{1}{2}$	$\frac{1}{5}$	$\frac{0}{2}$	$\frac{1}{1}$	$\frac{0}{2}$	$\frac{0}{5}$	$\frac{0}{2}$	$\frac{0}{1}$
	70%				30%				0%			
RA	$\frac{0}{2}$	$\frac{1,1:a}{5}$	$\frac{0}{2}$	$\frac{0}{1}$	$\frac{1}{2}$	$\frac{2,1:f}{5}$	$\frac{2}{2}$	$\frac{1}{1}$	$\frac{1}{2}$	$\frac{1}{5}$	$\frac{0}{2}$	$\frac{0}{1}$
	10%, 10% : a				60%, 10% : f				20%			
Overall	42.5%, 2.5% : a				37.5%, 2.5% : f				17.5%			

Note: For every fraction, the denominator gives the total number of experimental cases and the numerator denotes the number of occasions where the particular method outperforms others for the particular dataset type. $x, y : z$ in the numerator means that in x number of cases the particular method outperforms others, and in y number of cases a tie happens with weighting model z. % denotes the percentage of outperforming (or tied) cases instead of number.

responding weighted method and its unweighted counterpart are selected and plotted in Figure 3.3. Table 3.3 presents the percentage of the baseline prediction methods where weighted and unweighted methods dominate, for each dataset. Like in Figure 3.2, the responses from the weighted and unweighted methods are evenly distributed throughout the experimental cases.

Table 3.3: Influence of tie weight over different datasets. For oclinks, weighted and unweighted have equal influence.

	Weighted	Unweighted	Dominant
dblp-1	100%	0	Weighted
dblp-2	100%	0	Weighted
enron	25%	75%	Unweighted
newman	75%	25%	Weighted
oclinks	50%	50%	-
openflights	75%	25%	Weighted
astro	75%	25%	Weighted
hep-th	0	100%	Unweighted
netscience	25%	75%	Unweighted

Above observations clearly show that the response of edge weight depends on many factors, such as, (i) type of the network, (ii) influence model and (iii) edge weight formation. Therefore, it is important to study the influence of these factors on the prediction performance deeper. The remaining part of this chapter attempts to analyze the influence of each of these factors on link prediction. We also analyze the effect of tuning the weighting models. We investigate the effect of link weight over datasets of different nature next.

Effect of edge weight on collaboration network

As mentioned in Section 3, collaboration networks are divided into two groups namely `Col-1` and `Col-2`, based on the way edge weights are defined. Referring to the plots of `dblp-1` and `dblp-2` datasets in Figure 3.3, a clear case of favoring weighted methods is evident. For all the eight cases, weighted methods respond positively, and there is no

significant difference in performance between `dblp-1` and `dblp-2` over the methods.

Referring to the performance of `newman-1` and `newman-2` datasets in Figure 3.3, it is surprising to observe that there is no clear winner between unweighted and weighted methods. Though `Col-1` and `Col-2` are of similar nature (i.e., collaboration networks), their responses contradict in `newman` when weighted and unweighted methods are compared. It also indicates that tie weighing method also influences the link prediction performance.

Effect of edge weight on communication networks and `openflights`

Though the weighted methods outperform their unweighted counterpart for `openflights`, unlike collaboration network, unweighted methods tend to perform better than weighted methods in communication networks (refer Figure 3.3). The contradictory response of prediction methods in these datasets indicates the presence of highly centered nodes in the graph. In `enron` and `oclinks` datasets, few nodes have connectivity with a large fraction of the total nodes in the network. A large number of participating nodes are neighbors of these centered nodes, and the connecting edges to these centered nodes have high edge weights (see Section 3.5.1). Because of these edges, scores of undeserving node pairs are expected to be boosted inappropriately, resulting in lower AUC scores.

Even after tuning on weighting parameters, discussed in Subsection 3.4.3, the performance do not improve. It indicates that these two tuning methods are not suitable for the datasets having highly centered nodes. Suitable study on tuning methods for the dataset having highly centered node is beyond the scope of this work. Instead we investigate density based enhancement (see Section 3.6), where performance of the weighted methods is significantly enhanced.

3.4.2 Effect of edge weights on Prediction Methods

This subsection investigates the influence of edge weights from different perspectives by introducing edges in the network in incremental fashion. In this experiment, the graph is initially assumed to be empty. Then the edges are introduced in increasing order of their

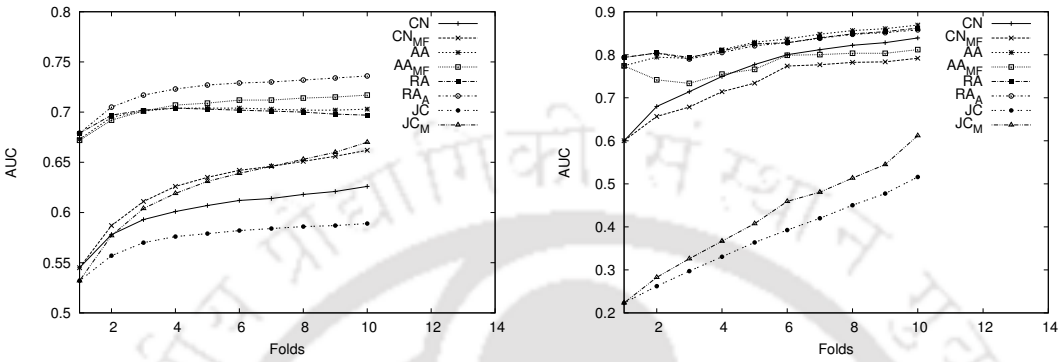


Figure 3.4: Effect on different prediction methods after introducing edges in increasing order of their weights.

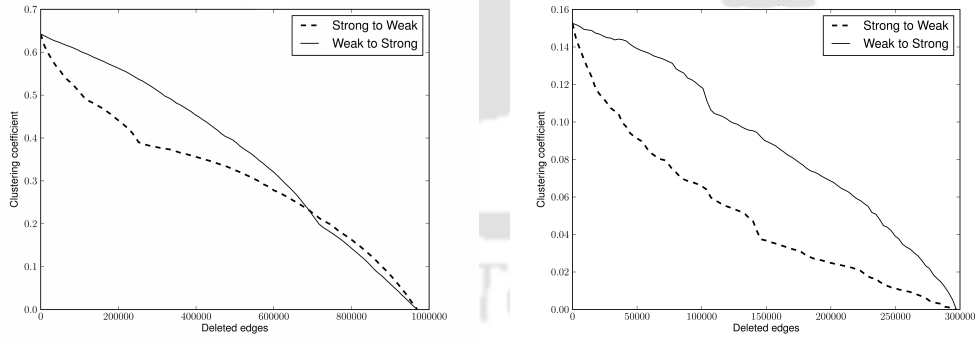


Figure 3.5: Decrease of clustering coefficient by deleting links in increasing and decreasing order of their weights

edge weights. The edges are divided into 10 folds in increasing order of their weights, and edges belonging to each fold are added to the graph incrementally. After adding edges of each fold, link prediction methods are applied to the resulting graph. This experiment helps us to understand influence of the edge weights over the link prediction methods. The plots in the Figure 3.4 shows an interesting characteristic:

> Prediction performance gradually increases with the increase in the tie weights for both JC and CN. However, plots quickly stabilize in the case of AA and RA. It indicates that AA and RA are more resistive towards the tie weights as compared to JC and CN. Hence, one can expect a larger effect of the link weight on JC and CN as compared to AA and RA, which is presented in Figure 3.3 as well.

In Figure 3.5, we further analyze the change in the average clustering coefficient of two graphs while deleting the edges in increasing and decreasing order of weights. The two graphs have a significant difference in characteristics: (i) the area between the two curves is small in `dblp-1` and large in `enron`, and more interestingly, (ii) the two plots intersect in case of `dblp-1` and does not intersect in the case of `enron`. The rate at which the plot decreases, indicates the change in the local density surrounding the nodes. While deleting edges starting with stronger to weaker, `dblp-1` gradually decreases its clustering coefficient, while for `enron`, it exponentially decreases. It indicates that the cohesiveness of `enron` is bounded by few central nodes with highly weighted edges, whereas, it is distributed uniformly in case of `dblp-1`. These observations provide an indication that the response of tie weight on prediction methods depends on the characteristics of the underlying graph. A detailed analysis on the clustering coefficient is presented in Section 3.5.1.

3.4.3 Tuning the weighted models

This section investigates if the performance of the prediction methods can be improved by tuning the weighted factor. It proposes two tuning methods namely *scaling* and *linear sum*. These tuning methods are applied on RA. We have selected RA for this analysis, because RA consistently performs better than other prediction methods as seen from Figure 3.6.

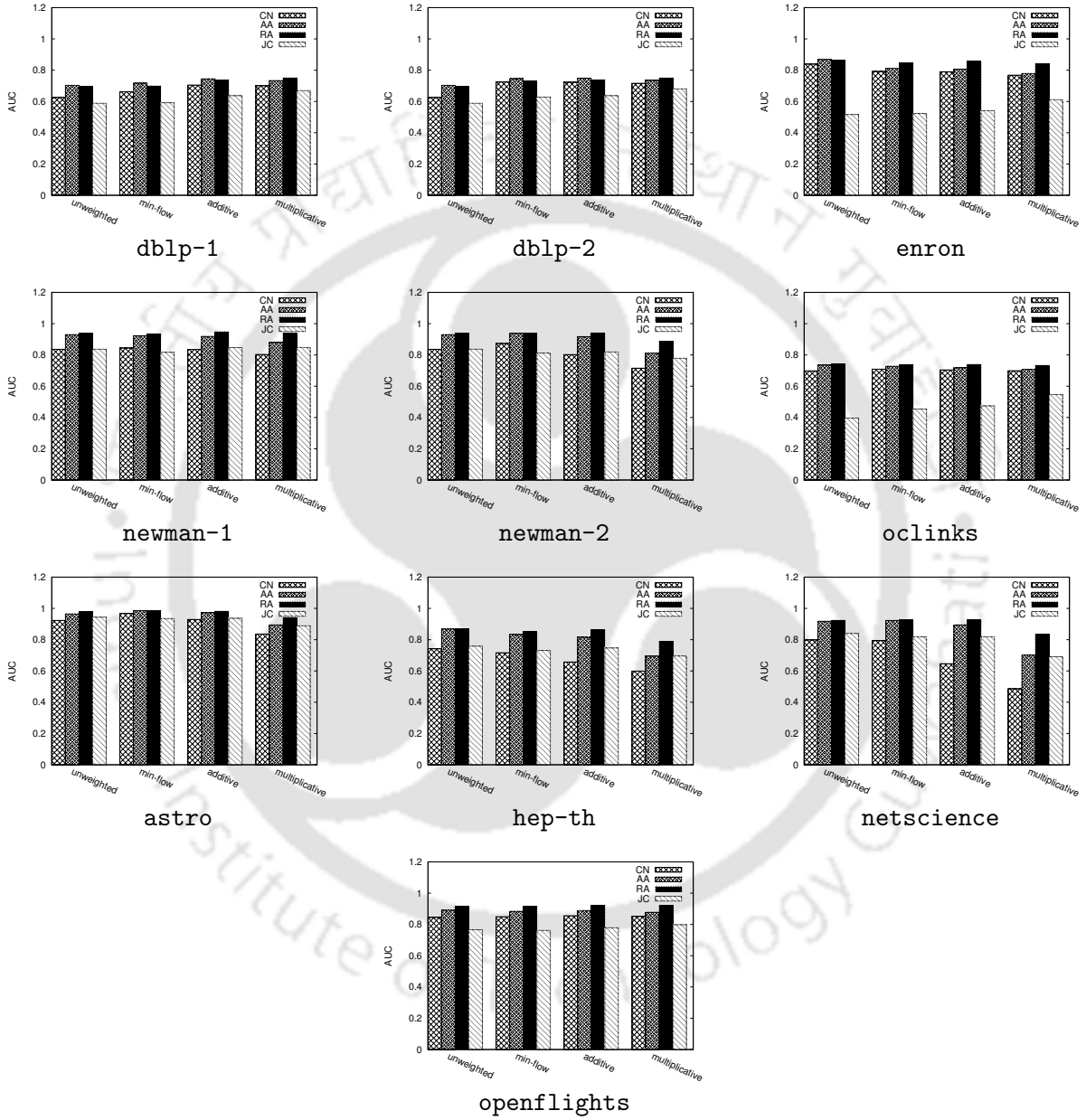


Figure 3.6: Performance of prediction methods over datasets and weighting methods.

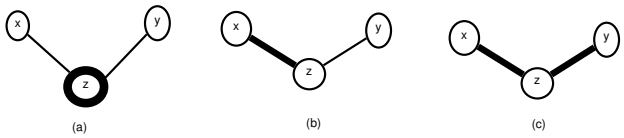


Figure 3.7: Degenerate case of additive models of RA

Out of the 40 cases, *RA* performs better than others in 29 cases, and *RA* and *AA* jointly outperforms in 9 more cases. Further, it is observed in Table 3.2 that *RA* performs best with the additive model. So, this subsection focuses on RA_A as the weighted method for tuning.

Scaling

RA_A suffers from a degenerate situation when common neighbors do not have any links other than the ones they share with the target nodes. Three possible connection settings depicting the situation are shown in Figure 3.7, where x and y are the target nodes, and z is one of their common neighbor. The thicker line represents a higher weight. Additive model based weighted *RA* returns same score for all these cases because, there is no neighbor of z other than x and y. However, their response should be different considering *triadic closure property* of the weak tie theory [19]: *if edges* (x, z) *and* (y, z) *are connected by strong ties, the chances of connecting* x *and* y *by at least a weak tie is high.* So, the prediction score should be biased by the tie weight of the edges connecting the common neighbors. To mitigate this degenerate situation, we introduce a tuning factor in RA_A as follows.

$$RRA_s(x, y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{(w(x, z) + w(z, y)) \times \Omega(x, y, z)}{s(z)}$$

where $\Omega(x, y, z)$ is the tuning factor. In this study, the tuning factor is defined as $\Omega(x, y, z) = \log(1 + w(x, z) + w(z, y))$. It cooperates with the triadic closure property.

Table 3.4: Effect of Regularized RA.

	RA	RA_A	RRA_S	RRA_l	Dominant
dblp-1	0.697	0.736	0.747	0.736	RRA_s
dblp-2	0.697	0.738	0.747	0.738	RRA_s
enron	0.862	0.858	0.853	0.682	RA
newman-1	0.938	0.948	0.944	0.958	RRA_l
newman-2	0.938	0.941	0.934	0.952	RRA_l
oclinks	0.740	0.739	0.737	0.683	RA
openflights	0.914	0.922	0.921	0.921	RA_A
astro	0.979	0.982	0.939	0.985	RRA_l
hep-th	0.868	0.864	0.806	0.876	RRA_l
netscience	0.924	0.928	0.844	0.937	RRA_l

Linear Sum

In each of CN, AA and RA, importance of $z \in \Gamma(x) \cap \Gamma(y)$ alone is considered. The relative importance of the participating nodes is entirely ignored. However, participating nodes, which are having relatively higher amount of interaction with common neighbors compared to their other neighbors, are expected to reflect stronger bonding. RA is regularized using linear summation as follows.

$$RRA_l(x,y) = \alpha.\Omega(x,y) + (1-\alpha)RA_A(x,y)$$

where $0 \leq \alpha \leq 1$ and $\Omega(x,y)$ is the regularization factor. In this study, we use $\frac{1}{2} \times \sum_{z \in \Gamma(x) \cap \Gamma(y)} (\frac{w(x,z)}{s(x)} + \frac{w(y,z)}{s(y)})$ as $\Omega(x,y)$.

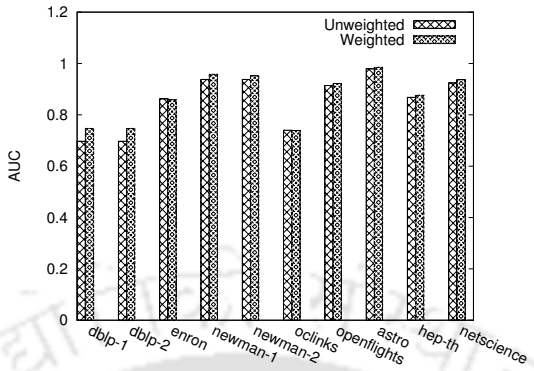


Figure 3.8: Comparison between AUC scores of unweighted RA and best weighted measure among all variants of weighted RA.

Effect of weight after tuning

Table 3.4 compares the performance of weighted RA after tuning with its counterparts: unweighted RA and RA_A. It clearly shows that the tuning methods influence the datasets diversely. Regularization using linear weighted sum (denoted by RRA_l) enhances the performance significantly over all collaboration networks for both the frequency based and the Newman's edge weighting measure. However, scaling enhances only for the networks whose edges are weighted using frequency. Surprisingly, there is no influence of the above tuning methods on `openflights` datasets, and there is negative influence of the tuning methods on communication networks. Figure 3.8 compares the performance of unweighted RA and best one among all of the weighted versions of RA. It shows that, a right choice of weighting and tuning methods, edge weights can indeed enhance the link prediction performance over the collaborative networks. For other datasets, further investigation is needed on tuning methods, which is beyond the scope of this analysis.

3.5 Node proximity based analysis of `dblp` and `enron`

From the empirical study presented in the previous sections, it is observed that the effect of incorporating weight to the common neighbor based link prediction methods is not consistent over networks. In previous sections, analysis has been limited to applying

different weighting models to local proximity based link prediction methods. Further, with suitable weight tuning methods, it is also observed that prediction methods can use the weight effectively. Though, with the tuning methods introduced in Subsection 3.4.3, the prediction performance of all the collaboration networks improves with the weight factor, for communication networks, the response is observed otherwise. Such observations raise two questions : 1) which property of the network governs the effect of link weight on the common neighbor based link prediction methods, and 2) given a network graph, can we effectively decide beforehand whether to incorporate weight or not? This section addresses these questions.

One such study based on network characteristic has been presented in [41]. This paper has reported that incorporating link weights with the local proximity indices worsen the performance of link prediction. They have added an exponent to the link weights while calculating the score, and have discovered that for most of the cases, the prediction performance improves for negative values of the exponent. They have concluded that their result has been supported by Granovetter's *weak-tie theory* [19] and have done a subgraph analysis on the network graphs. Though they have claimed it to be *Motif analysis*, there is no clear justification of considering every possible 3-node subgraph as Motif [5].

Formation of new links in a social network is governed by the *triadic closure property* [19] (also referred as triangle closing), which states that, if two nodes have a common friend, then they are likely to be friends. Its weighted counterpart, the *strong triadic closure property* says that probability of two nodes being friends increases with the weight of the links that connect them with a common friend [19, 101]. According to the subgraph analysis presented in [41], violation of the *strong triadic closure property* is clear. Their conclusion suggests that weak ties play major role in triangle closing. In this work, clustering co-efficient of a node, which also relies on the *triadic closure property*, has been analyzed to investigate the effect of weights on the link prediction measures. `dblp-1` and `enron` graphs are chosen for the analysis as the former one is a collaboration network, and the later is a communication network. These networks are considerably larger compared to others, and time-stamp information is also available for both, that allows us to test the

performance of the prediction methods with true future links.

3.5.1 Analyzing Clustering Co-efficient (CC)

Clustering co-efficient (CC) [102] of a node x in a network can be given by $2 \times |t(x)|/\Gamma(x)(\Gamma(x)-1)$, where $t(x)$ is the set of triangles formed by the node x and its neighbors. It reflects the density in the neighborhood of a particular node. High CC of a node indicates that the triadic closure property holds strong in its neighborhood. Average CC of a network, denoted as C, is defined as the average of clustering coefficients of all nodes of that network.

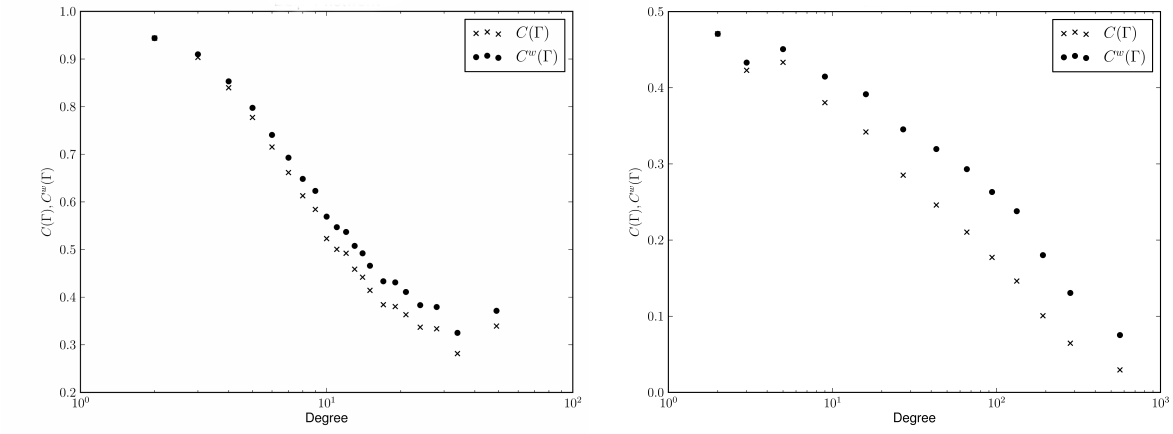
Barrat et al. [103] have proposed a version of weighted CC that can be defined as (for undirected graph):

$$CC^w(x) = \frac{1}{s(x)(\Gamma(x)-1)} \sum_{\{x,y,z\} \in t(x)} (w(x,y) + w(x,z))$$

This measure not only quantifies the local cohesiveness in terms of triangle closure, but also takes the link weights into account to calculate the likelihood of forming triangles by a node with its neighborhood. The average weighted clustering co-efficient of a network C^w can be calculated in the same way as that of its unweighted version. The value of both C and C^w range between 0 to 1.

By the argument presented in [103] and [104], if $C^w > C$ holds in a particular network, it indicates that the triangle closing is governed by the stronger links, and the weaker ones dominate in case of $C^w < C$. As per our result of link prediction on `dblp-1` and `enron` graphs, this argument directs us to think that the first case holds in `dblp-1`, and `enron` falls in the second case. However, surprisingly, both the networks show the same characteristic: $C^w > C$. In both the cases, the higher weights play more significant role than the lower weights in the phenomenon of triangle closing, and both of the datasets support *strong triadic closure property.*

To further analyze these two graphs in terms of CC, inspired by [103], we compare $C(\Gamma)$ and $C^w(\Gamma)$, where $C(\Gamma)$ is the average CC of the nodes having degree Γ, and $C^w(\Gamma)$ is its weighted counterpart. Figure 3.9 shows the comparison of the spectrum of $C(\Gamma)$ and



dblp-1 enron

Figure 3.9: Spectrum of $C(\Gamma)$ and $C^w(\Gamma)$ of dblp-1 and enron.

$C^w(\Gamma)$ for the two networks. For both networks, $C(\Gamma)$ and $C^w(\Gamma)$ decay as Γ increases. It supports the fact that the hub nodes with high degree, have lower CC as they work as bridges between different groups. However, the ratio between $C^w(\Gamma)$ and $C(\Gamma)$ is much higher in enron network than dblp-1, particularly for hub nodes. This is due to the fact that the concept of hub nodes in dblp-1 and enron networks are different. In dblp-1, the researchers, who work in interdisciplinary research areas and have a large amount of contacts spread throughout different geographical areas, act as hubs. The hub nodes may represent experienced and prominent researchers. Furthermore, publishing a research paper is a slow process. Hence, the average weight of the edges connecting the hub nodes do not grow fast. On the contrary, as enron is an e-mail network among the employees of an organization, the managers of the organization represent the hub nodes, and they communicate with other employees very frequently. This keeps the ratio of $C^w(\Gamma)$ and $C(\Gamma)$ very high for enron network. To summarize, very high amount of weight surrounds the hub nodes in enron network. The average degree of the hub nodes in enron network is also very high compared to dblp-1. This may cause the conflicting effect of edge weights in the link prediction performance on these datasets.

3.5.2 The proposed measure

From the findings of previous subsection, it is evident that CC is not always able to answer the second question raised at the beginning of Section 3.5. CC measures local cohesiveness of a particular node. However, all local link prediction methods exploit the number of common neighbors the target nodes share. Moreover, unlike average CC, which covers all the nodes of the network, the prediction methods apply only on the node pairs that have at least one common neighbor. This fact motivates us to propose the following measure.

Let $G = (V, E)$ be the undirected graph representing the network, where V is the set of vertices, and E is the set of edges. Let E' denote the set of edges each of which has at least one common neighbor, i.e., for each edge $\{x, y\}$ in E', $|\Gamma(x) \cap \Gamma(y)| > 0$. The average of the edge weights associated with the common neighbors of the end nodes of the edge $\{x, y\}$ in E' is given by:

$$\hat{C}_{\{x,y\}} = \frac{1}{K_{\{x,y\}}} \sum_{z \in \Gamma(x) \cap \Gamma(y)} (w(x, z) + w(z, y)),$$

where $K_{\{x,y\}} = 2|\Gamma(x) \cap \Gamma(y)|$. Note that, $\hat{C}_{\{x,y\}}$ is a normalized version of $CN_A(x, y)$. Then we define *group average* index by averaging $\hat{C}_{\{x,y\}}$ for all $\{x, y\}$s in E':

$$\hat{C}^{ga} = |E'|^{-1} \sum_{\{x,y\} \in E'} \hat{C}_{\{x,y\}}.$$

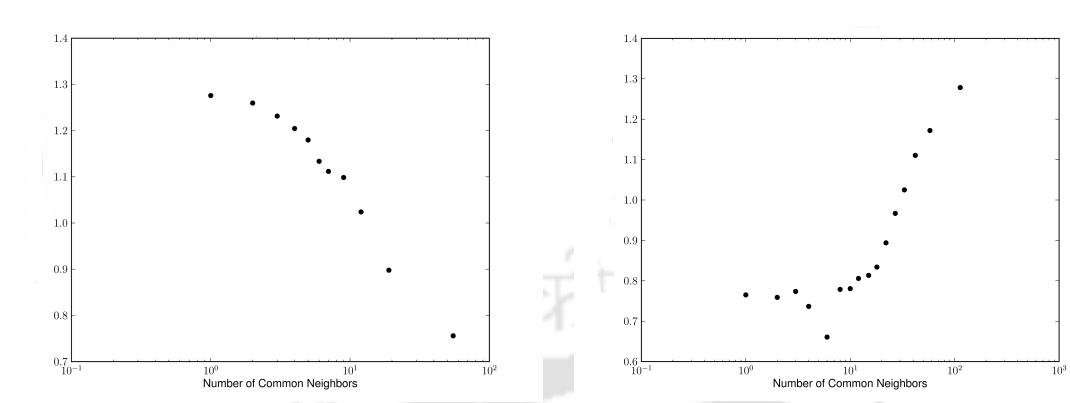
Another global index, *flat average* is defined as:

$$\hat{C}^{fa} = \frac{1}{K} \sum_{\{x,y\} \in E'} \sum_{z \in \Gamma(x) \cap \Gamma(y)} (w(x, z) + w(z, y)),$$

where $K = \sum_{\{x,y\} \in E'} K_{\{x,y\}}$.

A very interesting observation follows when $\hat{C}^{ga}$ and $\hat{C}^{fa}$ are compared in `dblp-1` and `enron` networks. $\hat{C}^{ga} > \hat{C}^{fa}$ holds for `dblp-1` and $\hat{C}^{ga} < \hat{C}^{fa}$ holds for `enron`. This finding inspires us to inspect the relation between C^{ga} and C^{fa} further from the perspective of underlying network property. The difference between $\hat{C}^{ga}$ and $\hat{C}^{fa}$ can be given by,

$$\hat{C}^{ga} - \hat{C}^{fa} = \sum_{\{x,y\} \in E'} \Big(\frac{1}{|E'|} - \frac{K_{\{x,y\}}}{K} \Big) \times \hat{C}_{\{x,y\}}$$



dblp-1 enron

Note: Label of vertical axes represents $\frac{Average\ C_{\{x,y\}}\ grouped\ by\ common\ neighbor\ count}{\hat{C}^{fa}}$.

Figure 3.10: Plots showing the average $C_{\{x,y\}}$, grouped by the number of common neighbors for dblp-1 and enron.

As all the link weights considered here are of positive values, a particular node pair $\{x, y\}$ contributes some positive value to the summation if $K_{\{x,y\}} < \frac{K}{|E'|}$ holds for that particular pair. Therefore, if higher weights are concentrated in the edges of the common neighbors connecting the node pairs $\{x, y\}$, which have lesser number of common neighbors, $\hat{C}^{ga} - \hat{C}^{fa}$ will give a positive value, and vice versa. Figure 3.10 shows the distribution of $C_{\{x,y\}}$, averaged over the number of common neighbors and normalized by $\hat{C}^{fa}$, for the two networks. For dblp-1, it clearly shows that the averaged $C_{\{x,y\}}$ is high for the node pairs, whose number of common neighbors are small, and low when the number of common neighbors are large. enron shows exactly the opposite characteristics.

From the analysis of $\hat{C}^{ga}$ and $\hat{C}^{fa}$, we can loosely infer that if $\hat{C}^{ga} > \hat{C}^{fa}$ holds for a particular network, the weighted methods are likely to perform better than its unweighted counterpart and vice versa. Observation from Figure 3.10 also supports the observation on the hub nodes presented in the previous subsection. For both the networks, it is observed that the number of common neighbors of two nodes increases with the average degree of the two nodes. So, in enron network, when two manager nodes are linked and they have large number of common neighbors, as large weights are concentrated in their links, their

$C_{\{x,y\}}$ gives higher value.

3.6 Effect of localized weight distribution

One of the drawbacks of local node proximity based link prediction methods is its dependency on local density. In the previous section, we have observed that for both `dblp-1` and `enron` graphs, triangle closing is governed by stronger links. In [19], the author have said that close knit community is built by stronger ties and weak ties form *bridges* between communities. Motivated by these observations, we further investigate the effect of weighted and unweighted methods by distributing the target node pairs at different odd ratio. We define the odd ratio between unweighted and weighted as follows.

$$Odd = \frac{unweighted\ score}{weighted\ score}$$

We partition the test edges into two disjoint sets: *low* and *high*. We put an edge into `low` set if the average odd ratio between the two participating nodes is less than a threshold value. In this study, we sort the edges by their average odd ratio of the participating nodes and divide the edge set into two at different split points: 25-75%, 50-50% and 75-25%. For a 25-75% split, top 25% fall into `high` set and rest 75% in `low`. As mentioned earlier, RA_A performs best in most of the datasets, and therefore we restrict our odd-ratio study only on RA_A.

In this section, we focus on the odd ratio of unweighted and weighted degree. The value of odd ratio of a node reflects the relative distribution of weights among its neighbors. Very high value of odd ratio means that the neighboring nodes are likely to be connected by weak ties, and very low values of odd ratio means that the neighboring nodes are likely to be connected by strong ties. If both the participating nodes have very low odd ratio value, the chances that both the participating nodes belonging to the same dense region or cluster is high. It is expected to achieve higher prediction accuracy compared to its counterparts in the high odd-ratio region. Table 3.5 clearly shows that the estimates in the low region outperforms the estimates in the high regions. Splitting the dataset into the high and low regions is like setting an experimental constraint to remove noisy links.

Table 3.5: AUC comparison between variants of RA for test edges in low and high region using degree odd ratio with split $50-50\%$.

	Variant	dblp-1	dblp-2	enron
High	RA	0.711(2.0)	0.697(0.1)	0.832(-3.4)
	RA_A	0.710(-3.5)	0.705(-4.5)	0.818(-4.6)
	RA_M	0.693(-7.7)	0.674(-10.5)	0.772(-8.2)
Low	RA	0.682(-2.2)	0.695(-0.1)	0.890(3.3)
	RA_A	0.760(3.2)	0.771(4.5)	0.899(4.7)
	RA_M	0.807(7.4)	0.833(10.6)	0.910(8.2)

Note: An entry within parenthesis shows percentage of increase or decrease.

With 50-50% splits, we are able to achieve an improvement of AUC up to 10% on the weighted measures and up to 2% on the unweighted measures.

Assuming that the network satisfies Granovetter's theory on tie strength, it is expected that the unweighted estimates in the high region decrease their performances, and the weighted estimates in the low region increase their performances as we increase the range split (say from 25-75% to 50-50% and then 75-25%). With small split range, the chances of nodes falling into different clusters in low region is high, and hence low region is expected to have noisy edges resulting in lower performance of weighted estimates. As we increase the split range, the amount of noisy edges gets decreased, resulting in the increase in the prediction performance of weighted estimates. It is also true for unweighted estimates that the high region at the large split range has higher number of noisy edges as compared to small split range. Hence, it results in decreasing prediction performance of unweighted estimates in the high region as we increase the split range. Figure 3.11 clearly shows that the above expected output is satisfied for `dblp` dataset. In the case of `enron`, with the increase in split range, the performances of all estimators are also increased. However, the rate (slope of the plot) at which unweighted estimate increases

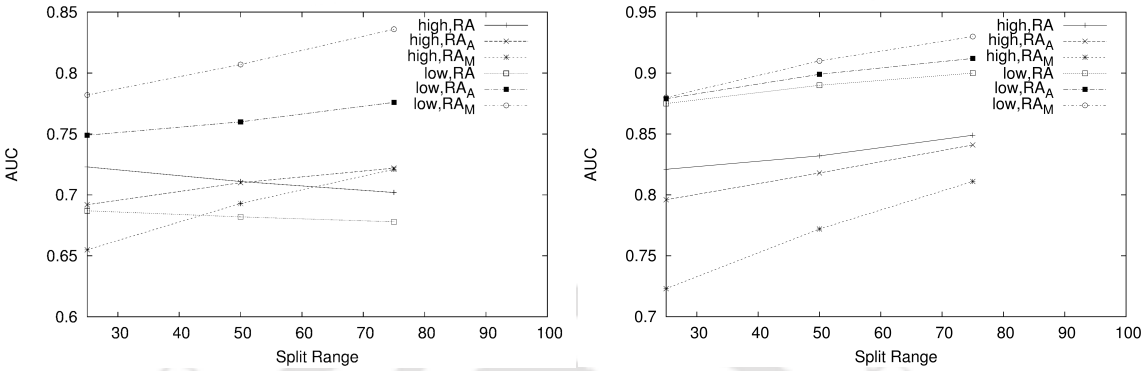


Figure 3.11: Performance at different split range.

its performance is smaller than its weighted counterpart.

It can also be noted from the Figure 3.11 that the weighted estimates of the low region in all split ranges outperform its unweighted counterparts including `enron` dataset. Therefore, with appropriate weighting model and appropriate tuning factors, weighted prediction method can perform better than its unweighted counterpart.

3.7 Summary

This study presented an empirical analysis on the effect of the tie weight on four node proximity based link prediction methods : CN, AA, RA and JC. Two weighting models namely, min-flow and multiplicative were introduced in this study, both of which outperform the traditional additive model in a significant number of dataset-proximity measure pairs. The empirical results have shown a diverse effect of the tie weight on datasets and proximity measures. In few datasets, which are negatively affected by the tie weight, the weighted methods have been observed to perform positively when regularization is applied. However, the proposed regularization methods are not effective in few other datasets (communication datasets).

To understand the reason behind the conflicting responses of tie weight over different

datasets, this study has further exploited the characteristics of the underlying social network graphs. From this analysis, it is observed that the number of hub nodes in the network and their weighted strength influence the performance of the weighted node proximity based link prediction methods on that network. Further, the degree odd-ratio analysis over the datasets has shown that unweighted models are effective for inter-cluster link prediction, and weighted are effective for intra-cluster link prediction.

Poor performance of the weighted prediction methods in communication networks gives us motivation to search for appropriate weighting mechanism and prediction methods for such networks, which we do in the next chapter.

Chapter 4

Exploiting reciprocity towards link prediction

4.1 Overview

Common-neighbor based *link prediction* methods like *common neighbor (CN), Jaccard's coefficient (JC), Adamic/Adar (AA), resource allocation (RA), etc.* [12, 38] are defined over undirected networks. However, many of the real networks, such as e-mail network, Twitter[1] network, etc., are directed in nature. While applying these methods on directed networks, existing studies [41, 105] usually convert them into undirected ones by ignoring directions of the links[2]. In the previous chapter, we followed the same conversion rule. This rule treats unidirectional (or one-way) and reciprocal (or both-way) links in the same way. Reciprocal links represent reciprocative feelings between the actors, and flow of information happens both-way. So, reciprocal links are considered more durable and stronger than

[1] `https://twitter.com/`

[2] Let a directed link from node x to node y be represented by a ordered pair (x, y). There are three possible connection set-ups between x and y: $\{(x, y)\}$, $\{(y, x)\}$, and $\{(x, y), (y, x)\}$. If any of the three set-ups exists between x and y in the directed network, x and y are connected by an undirected link in the resultant undirected network.

unidirectional ones. Following the *strong triadic closure property*[3] of social networks, it leads to an intuition that, reciprocal links may take lead role in *triad* formation. As triad formation is the building block of common-neighbor based link prediction methods, this work finds empirical evidence supporting the intuition, and exploits it to enhance the link prediction performance. Although there exist studies [106–109] which analyze the properties of directed edges and reciprocal links in dynamic networks, to the best of our knowledge, there exist no such study that exploits reciprocal links in the process of triad formation and common neighbor based link prediction methods.

In this chapter, we provide empirical evidence supporting "existence of high number (maximum 3) of reciprocal links in triads" in real networks using a null model. This evidence is exploited in three ways.

(1) A simple reciprocity-aware link weighting mechanism is proposed, and the weighted versions [40, 41, 110] of the link prediction methods are applied on the resulting weighted network. We revisit the debate concerning "effect of link weight in link prediction methods", introduced in the previous chapter. The previous chapter has concluded that link weight has positive influence on prediction methods in undirected networks (co-authorship networks), and negative influence in directed networks (communication networks), when the link weights are quantified by the traditional *frequency of interaction* link weighting method. The traditional frequency of interaction link weighting mechanism serves as the baseline link weighting mechanism to model link-strength in a static representation of network. However, the empirical analysis presented in Chapter 3 reflects that frequency of interaction may not be effective in all networks, as far as weighted link prediction is concerned. Here we exploit the empirical evidence supporting influence of reciprocal links in triangle closing, and propose a simple link weighting mechanism for directed networks. Experimental results show that prediction performance (using state-of-the-art weighted link prediction methods) is enhanced when the proposed reciprocity-aware link-weight is considered while estimating link weights, which is observed otherwise

[3]It says that, given three nodes x,y and z in a network, if there exist strong links between $x - z$ and $y - z$, then there exists at least a weak tie between x and y.

when frequency of interaction forms the link-weights.

(2) This work proposes three new common neighbor based (unweighted) link prediction methods for directed networks, which exploit directions of the edges.

(3) The proposed measures are considered as features, and several models combining them are prepared towards supervised prediction.

Experiments are performed over two real directed networks to demonstrate that proposed unsupervised and supervised methods enhance link prediction performance.

4.2 Datasets

Experiments are carried out on two directed network datasets of different characteristics: Facebook wall-post network [111] (`fb-d`), and Enron e-mail network[4] (`enron-d`). The Facebook wall-post dataset consists of all the wall-posts (along with their time-stamps), posted in Facebook[5] New Orleans network spanning the period September 26th, 2006 and January 22nd, 2009. A directed graph from Facebook wall-post dataset is built as follows: if a user x posts some message to another user y's facebook wall, then a directed edge from x to y (denoted by ordered pair (x, y)) is added. `enron-d` is the directed version of the `enron` dataset, which has been used for experiments in Chapter 3. It is constructed as: if x sends an e-mail to y, then a directed edge from x to y is added. We applied a bit preprocessing on these two datasets. Enron dataset contains some dummy nodes, which do not represent any real e-mail id (such as daddy@enron.com). So, we removed the nodes which have zero out-degree.

Reciprocity index (ρ)[6] [112] of `fb-d` and `enron-d` networks are 0.452 and 0.101 respectively. Reason for this notable difference in the values of ρ in the two networks is: `fb-d` represents a social network where ties tend to be reciprocal, whereas `enron-d` being an e-mail network acts more like information network where information flow is predominantly one-way.

[4] http://www.cs.cmu.edu/ enron/

[5] `https://www.facebook.com/`

[6] Reciprocity of a directed network is given as the fraction of bidirectional links in the network.

The traditional *frequency of interaction* based link weighting method assigns weights to the directed edges of the two networks as follows: in `fb-d`, if x posts n number of times in y's Facebook wall, then n is assigned as the weight of link (x, y); in `enron-d`, if x sends n number of e-mails to y, then n is assigned as the weight of link (x, y). These directed weighted graphs are converted to undirected weighted ones by ignoring the directions as follows. For a reciprocal links, sum of weights of the two directed edges is assigned as the weight of its undirected counterpart. Weight for a one-way links is kept same in its undirected counterpart. The proposed reciprocity-aware link weighting scheme is discussed in later part of this chapter. `enron-d` and `fb-d` are weighted using reciprocity aware weight as well as the traditional frequency of interaction weighting scheme, where frequency of interaction act as the baseline.

4.3 Empirical test

In this section, we demonstrate importance of reciprocal links in *triad formation*, which is the building block of the common neighbor based link prediction methods. We show using a null model that, in real directed networks, triangles or triads tend to be formed when high number (at most three possible) of reciprocal links reside in the triads, as compared to random graphs. A null model representing random graph is proposed and used for the test. The null model is designed in a way such that it preserves fundamental network properties[7] as that of the original graph. Randomization is done only on the direction of the edges keeping the total number of directed edges and the value of ρ (refer to Section 4.2) same as that of the original graph. Given a directed graph $G = (V, E)$ representing the dataset, where V and E represent the set of nodes and edges of G respectively, a random graph is generated as:

- Construct an undirected graph $G' = (V, E')$ from G by ignoring the directions of the edges in E. E' is the new set of edges satisfying the expression: $|E| - |E'| = \rho \times |E'|$.
- Assign random direction to the undirected edges in E' of G'. The resulting directed

[7]Such as *degree distribution, total number of triads, clustering coefficient*, etc.

graph with $|E'|$ number of one-way links is named $G'_{rand} = (V, E'_{rand})$.

- Randomly select $|E| - |E'|$ number of edges from E'_{rand} in G'_{rand}, and for each directed edge (x, y), add another edge (y, x) in the opposite direction to make the end nodes reciprocal. The resulting directed graph $G_{null} = (V, E_{rand})$ represents the null model.

The test comprises of comparing the number of triads with i number of reciprocal links in G, denoted as $N_G^{(i)}$, with the expected number of such triads $N_{null}^{(i)}$ in the null model. Possible values of i lies in $\{0, 1, 2, 3\}$. 1000 realizations of G_{null} are constructed, and $N_{null}^{(i)}$ is measured by averaging the number of triads with i number of reciprocal edges over those graphs. Deviation from the null hypothesis is measured by so-called *z-score*:

$$z^{(i)} = \frac{N_G^{(i)} - N_{null}^{(i)}}{\sigma^{(i)}}, \tag{4.1}$$

where $\sigma^{(i)}$ gives the standard deviation of the number of triads with i reciprocal links in the distribution over the null models.

Empirical test mentioned above is performed on the triads over two real directed networks: `enron-d` and `fb-d`. The results of the test are tabulated in Tables 4.1 and 4.2 respectively for `enron-d` and `fb-d`. The *z-score*, i.e., values in row named $z^{(i)}$ shows that, triads with higher values of i are highly overrepresented with respect to pure chance. On the other hand, significance of the triads lessens as the value of i decreases. The triads with lower values of i are highly underrepresented with respect to pure chance. Outcome of this test indicates that, presence of triads with high number of reciprocal edges is evident in real directed networks.

4.4 Proposed methods

Outcome of the test performed in the previous subsection motivates us to exploit the reciprocal links in link prediction. We consider unsupervised as well as supervised approaches towards it. In the unsupervised approach, there are two major contributions.

Table 4.1: Hypothesis test results for `enron-d`.

i	0	1	2	3
$N_G^{(i)}$	14102	93018	216203	66850
$N_{null}^{(i)}$	288296.302	192899.191	43048.563	1600.070
$\sigma^{(i)}$	886.006	1236.512	669.124	56.735
$z^{(i)}$	−309.472	−80.777	258.778	1150.083

Table 4.2: Hypothesis test results for `fb-d`.

i	0	1	2	3
$N_G^{(i)}$	352	6809	34343	24656
$N_{null}^{(i)}$	5134.884	25133.191	40946.166	11124.472
$\sigma^{(i)}$	80.963	156.160	203.775	154.599
$z^{(i)}$	−59.075	−117.342	−32.404	87.527

- We propose a reciprocity-aware link weighting mechanism. Then, traditional weighted link prediction methods are applied on the resultant weighted graph to achieve higher prediction accuracy than their unweighted counterparts.

- We propose three new common neighbor based link prediction methods (unweighted) by exploiting reciprocative nature of relationships.

Then we consider the individual methods (proposed and existing) as features, and build several models by combining them to perform supervised learning towards link prediction.

Table 4.3: Link prediction methods [12, 38, 40, 41, 110].

Method	Unweighted	Weighted
Common neighbor (CN)	$\lvert\Gamma(x) \cap \Gamma(y)\rvert$	$\sum_{z \in \Gamma(x) \cap \Gamma(y)} (w(x,z) + w(z,y))$
Jaccard's coefficient (JC)	$\frac{\lvert\Gamma(x) \cap \Gamma(y)\rvert}{\lvert\Gamma(x) \cup \Gamma(y)\rvert}$	$\frac{\sum_{z \in \Gamma(x) \cap \Gamma(y)} (w(x,z)+w(z,y))}{s(x)+s(y)-\sum_{z \in \Gamma(x) \cap \Gamma(y)} \min(w(x,z),w(z,y))}$
Adamic/Adar (AA)	$\sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{log\lvert\Gamma(z)\rvert}$	$\sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{w(x,z)+w(z,y)}{log(1+s(z))}$
Resource allocation (RA)	$\sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\lvert\Gamma(z)\rvert}$	$\sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{w(x,z)+w(z,y)}{s(z)}$

Note: $\Gamma(a)$ denotes the set of neighbors of node a; $w(a,b)$ denotes the weight of the link (a,b); $s(a) = \sum_{b \in \Gamma(a)} w(a,b)$ is the additive strength or weighted degree of node a.

4.4.1 Reciprocity-aware link weight and link prediction

In Chapter 3, it has been observed that weighted link prediction methods do not perform well in directed networks when frequency of interaction based link weights (and ignoring direction) are considered. In this subsection, we propose an alternative link weighting mechanism for directed networks, which exploits the reciprocative nature of social ties. The baseline unweighted and weighted link prediction methods are summarized in Table 4.3. The proposed link weighting method is described next.

The weighting mechanism

Link weight represents strength of social ties. Here we enumerate the strength of a social tie according to the reciprocative nature of the end nodes, i.e., one-way and reciprocal links are treated differently in this weighting mechanism. Reciprocal links signify both-way information flow between the actors, where the claim of association is bidirectional. On the other hand, in case of unidirectional links the claim of association is one-sided.

Hence, reciprocal links are usually considered stronger than unidirectional ones. We assign the reciprocity aware link-weights as follows.

- Assign value 2 as the weight of each reciprocal link.
- Assign value 1 as the weight of each one-way link.

This weighting mechanism allows the link prediction methods to honor the "importance of reciprocal links" in the triangle closing phenomenon, which is empirically shown in Section 4.3.

4.4.2 New link prediction methods

Here we propose three common neighbor based link prediction methods (unweighted) which take the direction of edges into consideration, and assign a similarity score to the target nodes. The proposed methods are defined next.

1. **Reciprocal flow (RF)**: RF index is formally defined as:

$$RF(x,y) = \begin{cases} 2, & \text{if both way flow between nodes } x \text{ and } y \text{ exists} \\ 1, & \text{if one way flow between nodes } x \text{ and } y \text{ exists} \\ 0, & \text{if no flow exists between nodes } x \text{ and } y \end{cases}$$

   If a flow exists between the target nodes x and y, the intermediate nodes that fall on the directed path from x to y, may relay information from x to y. If the flow is both-side, probability of x and y getting connected in future is higher.

2. **Reciprocal link count (RC)**: This index counts the number of reciprocal links that connects the target nodes to its common neighbor(s). Let $r(x,y)$ be a function that returns 1 if nodes x and y are connected by bidirectional edges, and 0 otherwise. The RC index is formally defined as:

$$RC(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} (r(x,z) + r(y,z))$$

3. **Normalized reciprocal link count (NRC)**: Normalized reciprocal link count is a normalized form of the RC index, which divides the RC index by the number of links that connects the target nodes to its common neighbor(s). It is formally defined as:

$$NRC(x,y) = \frac{\sum_{z \in \Gamma(x) \cap \Gamma(y)} (r(x,z) + r(y,z))}{2 \times |\Gamma(x) \cap \Gamma(y)|}$$

It represents the density of reciprocal links in all two-length paths between x and y.

4.4.3 Supervised prediction

The proposed link prediction measures are used as features in our supervised prediction models. We categorize the features that are used to build various supervised models in four:

1. U: the **u**nweighted prediction methods presented in Table 4.3 plus *in-degree, out-degree, preferential attachment, prop-flow*,[8]
2. T: the weighted counterparts of the features in U, when the **t**raditional link weighting mechanism is applied,
3. R: the weighted counterparts of the features in U, when the **r**eciprocity-aware link weighting mechanism is applied,
4. P: methods **p**roposed in Subsection 4.4.2.

We use the *High performance link prediction (HPLP)* supervised framework (discussed in Appendix A) proposed by Lichtenwalter [105] in our experiments, because it is robust against class imbalance.

4.5 Experimental results

Evaluation of the link prediction methods is a bit tricky due to its longitudinal characteristics. For evaluating unsupervised methods, the whole dataset is divided into

[8]Last four features are included to reduce over-fitting due to very less number of variables. These features have been used by Lichtenwalter [105] too.

Table 4.4: Significance of the comparative performance of reciprocity-aware weight in link prediction over unweighted and traditional weighting.

Dataset	Comparison	$p-value$	*mean* difference
`enron-d`	Unweighted vs. Weighted (T)	0.1647	0.014
	Unweighted vs. Weighted (R)	0.3828	−0.001
	Weighted (T) vs. Weighted (R)	0.138	−0.015
`fb-d`	Unweighted vs. Weighted (T)	0.045	0.015
	Unweighted vs. Weighted (R)	0.598	−0.004
	Weighted (T) vs. Weighted (R)	0.051	−0.020
`combined`	Unweighted vs. Weighted (T)	0.005	0.015
	Unweighted vs. Weighted (R)	0.476	−0.003
	Weighted (T) vs. Weighted (R)	0.005	−0.018

Paired t-test is performed on the AUC values to calculate the significance. The results of `combined` is achieved by merging the sample sets for `enron-d` and `fb-d`.

two contiguous (with respect to time) parts. The first part is used to prepare the graph, and the second one is used to collect the future links among the nodes present in the graph. These future links serve as the target class examples [12]. In case of supervised prediction, due to temporal dependency, the moment when the model is tested, must follow the moment when the training is complete. We follow the procedure same as HPLP [105] to collect examples for training and testing. Detailed discussion on the method used to divide the training and test interval is presented in Appendix A. We fix the values of g and d (Fig. A.1) for the two graphs as: 31 months and 4 months respectively in `enron-d`, and 31 months and 3 months respectively in `fb-d`. We use *Area under the receiver operating curve (AUC)* as the evaluation metric.

4.5.1 Reciprocity-aware link weight

Links of the two directed networks `enron-d` and `fb-d` are assigned reciprocity-aware weights as mentioned in the Subsection 4.4.1. Then the effect of this link weight in link

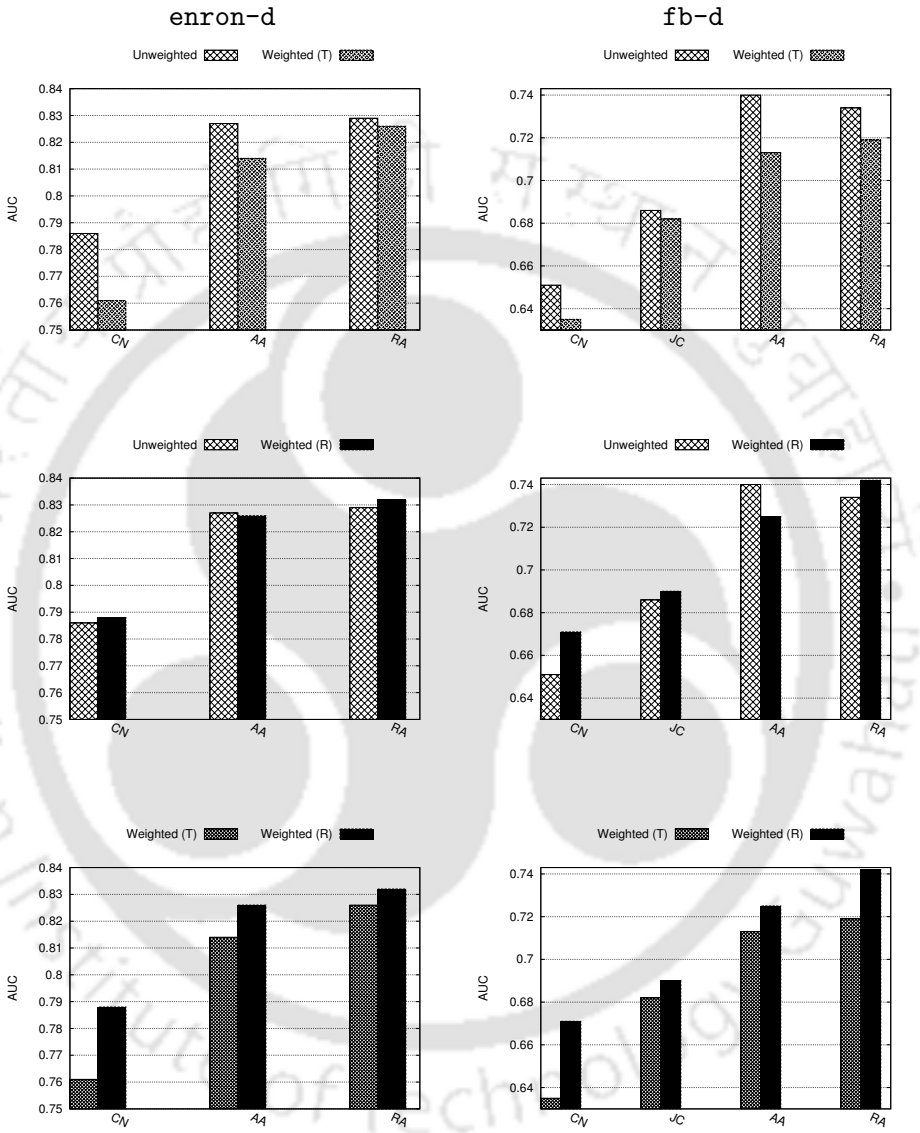


Figure 4.1: Performance of reciprocity-aware weight in link prediction. `Weighted (T)` and `Weighted (R)` represent the traditional and reciprocity-aware weighting method respectively, used for constructing the weighted graphs.

prediction is investigated. Figure 4.1 shows the comparison of link prediction results for the three cases: unweighted, weighted with traditional frequency based weighting method and reciprocity-aware weighting methods over `enron-d` and `fb-d`. Common neighbor (CN), Jaccard's coefficient (JC), Adamic/Adar (AA) and resource allocation (RA) are the baseline prediction methods applied. Results corresponding to JC in `enron-d` is not shown because of its very poor performance (AUC value less than 0.58 for all three cases). `Weighted (T)` and `Weighted (R)` represent the traditional and reciprocity-aware weighting method respectively, used for constructing the weighted graphs. Figure 4.1 shows that prediction performance improves substantially when reciprocity-aware weights are applied as compared to the traditional weights over all prediction methods and datasets. The subplots which lie at the top row of Figure 4.1 is presented to echo the observation of Chapter 3 that frequency based link weighting measure is not suitable for directed networks. The subplots which lie in the middle row of the Figure 4.1 shows that by proper selection of link weighting measure (in this case, reciprocity aware) it is possible to make the weighted prediction methods perform better than their unweighted counterparts. The last pair of subplots demonstrate that reciprocity aware link weight boosts the weighted link prediction performance substantially as compared to that of frequency based link weight. It is also observable that except for AA, all other weighted prediction methods achieve performance boosts over their unweighted counterparts when links are weighted by reciprocity-aware weights. Table 4.4 shows the significance scores of the improvements. The $p-values$ are calculated using *paired t-test* over the AUC values. $p-values$ in the rows corresponding to "Weighted (T) vs. Weighted (R)" indicate that the improvement in the performance of the reciprocity-aware link weight over the traditional link weight is significant. This result shows that weighted prediction methods can outperform their unweighted counterparts in directed networks if proper weighting mechanism is chosen, and reciprocity-aware weight is one such example.

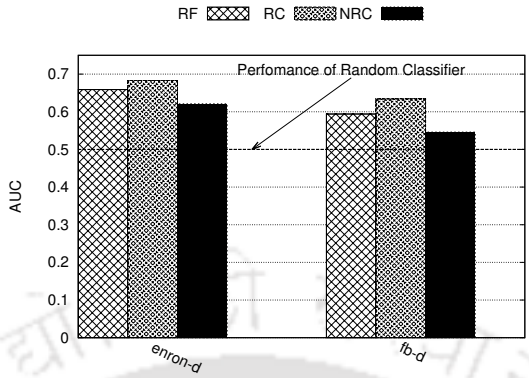


Figure 4.2: Performance of proposed indices.

4.5.2 New link prediction methods

Effectiveness of the proposed similarity indices defined in Subsection 4.4.2 are demonstrated here. Figure 4.2 summarizes performance of the proposed indices towards link prediction in terms of their AUC scores. Among the three methods, RC performs best, because it has positive correlation with the common neighbor method. The normalized version of the RC index (NRC), which is independent to the number of common neighbors, performs better than random classifier (which results in AUC value 0.5). It provides another empirical evidence supporting the importance of the reciprocal links in triad formation. Although RF index returns only three distinct similarity values $\in \{0, 1, 2\}$, its encouraging performance indicates that it may be useful in complex multiple-feature classification models with the cost of little overhead.

4.5.3 Supervised prediction

As we mention in section 4.4.3, the supervised experiments in this work uses $HPLP$ framework [105] which considers *bagging with random forests*, because it is robust against over-fitting in class imbalance scenario. It involves two levels of bootstrapping with replacement, which overcomes over-fitting. We use 15 bootstraps for bagging at the topmost level, and each bootstrap is fed to a random forest[9]. In our experiments, each random forest contains 100 decision-trees. Each experiment is run with 10 different seeds,

[9]R randomForest library is used.

Table 4.5: Supervised models.

Network	Model			
	U	$U+T$	$U+R$	$U+R+P$
`enron-d`	0.854	0.856	0.861	0.864
`fb-d`	0.737	0.741	0.745	0.747

and the average of their performance in terms of their AUC score is presented here.

Table 4.5 summarizes the results obtained from various models, using different combinations of the four feature subsets, named U, T, R and P, defined in Subsection 4.4.3 over the two networks. It shows that, the $U + R$ model which uses features exploiting reciprocity-aware link weights, outperforms its traditionally weighted counterpart $U + T$ in both networks. Usefulness of the proposed link prediction methods is also depicted from the results of the $U + R + P$ model, which outperforms $U + R$.

4.6 Summary

This chapter has exploited reciprocal links to enhance link prediction performance. First, it has produced empirical evidence that reciprocal links take important role on triad formation. This evidence has motivated to propose several unsupervised methods and supervised models which exploit reciprocity. Rigorous experiment has been performed over two real datasets to demonstrate the effectiveness of the proposed methods and models. This chapter has also shown that with proper link weighting technique, weighted link prediction methods can outperform their unweighted counterparts.

From the next chapter onwards, we exploit temporal dimension in link prediction. We model the change in interaction pattern and topological similarity between node pairs using time-series forecasting models towards it.

Chapter 5

Time aware cleaning of dull nodes and links

5.1 Overview

In last two chapters, underlying graphs have been considered as static snapshot of the network. However, in reality, social networks are dynamic in nature. This chapter demonstrates the importance of exploiting temporal information of network formation in link prediction. This is our first step towards *temporal link prediction*.

Evolution of social network [20, 28] has been receiving substantial attention in the field of social network analysis (SNA). A social *tie* (or a link) is built through social interaction between two *actors* (or nodes); and structure of a social network evolves with addition of new nodes and links in the network. Association or friendship between two actors is maintained by the amount of information that flows between them, which is often quantified using the pattern of interactions they are involved in. Most of the studies on the evolution of social network have concentrated on addition of new links and nodes in it. However, in reality, some links become inactive with time as friendships break down, and some nodes withdraw themselves from the network. Inaction or disappearance of those links and nodes alter the structure of the network. We refer such nodes and links as *dull nodes* and *dull links* respectively. Ignoring dull nodes and links

may affect analysis and prediction tasks in the network, because these nodes and links provide spurious information. This work first proposes a novel time aware method to predict dull nodes and links at an early stage as a *preprocessing* task in social networks, and then investigates the effect of these nodes and links in link prediction. We formally define the task of predicting such nodes and links at an early stage by exploiting their historical properties: `by observing a time-evolving social network up to time` τ`, predicting the nodes (and links) which are declared dull at a future time` τ'. The nodes and links, which are idle for long, are labeled as dull. A novel scheme is proposed to fix time duration thresholds, each for nodes and links of a particular network. If a node or link remains idle longer than its respective threshold, it will be labeled as dull. Removal of predicted dull nodes and links from the network reduces noisy information, and is considered as a *data cleaning* step for dynamic networks.

In recent times, researchers have started exploring temporal dimension in SNA. There are studies [50, 53, 113, 114] on two major SNA tasks namely *link prediction* [12] and *community detection* [115], which consider temporal information as an enhancement to baseline methods. These temporal methods inherently carry properties that can discriminate between an active node (or an active link) and a dull node (or a dull link). However, to the best of our knowledge, none of them has formulated and solved the problem of predicting dull nodes (and dull links), nor considered the time-aware data cleaning for dynamic networks. After pruning the predicted dull nodes and links from the network, the preprocessed network can be used for any SNA task using simple non-temporal baseline methods, which may achieve performance gain along with improvement in running time.

Experiments on two real network datasets demonstrate that the proposed method effectively predicts potential dull nodes and links. We further validate our claim of noise reduction by investigating link prediction performance, subject to data cleaning. We show that, the removal of predicted dull nodes and links results in considerable improvement in performance of state-of-the-art link prediction methods for two datasets.

More specifically, contributions of this work are as follows.

- It proposes a novel data cleaning method for dynamic networks. This method

predicts and removes dull nodes and links, which will eventually become inactive or leave the network in future.

- It also proposes a novel scheme to label the dull nodes and links.
- A case study is presented, which demonstrates the effect of the proposed data cleaning method on state-of-the-art link prediction methods.
- Experiments are performed on two real network datasets, which show that the proposed data cleaning method effectively predicts the dull nodes and links, enhancing link prediction performances.

5.2 Existing works on network preprocessing

A brief literature review on existing preprocessing techniques for social networks is presented next. Zhang et al. [116] have proposed SocConnect, which integrates social network data collected from multiple on-line social networking sites. Network data cleaning has been considered in [117–120]. Benevenuto et al. [117] have proposed a method for detecting spammers in Twitter. Bhagat et al. [118] have developed algorithms for anonymizing the actors in social networks in order to preserve privacy. Ferreira et al. [119] have compiled a nice survey on name disambiguation methods for bibliographic citation networks. Huisman et al. [120] have proposed methods to fill the missing informations in network data. Hernández et al. [121] have proposed algorithms to store and retrieve large social network data in compressed format. Macskassy [122] has proposed a method for identifying nodes which leave a particular community in dynamic networks. To the best of our knowledge, none of the existing studies has attempted to predict dull nodes and links in evolving networks as a network preprocessing task.

We have found a few studies [123–125], which passively relate to the concept of dull nodes or links. Kamath et al. [123] have proposed a method to track short-term group formation in on-line social networks like Twitter and Facebook. Raeder et al. [124] and Miritello [125] have predicted “persistence” of links. Given the communication pattern of

a link in a time window, their methods predict whether the link remains active in the next window.

5.3 Problem formulation

This section formulates the problem of predicting dull nodes and links. Social networks typically evolve with social interactions among the actors. When two actors interact, they become connected by a link. With time, pair(s) of actors connected by a link, may interact again. We preserve this history of interaction between two actors by storing the time-stamp of occurrence of the interactions they have been involved in, and associate it to the link formed by them. We represent evolving networks up to time instant t by an undirected graph $G^t = (V, E, T, \mathcal{Q})$, where V is the set of vertices representing actor nodes; $E \subset V \times V$ is the set of edges representing links; T is the set of time-stamps of interactions occurred in the network till t; $\mathcal{Q}$ is a function $\mathcal{Q} : E \rightarrow 2^T$, which returns the historical time-stamps of all interactions associated with a link. For an example, let G^t be a Facebook wall-post network. When a user x posts on another user y's Facebook wall, they become connected by a link (x, y). All such historical wall-posts (up to time instant t) between x and y are stored and retrieved by $\mathcal{Q}(x, y)$.

Given a graph G^t, (x, y) is a *dull link* if the end nodes x and y have not interacted with each other since time $\tau < t$, i.e., $\max \mathcal{Q}(x, y) < \tau$, and the span of being inactive, $t - \tau$ should be sufficiently large so that there is very small probability that they will interact again. Similarly, x is a *dull node* if the node x have not interacted with any other node since time $\hat{\tau} < t$, and the span of being inactive, $t - \hat{\tau}$ should be sufficiently large so that there is very small probability that the node will interact with others again.

Given a graph G^t, the preprocessing task is to predict the nodes and links, which are going to be dull at a future time t', and construct another graph G^t_r by removing these dull nodes and links from G^t.

5.4 Predicting dull nodes and links

Historical activities of nodes and links are modeled to predict the dull nodes and links. First, several time-series representing historical change in baseline node and link properties are prepared and modeled using *simple exponential smoothing* [54] model. Future values of these time-series are forecast to identify features for predicting dull nodes and links. Lastly, vector distance based unsupervised method is applied to predict dull nodes and links.

5.4.1 Preparing time-series

The time-series which we are going to introduce in this subsection, are of two types: *dyadic* and *topological*. Dyadic time-series encapsulate the temporal characteristics relating *dyads*, i.e., the historical interaction between nodes, whereas topological time-series deal with change in graph's topological measures like degree of a node, number of common neighbors of a pair of nodes, etc. Given a graph G^t, time is discretized into a sequence of contiguous time-windows of constant size Δ for a particular network. A time-window ending at time $t - \Delta i$ is denoted as $w_{-i} : i \in \mathbb{N}$. All the time-stamps that fall in $\big(t - \Delta(i+1), t - \Delta i\big]$ define the window w_{-i}. w_0 represents the most recent window, and w_{-i} represents the $(i+1)^{th}$ last window. We populate the contiguous time-windows with some value $\in \mathbb{R}$ to form a time-series. The method of populating the time-windows of dyadic and topological time-series is described next.

The window $w_{-\delta}$ of a dyadic time-series $\mathscr{D}(x, y)$, defined on a link (x, y) in G^t, is populated as:

$$\mathscr{D}_{-\delta}(x, y) = \big|\{\tau : \tau \in \mathcal{Q}(x, y) \wedge \tau \in \big(t - \Delta(\delta + 1), t - \Delta\delta\big]\}\big|, \tag{5.1}$$

which gives the number of interactions between the end nodes during time-window $w_{-\delta}$. $\mathscr{D}(x, y) = \big\langle \mathscr{D}_{-q}(x, y), ..., \mathscr{D}_{-1}(x, y), \mathscr{D}_0(x, y)\big\rangle$, $q \in \mathbb{N}$, gives the dyadic time-series of the link (x, y), where w_{-q} is the time-window when the link appeared.

Similarly, the window $w_{-\delta}$ of a dyadic time-series $\mathscr{S}(x)$, defined on a node x in G^t, is populated as the number of interactions between node x and its neighbors $\Gamma(x)$ in G^t

during time-window $w_{-\delta}$:

$$\mathscr{S}_{-\delta}(x) = \sum_{y \in \Gamma(x)} \big|\{\tau : \tau \in \mathcal{Q}(x,y) \wedge \tau \in \big(t - \Delta(\delta+1), t - \Delta\delta\big]\}\big|, \tag{5.2}$$

To construct topological time-series of nodes, we consider two node properties namely *degree* and *clustering coefficient (CC)*[1]. We define time-series data for *degree* and *CC* corresponding to node x in a time-window $w_{-\delta}$ as follows:

$$d_{-\delta}(x) = d^{t-\Delta\delta}(x) - d^{t-\Delta(\delta+1)}(x), \tag{5.3}$$

$$c_{-\delta}(x) = c^{t-\Delta\delta}(x) - c^{t-\Delta(\delta+1)}(x), \tag{5.4}$$

where $d^{t-\Delta\delta}(x)$ and $c^{t-\Delta\delta}(x)$ give the *degree* and *CC* of node x respectively in the snapshot of G^t at $t - \Delta\delta$, $G^{t-\Delta\delta}$. We denote these two time-series as $d(x)$ and $c(x)$ respectively. Common neighbor (CN) index is the baseline property which is used to construct the topological time-series for links. We define time-series data for CN corresponding to link (x, y) in a time-window $w_{-\delta}$ as follows:

$$C_{-\delta}(x,y) = CN^{t-\Delta\delta}(x,y) - CN^{t-\Delta(\delta+1)}(x,y), \tag{5.5}$$

where $CN^{t-\Delta\delta}(x,y)$ gives the common neighbor score between node-pair x and y respectively in $G^{t-\Delta\delta}$. We populate the values for $C_{-\delta}(x,y)$ in the sequence of contiguous time-windows that starts with the window where the first common neighbor of x and y appeared, and ends with w_0. This time-series is denoted as $C(x,y)$.

5.4.2 Modeling the time-series

The time-series defined in Subsection 5.4.1 are used to forecast future trends. We apply *simple exponential smoothing* time-series forecasting method to model the time-series. The exponential smoothing method gives high importance to the recent activities, and importance decays exponentially from recent to less recent past. Let $\mathscr{Q}_{-\delta}$ denotes the

[1] CC represents the local density in the neighborhood of a node. It is quantified by the ratio of the number of interconnections among the node's neighbors, and the number of maximum possible interconnections [2].

data corresponding to window $w_{-\delta}$ of a time-series $\mathscr{Q}$. Forecast equation of the simple exponential smoothing model for Q can be given by following recurrence equation [54]:

$$\mathscr{Q}'_{-\delta+1} = \alpha\mathscr{Q}_{-\delta} + (1-\alpha)\mathscr{Q}'_{-\delta}, \tag{5.6}$$

where $\mathscr{Q}'_{-\delta}$ gives the forecast value for time-window $w_{-\delta}$, given the time-series data present in previous time-windows; and $0 < \alpha \leq 1$ is called smoothing parameter. When $\alpha \to 0$, simple exponential smoothing gives same weightage to every window during forecast. As the value of α increases from 0 to 1, importance of the recent time-windows increases monotonically. As $\alpha \to 1$, importance of older windows decreases, and $\alpha = 1$ forecasts the value present in the last time-window. Equation 5.6 is used to forecast the value in the window w_1 representing the window just after time t, can be written as:

$$\mathscr{Q}'_1 = \textstyle\sum_{i=0}^{q} \alpha(1-\alpha)^i \ \mathscr{Q}_{-i} + (1-\alpha)^{q+1}\mathscr{Q}'_{-q},$$

where $\mathscr{Q}'_{-q} = \mathscr{Q}_{-q}$ gives the first forecast, w_{-q} being the oldest window. α is estimated by minimizing the sum of the squared errors (SSE):

$$SSE = \textstyle\sum_{j=q-1}^{0} (\mathscr{Q}_{-j} - \mathscr{Q}'_{-j})^2 = \sum_{j=q-1}^{0} e^2_{-j} = e^2,$$

where $e_{-j} = \mathscr{Q}_{-j} - \mathscr{Q}'_{-j}$ gives the error in window w_{-j}. Here, minimizing SSE is a nonlinear optimization task. We use vertical least square fitting procedure that estimates α by solving the equation:

$$\tfrac{d}{d\alpha}(e^2) = 0 \implies \textstyle\sum_{j=q-1}^{0} \tfrac{d}{d\alpha}(\mathscr{Q}_{-j} - \mathscr{Q}'_{-j})^2 = 0.$$

5.4.3 Feature generation and unsupervised method

In this subsection, we propose an unsupervised method to predict dull nodes and links in a given network G^t, which are declared dull at a future time t'. Several features based on temporal change in dyadic and topological time-series are proposed here. These features capture the distinctive properties of a node (and a link) which is likely to be inactive or disappear, based on recent trends. As an example, from a topological perspective, if a node x does not make new connections lately, or there is a decreasing trend in making new

connections, it may be a potential candidate of being dull in future. This is reflected in the time-series $d(x)$. Similarly, the trend in activity of a node x in recent times is reflected in the time-series $\mathscr{S}(x)$. In case of links, if there is a decline in adding new common neighbors between the end nodes in recent time, they might break their relationship in near future. The forecast value of these time-series, generated using simple exponential smoothing model depicts how the nodes and links will behave in future in terms of the underlying node and link properties. We propose a number of features to predict dull nodes and links exploiting these forecast values. Table 5.1 summarizes the features proposed in this work. Suffix 1 in each of the feature values represents w_1, and the values are the forecast values for w_1 of the corresponding time-series (refer to Subsections 5.4.1 and 5.4.2). $zeros_{node}$ and $zeros_{link}$ give the number of trailing zeros in dyadic time-series of the target node and the target link respectively. The $pref_{link}$ feature for a link is a combination of forecast values of the *degree* time-series of the two end nodes. This feature is the temporal version of the preferential attachment property of real networks [65].

A node or a link is described by its feature vector. *Min-Max normalization* of boundary $[0, 1]$[2] is applied to all features. For simplicity, we use distance based unsupervised method to predict the dull nodes and links. We identify a *reference vector* that represents a dull node (and a dull link), and calculate its distance from each sample's feature vector. *Chebyshev distance* is used to measure the distance between the samples' feature vector and the reference vector. Chebyshev distance of the feature vector (say $\boldsymbol{X}$, where its elements are denoted as x_i's, i being integer values ranging from 1 to $|\boldsymbol{X}|$) and the reference vector (say $\boldsymbol{V}$, where its elements are denoted as v_i's) is calculated as follows:

$$D_{chebyshev}(\boldsymbol{X}, \boldsymbol{V}) = \max_{i=1}^{|\boldsymbol{X}|} |x_i - v_i|. \tag{5.7}$$

$(1 - D_{chebyshev}(\boldsymbol{X}, \boldsymbol{V}))$ gives the sample's *proximity score* with a dull node (and a dull link). The values of all features other than $clust_{node}$, $zeros_{node}$ and $zeros_{link}$ of the reference vector are set to 0, because these features should have negative correlation with the dull ones. Corresponding values of $clust_{node}$, $zeros_{node}$ and $zeros_{link}$ are set to 1.

[2]It reassigns a feature value v of feature V as: $\frac{v - \text{minimum value in } V}{\text{maximum value in } V - \text{minimum value in } V}$

Table 5.1: Time series data and corresponding features.

	Value	Feature
Node x	$\mathscr{S}_1'(x)$	$strength_{node}$
	—	$zeros_{node}$
	$d_1'(x)$	deg_{node}
	$c_1'(x)$	$clust_{node}$
Link (x, y)	$\mathscr{D}_1'(x, y)$	$strength_{link}$
	—	$zeros_{link}$
	$d_1'(x)$	deg^x_{link}
	$d_1'(y)$	deg^y_{link}
	$d_1'(x) \times d_1'(y)$	$pref_{link}$
	$C_1'(x, y)$	$common_{link}$

Table 5.2: Dataset Specification

			γ	
Network	#Nodes	#Links	Node	Link
`fb-t`	44937	166511	13	18
`dblp-t`	1304128	5258985	14	10

After predicting the potential dull nodes and links, we remove them from G^t, for data cleaning task.

5.5 Datasets

Experiments are carried out on two time-stamped social networks of different characteristics: Facebook wall-post network [111] (`fb-t`), and DBLP co-authorship network (`dblp-t`). `fb-t` is the temporal version of `fb-d` network, used in Chapter 4. The data is available with Unix time-stamp representing the time of each wall-post, and the information of who is posting on whose wall. Each wall-post is considered as an interaction. We ignore the direction of interactions in order to prepare an undirected graph so that our method can be applicable on it. `dblp-t` dataset has been downloaded from the web-link `http://dblp.uni-trier.de/xml/` in .xml format. It contains the publication information in the field of computer science from the year of 1936 upto the starting of 2014. Early years contain very few publications. So, in this work we consider only the publications which occurs during the years $1980-2013$. The graph is constructed following similar procedure described in Section 3.3. `fb-t` and `dblp-t` consider a *month* and a *year* long time-windows respectively. A summery of characteristics of the networks is presented in Table 5.2.

5.6 Dull nodes and links

Given a network G^t, after predicting the nodes and links which becomes dull at a future time $t' = t + \Delta i$, we need to evaluate our method against true dull nodes and links in

$G^{t'}$. The concept of dull nodes and links is subjective, which is defined as the nodes and links kept silent for long, probably never interact in future. In this section, we propose a scheme to label the dull nodes and links by using the statistical property of the network $G^{t'}$.

To label a dull node, we consider the dyadic time-series ($\mathscr{S}(x)$) of all nodes upto time-window w_0 (with respect to $G^{t'}$). A node is labeled dull if it has not interacted with others for a long time, i.e., its dyadic time-series ends with large number of consecutive zeros. Subsequently, question arises; *how large that number should be, so that we label it as a dull node?* Here we define the number by exploiting the distribution of zero-burst (a series of consecutive zeros) patterns present in all nodes' time-series. It is explained using a toy example shown in Figure 5.1, which presents the time-series data of two nodes in $G^{t'}$. The first one from the top has all zeros in last eight windows; so it may be labeled dull in

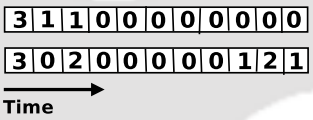
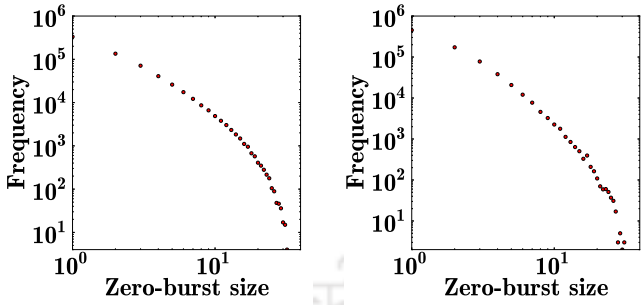


Figure 5.1: A toy example.

$G^{t'}$, if a dull node is defined by at least eight or more number of consecutive zeros at the end of the dyadic time-series. The second one is having two series of intermediate zeros (or zero-burst) of lengths one and five. The tail of the distribution of such zero-burst lengths over the time-series data of all nodes in $G^{t'}$ gives a view about *watching how many consecutive zeros, one can be confident enough that the node may not interact in future, ie., is dull.* Figure 5.2 shows the length distribution of intermediate zero-bursts for the `dblp-t` bibliographic network. Linearity shown in the plots, which are drawn in $log - log$ scale, indicates that the distributions follow power law. This finding supports the observation of [126] that *inter-event time*[3] distribution follows power law. Figure 5.2 indicates that, in real networks, the distribution function decreases rapidly with the increase of zero-burst length, and there are few cases of high number of intermediate zero bursts. This fact

[3]Inter-event time is the time interval of occurrence of two consecutive events in a dynamic network.



(a) Node (b) Link

Figure 5.2: Log-log plot of the frequency of zero-burst patterns for DBLP bibliographic network.

encourages us to label dull node as: let $F(z)$ be the *cumulative distribution function* of the intermediate zero-burst lengths in the time-series data, over all nodes for a given network $G^{t'}$. `A node` x `in` $G^{t'}$ `is labeled dull iff its time-series data is having at least` γ `number of consecutive zeros at the end of its dyadic time-series such that,` $F(\gamma-1) \geq \beta$, `where` $0 < \beta < 1$ `is a constant representing the confidence level.` Without loss of generality, we set $\beta = 0.99$. Similarly, the class labels for *dull link* is labeled using the dyadic time-series data of all edges in $G^{t'}$. Table 5.2 presents γ values for `dblp-t` and `fb-t`.

Relation between dull nodes and links: By definition, dull nodes and dull links are closely related. Let, in a given network, γ_n represents the parameter γ to define dull nodes, and γ_l represents the parameter γ to define dull links. Let x be a node in the network. When $\gamma_n < \gamma_l$, if $\mathscr{S}(x)$ has γ_n number of trailing zeros, it is possible that some of the links connecting x to its neighbors may not qualify as dull. Hence, predicting and pruning x removes those links too, which would not have been possible if predicting and pruning only the dull links were considered. Pruning the potential dull nodes adds an extra level of filtering.

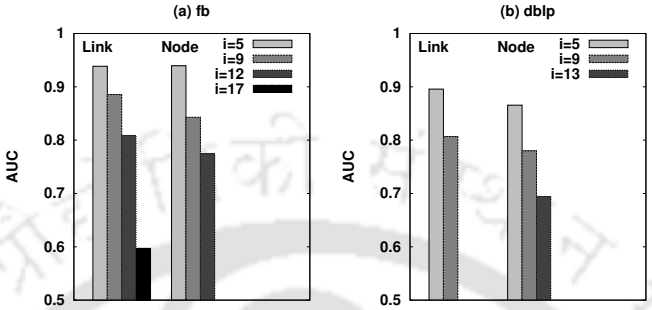
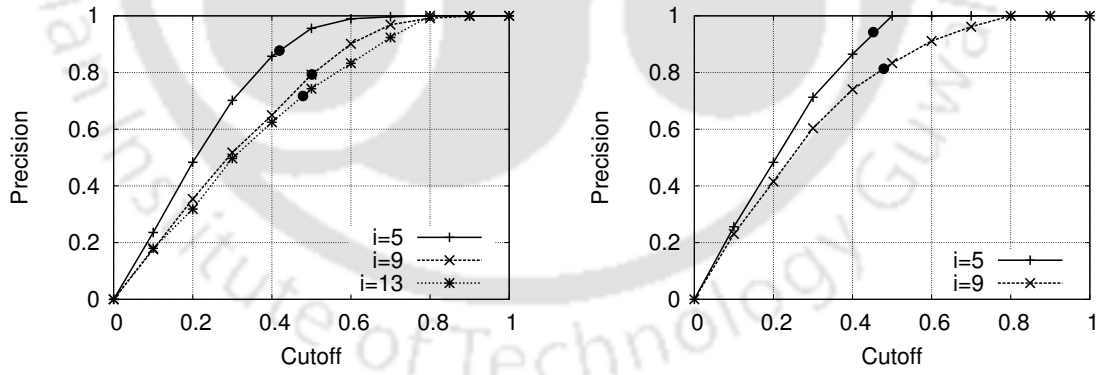
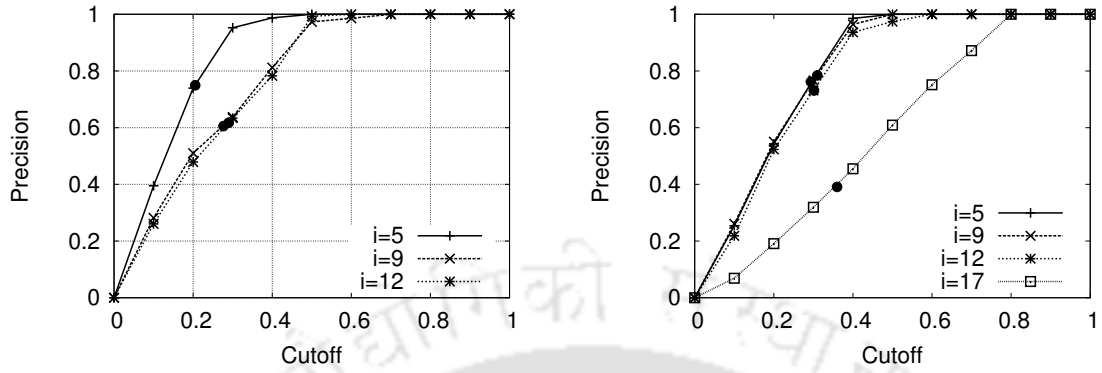


Figure 5.3: Performance of dull nodes and links prediction at time $t' - \Delta i$ in terms of AUC score.



(a) Node (b) Link

Figure 5.4: Precision curve for dull nodes and links prediction in `dblp-t`.



(a) Node (b) Link

Figure 5.5: Precision curve for dull nodes and links prediction in `fb-t`.

5.7 Evaluating proposed method

In this section, we evaluate the method proposed in Section 5.4. Experiments are performed over several snapshots: $G^{t'-\Delta i}, i < \gamma$, where $G^{t'-\Delta i}$ represents the given network G^t and the task is to predict the nodes and links which are declared dull at t'. Class imbalance is an inherent issue in this problem, because there are very few number of dull nodes and links as compared to the nodes and links which are not dull. We handle class imbalance by selecting only the nodes (and links) which have not interacted in window w_0 and w_{-1} (with respect to G^t) to represent the non-dull nodes (and links). In this work, we pick top k number of samples in terms of their proximity score (described in Subsection 5.4.3) as potential dull nodes (and links), where k represents the number of true dull nodes (and links). Figure 5.3 presents the prediction performance in terms of *area under receiver operating characteristic (ROC) curve* (AUC scores) [127] for both datasets. High AUC scores in the results demonstrate that potential dull nodes and links can be predicted effectively. Prediction performance for the lower values of i is much better than the higher ones, because a high number of zeros at the end of all time-series of the dull samples boosts the performance when the value of i is small. Figures 5.4 and 5.5 present the prediction performance in different thresholds in terms of their *precision* scores. The

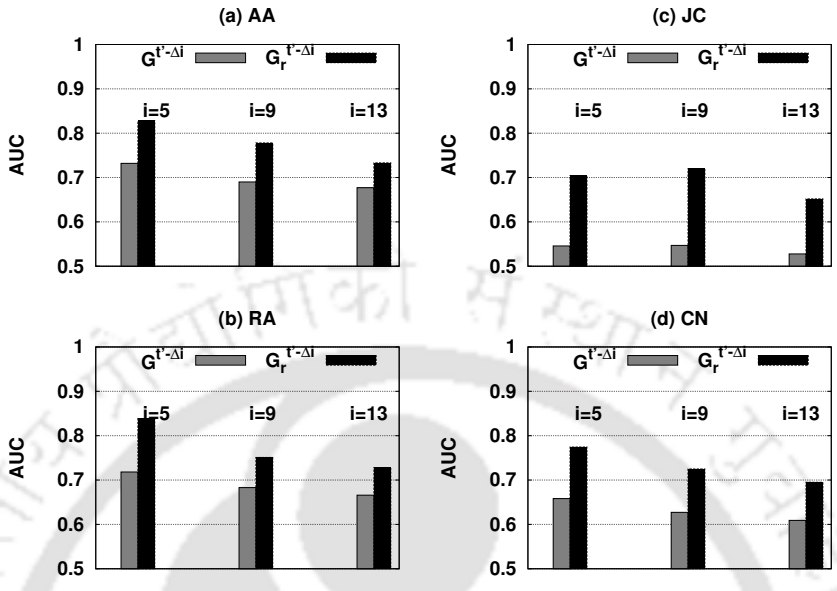


Figure 5.6: Link prediction performance (`dblp-t`)

black dots in each plot represent *precision@k* values, k representing the number of true dull ones (the pruning threshold). The precision plots demonstrate that the proposed method achieves very high precision before reaching 0.5 cutoff value. As we set k as the cutoff in our prediction mechanism, *precision@k* governs the performance of our method in practical scenario, which is basically a retrieval process. The plots show that our method can achieve high *precision@k* values. Although rise of the precision curves for `fb-t` is more rapid than `dblp-t`, *precision@k* values win in `dblp-t`. It is also notable that, in spite of overcoming class imbalance, the proportion of dull nodes and links are much less in `fb-t`, which is another cause of performance degradation at the retrieval point.

5.8 Data cleaning and link prediction: a case study

After predicting the dull nodes and links from $G^{t'-\Delta i}$, we investigate the effect of cleaning the datasets by removing those nodes and links, on identifying new links that may appear in $G^{t'}$. We first sort the proximity scores for all samples (links and nodes) and consider

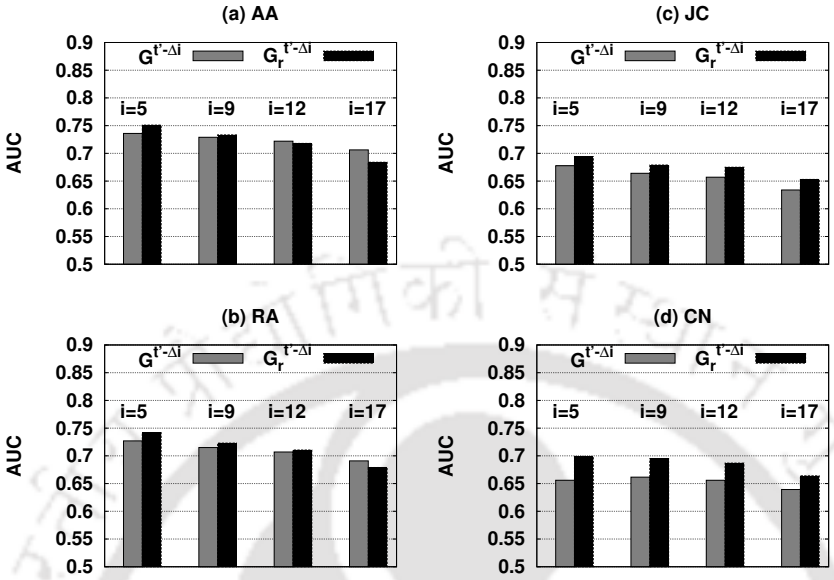


Figure 5.7: Link prediction performance (`fb-t`)

a number of best performing k samples, where k represents the number of labeled dull nodes and links, as the potential candidate of being dull. We prepare a graph $G_r^{t'-\Delta i}$ by removing the predicted dull nodes and links from $G^{t'-\Delta i}$. We perform link prediction on both of $G^{t'-\Delta i}$ and $G_r^{t'-\Delta i}$ using state-of-the-art link prediction methods, and compare their performance in terms of AUC scores. We consider four baseline unweighted link prediction methods: *Common neighbor (CN), Jaccard's coefficient (JC), Adamic/Adar (AA), and Resource allocation (RA)*, described in Chapter 2. We perform the experiments for multiple values of i. Figures 5.6 and 5.7 summarizes the link prediction results for `dblp-t` and `fb-t`. These figures show that prediction performance in the reduced graph $G_r^{t'-\Delta i}$ improves significantly for almost all values of i's over both the datasets. `dblp-t` network is benefited more than `fb-t` by data cleaning. It can be justified by its higher *precision@k* values in dull link prediction (refer to Section 5.7). In most of the cases, over all link prediction methods and datasets, effect of data cleaning increases with increase in the value of i. In `fb-t`, sometimes data cleaning worsens the link prediction when $i = 17$. It is expected because, as shown in Figure 5.5, *precision@k* value for dull link prediction

is as low as .39 in $G^{t'-17\Delta}$. Among the link prediction methods, JC and CN are the most affected prediction methods, whereas AA and RA are more robust against noise.

5.9 Summary

In this chapter, we have proposed a time aware network data cleaning method. The proposed method predicts dull nodes and links, based on historical network properties. It then prunes them from the network. This work has also proposed a scheme to label dull nodes and links for evaluating the proposed method. Exhaustive experiments on two real network datasets have shown that the proposed method can effectively predict the potential dull nodes and links. It further has demonstrated the effect of network data cleaning on state-of-the-art link prediction methods. Significant improvement in link prediction performance has been observed when the proposed data cleaning method is applied on real datasets. This observation motivates us to incorporate temporal measures for link prediction in following chapters.

Chapter 6

Temporal link prediction in heterogeneous networks: a supervised approach

6.1 Overview

State-of-the-art link prediction methods mentioned in Chapter 2 apply on static snapshot of a network. A static snapshot is represented as a graph, where the nodes represent actors, and two actors share an edge if a relationship exists between them. Relationships are built through interactions between actors, and it is assumed that if two actors interact at least once, they share a link. Sometimes, the graph is weighted, where edge-weights represent strength of relationships, usually quantified by the total number of interactions between the end nodes. Chapter 3 has investigated effect of link-weights on link prediction methods. Chapter 5 has shown that temporal history of interactions can improve the link prediction performance. In this chapter, we combine temporal history of interactions, and heterogeneous properties of social networks towards solving the link prediction problem.

6.1.1 Background literature and motivation

Initial studies on link prediction focus on static and homogeneous networks. These studies construct a static graph by ignoring temporal information embedded in it, and apply several unsupervised and supervised techniques to predict future links. Lü et al.summarizes a nice survey on such studies [49]. However, valuable information is missed by ignoring the temporal dimension. New nodes and links appear in a dynamic network as it evolves with time. Existing nodes and links also disappear. Such information can not be embedded in static snapshot of a graph. Some of the recent studies [50–53] consider temporal dynamics of the underlying graph in order to predict future links. Tylenda et al. [50] have exploited time-agnostic functions to model a relationship and used it to modify the state-of-the-art link prediction methods. Potgieter et al. [51] has used recency of a node in communicating with others as a parameter for link prediction. Tensor factorization based temporal link prediction method has been proposed in [52], where the third dimension of a tensor has been used to embed temporal information. In [53], Richard et al.have assumed that social networks evolve with stationary dynamics, and modeled the history of graph properties using auto-regressive models for dynamic link prediction. All of these studies develop their link prediction approaches over homogeneous networks, where only one type of node and a single type of relationship among the nodes is considered.

Apart from temporal dynamics, many of the dynamic networks like *Facebook*[1] online social network, bibliographic network, etc., are heterogeneous in nature, where multiple types of nodes play heterogeneous roles in the network, and connect to each other creating various types of links. In bibliographic network, various types of nodes such as *author*, *paper*, $conference$ exist, which represent authors, papers and conferences respectively. When an author writes a paper in a conference, corresponding nodes representing the paper, author and conference connect among themselves with heterogeneous links. Authors also get connected among themselves through paper and conference nodes. When multiple authors write a paper collaboratively, the author nodes form a clique of co-authorship links, which are used to build co-authorship network. Similarly, two authors

[1] `https://www.facebook.com`

may also be connected through conference nodes when they publish or attend same conference. Predicting different types of link represents multiple prediction tasks; such as predicting which author pairs are likely to co-author a paper in future, or predicting whether an author is likely to write a paper in a particular conference or not, etc. These can be considered as different prediction problems. Moreover, for predicting a link of a particular type, information-flow through links of other types may be useful. Recently, researchers have started exploring link prediction problem in heterogeneous networks [43, 48, 128], which exploits heterogeneous information embedded in networks in order to predict future links of single or multiple types. Sun et al. [43] have proposed a meta-path based topological measure, Lichtenwalter and Chawla [128] has proposed vertex collocation profile, and very recently Ermi et al. [48] have used tensor factorization method for link prediction in heterogeneous networks. Heterogeneous networks are referred as *multi-relational*, *multi-dimensional*, *multi-modal* networks as well, depending on underlying applications. From this point onwards we shall use the *multi-relational* word to represent heterogeneous networks in order to avoid ambiguity.

Although temporal dynamics and multi-relational property of social networks have been explored independently to solve the link prediction problem, attempts of combining both is rare in literature. Work reported in Yang et al. [44] is one such study. It has proposed a number of temporal and multi-relational features for supervised link prediction. Experimental results of Yang et al.shows that combining both types of features improve link prediction performance. The features of Yang et al.are based on historical preferences of nodes defined in terms of some network properties like *degree*, number of *common neighbors*, etc. In this chapter, we show that historical preferences based approach is not appropriate for link prediction in real networks, and propose a robust and efficient approach for temporal link prediction in multi-relational networks.

6.1.2 Contributions

We apply window-based time-series approach to model several time-series obtained from various multi-relational properties of the network. We use *simple exponential*

smoothing [54] method to model the time-series. The main assumption of the simple exponential smoothing model is that influence of the historical data on the current time-step increases exponentially with time. The model has only one parameter, which makes the model simple and efficient in terms of parameter estimation. Moreover, it does not require stationarity assumption of auto-regressive models, which Richard et al. [53] have adapted for modeling the evolution of networks. Another non-stationary model called *auto-regressive integrated moving average (ARIMA)* [90] is sometimes used to model time-series in social network [34]. However, it is very complex in nature with a lot of parameters, and thus prone to over-fitting. It does not scale for large network dataset.

We categorize the time series into two: *topological*, and *dyadic*. Topological time-series encode the historical information of static graph topology based measures such as CN, AA, etc., whereas dyadic time-series describe the history associated to dyads (such as interaction between two nodes). In our approach, dyadic time-series, which have been ignored in most of the existing studies on temporal link prediction, is the key for combining temporal and multi-relational properties of networks towards link prediction. We identify several features by embedding time-series methods to baseline link prediction methods, and apply supervised learning in order to predict future links. We call the feature set as *time-aware multi-relational link prediction (TMLP)* feature set.

We perform rigorous experiments over two bibliographic datasets to show the effectiveness of our method. We first analyze unsupervised performance of individual features, and then apply supervised learning method that combines multiple features towards link prediction. Bagging with random forests supervised learning framework is used in our experiments, because it deals high class imbalance of the link prediction problem effectively [42]. An inherent problem of supervised link prediction is that, due to temporal dependence of dataset, the time-line used for training can not be used for testing. Time-line for testing must be shifted towards future, which causes longitudinal bias. It does not let the model learned during training period to fit best for the testing period. It affects link prediction performance. We show that proposed feature set has high robustness against longitudinal bias. Rigorous supervised learning experiments with

multiple combinations of the TMLP features are performed to show the effectiveness of considering dynamics and multi-relational property, and need for combining the two for link prediction. Experiment also reveals that dyadic history indeed helps a lot in link prediction. We also compare our results with the models proposed in Yang et al. [44] and Lichtenwalter et al. [42], which shows that our method outperforms the existing studies significantly. Our contributions in this chapter are summarized as follows:

- We propose a robust and efficient set of features named *time-aware multi-relational link prediction (TMLP)* features to predict future links using supervised learning framework in dynamic multi-relational network.
- TMLP feature set exploits topological and dyadic history to combine dynamics and multi-relational property of a network.
- Rigorous experiment over two bibliographic networks is performed to demonstrate the effectiveness of our approach. Unsupervised performance of individual features are compared. Supervised learning is performed using multiple models with several combinations of the proposed feature set, which shows significant improvement in prediction performance of multi-relational over single-relational prediction, and temporal over static prediction. It also demonstrates that combining multi-relational and temporal properties performs superior than when considered individually.
- We empirically demonstrate that the proposed feature set is able to handle longitudinal bias effectively.
- We compare our model with two recent and relevant studies to show that our model predicts future links more accurately than other existing studies.

6.2 Defining temporal link prediction problem in multi-relational network

A multi-relational network can be represented as a multi-graph, which allows multiple edges between node-pairs. Each link is assigned a label denoting the type of relationship

binding the two nodes. Different labels are due to several classes of activities or events occurring inside the network. Each node is associated with some event(s) of each class. Two nodes get connected by a link of a particular type (i.e. label) when the end nodes participate in a common event of that type. We define a time aware multi-relational network at time t by an undirected multi-graph with temporal information as $G^t = (V, L, E, T, \mathcal{R})$, where:

- V is the set of vertices representing nodes.
- L is the set of labels denoting the relationship types.
- $E \subset V \times V \times L$ is the set of edges, each of which represents link of particular type. We denote a link of type l that connects nodes x and y by a 3-tuple (x, y, l).
- T is a set of time-stamps which preserves the time instants of occurrence of all events on or before time t.
- $\mathcal{R}$ is a relation, $\mathcal{R} : E \rightarrow \{(i, X_i, Y_i) : i \in I, I \subset \mathbb{N}, X_i \in 2^T, Y_i \in 2^T\}$. Given a link (x, y, l), $\mathcal{R}(x, y, l)$ returns a set of 3-tuples. Each of the 3-tuples corresponds to an event of type l, where both of the nodes x and y have participated. Each event of a particular relationship class l is represented as a unique identifier, given by a natural number. X_i and Y_i respectively are the sets of timestamps depicting the time instants when x and y have participated in event i. Relation $\mathcal{R}$ carries historical information about the dyad formed by the end nodes, where the events keep the dyad tied. We refer the set returned by $\mathcal{R}$ as *dyadic history*. It is explained with appropriate example later in this section.

Given a multi-graph $G^t = (V, L, E, T, \mathcal{R})$, the link prediction problem is defined as: given two nodes $x, y \in V$ and a relationship type l, such that $(x, y, l) \notin E$, predicting whether the nodes will be connected by a link of type l in future or not. We call x, y as target nodes, and l as target link-type.

In this work, we propose a temporal and multi-relational feature set to solve the temporal link prediction problem in multi-relational bibliographic network. We refer the

proposed feature set as *time-aware multi-relational link prediction (TMLP)*. Our method can be easily extended for any multi-relational network. We frame the problem for bibliographic network in following subsection.

6.2.1 Focusing on bibliographic network

In a multi-relational bibliographic network G^t, nodes represent authors. We consider three classes of events: *paper, conference* and *keyword* happening inside G^t, which are responsible for three types of link, i.e. $L \in \{paper, conference, keyword\}$. Each paper, which has been written collaboratively at least by two researchers, represents a paper event. Each conference[2] depicts a conference event. Keyword events are the unique keywords describing the subject areas over all collaboratively written papers. All events of each of these three event classes has an unique identifier, given by a natural number. Two authors become connected by a paper link when they collaboratively write a research paper. Similarly, if two authors write paper(s) in same conference, they share a conference link; and if their papers contain at least one common keyword, they share a keyword link. Conference and keyword links represent relationship of two authors based on their subject area. The set T gives the time instants of occurrence of all events. $\mathcal{R}(x, y, l)$ defines the temporal characteristics of the link (x, y, l). For an example, suppose nodes x and y are connected by a link of type conference, and $\mathcal{R}(x, y, conference) = \{(1, \{1994, 2001, 2010\}, \{2002, 1013\}), (8, \{1999, 2000\}, \{2003, 2014\})\}$. It means that, both of the authors x and y have published some paper(s) in the conferences 1 and 8. Node x has published papers in conference 1 in the years of $1994, 2001$ and 2010, where node y has published in the same conference in 2002 and 1013. Similarly, x and y have published some paper(s) in conference 8 in the years of 1999 and 2000, and 2003 and 2014 respectively. Temporal activities of keyword and paper events are also preserved by the relation $\mathcal{R}$ for the links of corresponding types. In case of keyword links, it returns the common keywords describing the publications of the end nodes, along with the publication year of

[2]If conference A has taken place in multiple years, the conference A represents a conference event, not each of its versions.

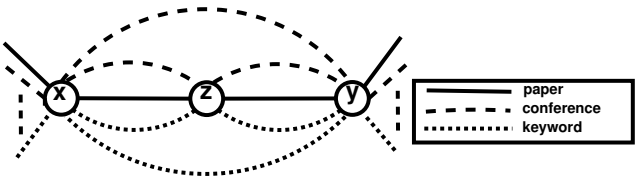


Figure 6.1: Connection setup in a typical bibliographic network

such papers that are described by the keywords. Interpretation of $\mathcal{R}$ is a little different for paper links. Each collaborative publication itself is a paper event, and a common paper event depicts same time-stamp for all of the collaborating authors. Thus, for a link $(x, y, paper)$, and for each tuple (i, X_i, Y_i) corresponding to each common paper event i, $X_i = Y_i$.

In this paper, we predict whether two authors in a bibliographic network will collaboratively write a research paper in future or not with the help of the constructed multi-graph, i.e., *paper is the target link-type*.

6.3 Preparing TMPL feature set

In this section, we propose time-aware multi-relational link prediction (TMLP) feature set, which is used in supervised learning framework to predict future links. Let us explain the feature generation process with a toy example presented in Fig. 6.1. It shows a possible connection setup among three nodes in a typical bibliographic network. Nodes x and y are connected with node z by a paper link, but they do not share any paper link between themselves. Traditional uni-relational link prediction approaches consider that node z influences the likelihood of x and y collaboratively writing a paper in future; and important information flows through the paper links (x, z) and (y, z). However, they miss the fact that a link of type other than target (i.e., paper) may also take important role in finding the likelihood. For an example, let nodes x and y are having high association with node z in terms of common keywords present in the title of the papers they write. This fact reflects that both x and y work in similar subject areas as of z. Therefore, z may introduce

Table 6.1: Topological link prediction methods.

Method	Unweighted	Weighted
Common Neighbor	$\lvert\Gamma(x) \cap \Gamma(y)\rvert$	$\sum_{z \in \Gamma(x) \cap \Gamma(y)} (w_l(x,z) + w_l(z,y))$
Jaccard Coefficient	$\frac{\lvert\Gamma(x) \cap \Gamma(y)\rvert}{\lvert\Gamma(x) \cup \Gamma(y)\rvert}$	$\frac{\sum_{z \in \Gamma(x) \cap \Gamma(y)} (w_l(x,z)+w_l(z,y))}{s_l(x)+s_l(y)-\sum_{z \in \Gamma(x) \cap \Gamma(y)} \min(w_l(x,z),w_l(z,y))}$
Adamic/Adar	$\sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{log\lvert\Gamma(z)\rvert}$	$\sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{w_l(x,z)+w_l(z,y)}{log(1+s_l(z))}$
Resource Allocation	$\sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\lvert\Gamma(z)\rvert}$	$\sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{w_l(x,z)+w_l(z,y)}{s_l(z)}$
Preferential Attachment	$\lvert\Gamma(x)\rvert \times \lvert\Gamma(y)\rvert$	$s_l(x) \times s_l(y)$

Note 1: $\Gamma(a)$ denotes the set of neighbors (target link) of node a; $w_l(a,b)$ denotes the weight of the link (a,b,l) of type l; $s_l(a) = \sum_{b \in \Gamma(a)} w_l(a,b)$ is the additive strength or weighted degree of node a, considering link-type l.

x to y, and they collaborate thereafter. Moreover, x and y may collaborate without z's influence if they work in similar research areas. Similar argument can be presented to justify the necessity of introducing conference links to find future collaborations. If authors x and y meet in some conference venue, they may get introduced to each other and collaborate later. Even if they do not meet at any conference venue, but publish some paper in same conference (when the publication years of the two authors do not overlap, or at least one of them is absent in the venue in case of any overlap), then also they may collaborate in future as their scope of research matches. Node z may also act as a mediator and facilitate the collaboration. The TMLP feature set considers all the links present in Fig. 6.1 to incorporate multi-relationality in prediction of future collaborations between x and y. Temporal dimension is incorporated with multi-relational properties by assigning temporal weights to all links (of all link-types). Weight of a link represents *strength* of the link, i.e. it quantifies the degree of relationship between the end nodes. Temporal weight incorporates temporal characteristics of a dyad, such as recent trend in the relationship, ups and downs in the relationship, etc. We model the dyadic history (returned by relation $\mathcal{R}$ in the definition of G^t) by time-series method, and forecast the future value of the time-series to get temporal weights. We refer the time-series prepared from dyadic history as *dyadic time-series*. Following two subsets of features are formed using these weights.

(1) Weighted node-proximity based link prediction methods are considered as

features. The individual methods used, along with their weighted counterpart are briefed in Table 6.1. Here we incorporate the multi-relational component to the baseline unweighted and weighted methods presented in Chapter 3. These subset of features considers the influence of node z on the likelihood of x and y collaborating in future. Influence flow through links of all the link-types: paper, conference and keyword is considered to device these features.

(2) In absence of a target link between two authors, it is still possible that they are connected by a link of link-type other than target. In Fig. 6.1, nodes x and y are not connected by a paper link, but they are connected by conference and keyword link. Temporal strength of these links also carries valuable information about the temporal characteristics of matching subject areas of the two nodes. Here we consider temporal weights associated with these two links as another subset of features.

As new nodes and links appear, topology of the network also changes. Change in network topology affects topological similarity measures between nodes x and y like number of common neighbors (CN), preferential attachment (PA), etc. Like the dyadic history, change in topological similarity also affects the likelihood of collaboration between the end nodes. We capture this change in *topological time-series*. We define another subset of features by modeling the topological time-series using time-series method and forecasting the future value. Note that these features are uni-relational, i.e. the topological similarity measures are solely based on the target links.

In bibliographic networks, each paper event associates a very less number of authors[3]. This keeps the co-authorship graph (comprising only paper links) very sparse. However, the graph generated by conference links are much denser than the co-authorship graph for the reason that each conference accepts many papers and spans multiple years, which connects a large number of authors by a clique of conference links. Similar situation arises for keyword graph, because each keyword is also shared by a large number of authors. To maintain sparsity in the dataset, we first build the graph topology using paper links, and then add selected conference and keyword links between nodes having shortest path length

[3]Typical value of the number of authors per scientific paper varies in the range of $2-4$ [16]

≤ 2 (see Fig. 6.1). It also satisfies all the requirements for applying the multi-relational node-proximity based methods given in Table 6.1, without any loss of multi-relational information.

Next, we describe how time-series forecasting methods are used to prepare the proposed features. A taxonomy of the TMPL feature set is also presented subsequently.

6.3.1 Preparing and modeling time-series

In this subsection, we describe the method of preparing the dyadic and topological time-series mentioned earlier. Given a multi-graph G^t, time is discretized into a sequence of contiguous time-windows of constant size Δ for a particular network. A time-window ending at time $t - \Delta i$ is denoted as $w_{-i} : i \in \mathbb{N}$. All time-stamps that fall in $\big(t - (i+1)\Delta, t - i\Delta\big]$ define the window w_{-i}. w_0 represents the most recent window, and w_{-i} represents the $(i+1)^{th}$ last window. We populate the contiguous time-windows with some value $\in \mathbb{R}$ to form a time-series. The method of populating the time-windows of dyadic and topological time-series is described below.

A dyadic time-series $\mathscr{D}(x, y, l)$ is defined on a link (x, y, l) in G^t, where x and y represent the end nodes, and l is the link-type. A window $w_{-\delta}$ of the dyadic time-series of (x, y, l), is populated by,

$$\mathscr{D}_{-\delta}(x, y, l) = \sum_{(i, X_i, Y_i) \in \mathcal{R}(x,y,l)} \big|X_i \cap Y_i \cap w_{-\delta}\big|. \tag{6.1}$$

Each element of the summation contributes the number of participation of x and y in a common event of type l during $w_{-\delta}$, and $\mathscr{D}_{-\delta}(x, y, l)$ sums up the occurrence of all such events to represent *dyadic strength* of link (x, y, l) during $w_{-\delta}$. $\mathscr{D}(x, y, l) =< \mathscr{D}_{-q}(x, y, l), ..., \mathscr{D}_{-1}(x, y, l), \mathscr{D}_0(x, y, l) >$, $q \in \mathbb{N}$, gives the dyadic time-series of the link (x, y, l), where w_{-q} is the time-window when the link appeared in the network. The features that utilize the dyadic time-series to measure the likelihood of two nodes x and y (having at least one common neighbor) collaborating in future uses the following time series:

(a) $\mathscr{D}(x, u, l), \forall u \in \Gamma(x), \forall l \in \{paper, conference, keyword\}$,

(b) $\mathscr{D}(y, u, l), \forall u \in \Gamma(y), \forall l \in \{paper, conference, keyword\}$,

(c) $\mathscr{D}(z, z', l), \forall z \in \Gamma(x) \cap \Gamma(y), \forall z' \in \Gamma(z), \forall l \in \{paper, conference, keyword\}$,

(d) $\mathscr{D}(x, y, l), \forall l \in \{conference, keyword\}$,

where $\Gamma(x)$ depicts the set of all paper link neighbors of node x.

Topological time-series depicts the change in score of the node proximity based topological measures between the end nodes x and y. We consider two types of topological time-series corresponding to two similarity measures: unweighted CN and PA (see Table 6.1). Time-series data for CN and PA respectively in a time-window $w_{-\delta}$ are measured as follows:

$$C_{-\delta}(x, y) = CN^{t-\Delta\delta}(x, y) - CN^{t-\Delta(\delta+1)}(x, y),$$
$$P_{-\delta}(x, y) = PA^{t-\Delta\delta}(x, y) - PA^{t-\Delta(\delta+1)}(x, y),$$

where $CN^{t-\Delta\delta}(x, y)$ and $PA^{t-\Delta\delta}(x, y)$ give the unweighted common neighbor, and unweighted preferential attachment score between node-pair x and y respectively in the snapshot of G^t at $t - \Delta\delta$, $G^{t-\Delta\delta}$. We populate the values for $C_{-\delta}(x, y)$ in the sequence of contiguous time-windows that starts with the window where the first common neighbor of x and y appears, and ends with w_0. For the case of time-series of PA, the starting window is the first window where both of the end nodes appears. We name these two time-series as $C(x, y)$ and $P(x, y)$ respectively.

The time-series defined above are used to forecast future trends. We apply *simple exponential smoothing* time-series forecasting method to model the time-series. The exponential smoothing method gives high importance to the recent activities, and importance decays exponentially from recent to less recent past. Given a time-series Q, where $Q_{-\delta}$ denotes the data corresponding to window $w_{-\delta}$, forecast for the next window, i.e., w_1, Q'_1 is achieved following the same procedure described in Section 5.4.2 in Chapter 5.

6.3.2 Preparing link weight

TMLP feature set exploits dyadic time-series to get the temporal weight of the links. Given a link (x, y, l) and its dyadic time-series $\mathscr{D}(x, y, l)$, the forecast value $\mathscr{D}'_1(x, y, l)$ for

Table 6.2: Taxonomy of the features used in the TMLP feature set.

Sl.	A, B, C	D	
		target	*other*
1	$topo, unwt, static$	CN,JC,AA,RA,PA	None
2	$topo/dyad, wt, static$	SCN_f,SJC_f,SAA_f,SRA_f,SPA_f, SCN_n,SJC_n,SAA_n,SRA_n,SPA_n	SCN_c,SJC_c,SAA_c,SRA_c,SPA_c, SCN_k,SJC_k,SAA_k,SRA_k,SPA_k
3	$topo/dyad, wt, temporal$	TCN_f,TJC_f,TAA_f,TRA_f,TPA_f, TCN_n,TJC_n,TAA_n,TRA_n,TPA_n	TCN_c,TJC_c,TAA_c,TRA_c,TPA_c, TCN_k,TJC_k,TAA_k,TRA_k,TPA_k
4	$topo, unwt, temporal$	TCN,TPA	None
5	$dyad, wt, static$	None	$conf$,key
6	$dyad, wt, temporal$	None	$Tconf$,$Tkey$

the next time-window w_1 is calculated by the method described in the previous subsection, which is assigned as the temporal weight of the link. We define the temporal weight of (x, y, l) as:

$$F(x, y, l) = \mathscr{D}'_1(x, y, l).$$

Along with temporal weight, TMLP feature set also includes features that exploit traditional static weight. The static weight of link (x, y, l) is given by:

$$W(x, y, l) = \sum_{(i, X_i, Y_i) \in \mathcal{R}(x,y,l)} |X_i \cap Y_i|. \tag{6.2}$$

$W(x, y, l)$ deals with number of occurrences over all common events of type l participated by nodes x and y. $F(x, y, l)$ groups such occurrence by time-windows and forecasts the value for next time-window. We call this group of weights as $frequency$. We use another group of link weights for paper links, which assign different weight to each paper event. Each paper event is given a weight that is inversely proportional to the number of authors of that paper [39]. We call this group of link weight as *newman*. For simplicity, we have avoided providing the compatibility of newman weighting scheme for paper events in the definition of G^t, presented in Section 6.2.

6.3.3 Taxonomy of TMLP feature set

In this subsection we list all features we include in the TMLP feature set, and give them proper notations. We categorize the features depending on their characteristics. Each feature-type is represented by a quadruple (A, B, C, D). The element A takes one of three values: $topo, dyad$ and $topo/dyad$ depending on whether the feature-type exploits only the topological property of the graph, only the dyadic history of the links, or both respectively. B takes one of two values: wt if link weight is considered, and $unwt$ otherwise. C is assigned $temporal$ if time dimension is considered, and $static$ otherwise. D takes one of two values: $target$ if it exploits the paper links, and $other$ if it exploits conference or keyword links. A taxonomy of all features present in the TMLP feature set is presented in Table 6.2. Individual features are point-wise discussed next.

(1) $(topo, unwt, static, target)$: These features consists of the five unweighted measures presented in Table 6.1.

(2) $(topo/dyad, wt, static, D)$: These features exploit static weight of links of all types (given by $W(x, y, l)$ for link (x, y, l)) presented in Subsection 6.3.2. The five weighted measures presented in Table 6.1 make this subset of features. Individual features are represented as SX_l, where X is the baseline topological feature (presented in Point (1)) and l is the link-type. Conference and the keyword link-types are represented as c and k respectively. As we consider two different groups of weights for the paper link: *frequency* and *newman*, the paper link is represented by f or n respectively for the two.

(3) $(topo/dyad, wt, temporal, D)$: These features are temporal counterparts of the features presented in Point (2), and exploit temporal weights of links, $F(-,-,-)$ (refer to Subsection 6.3.2). These are denoted by TX_l for the baseline method X and link-type l.

(4) $(topo, unwt, temporal, target)$: These features are derived as the forecast values $C'_1(-,-)$ and $P'_1(-,-)$ for window w_1, corresponding to the time-series $C(-,-)$ and $P(-,-)$ respectively. These features are represented by TCN and TPA respectively.

(5) $(dyad, wt, static, other)$: These features consider the links of a link-type other than target, if present between the target nodes. These features are denoted by $conf$ and key for the conference and keyword links respectively. $W(x, y, l)$ gives the feature value

Table 6.3: Statistics of the dataset.

Network	$\|V\|$	#Paper	#Paperlink	#Conference	#Keyword
`db-t-m`	28133	26380	73744	50	13844
`theory-t-m`	24765	41840	65913	46	17537

where x and y are the target nodes and l is the link-type.

(6) $(dyad, wt, temporal, other)$: These features are the temporal versions of the features of Point (5). Temporal weights of the links of a link-type other than target, given by $F(-,-,-)$, make the feature values. These features are denoted by $Tconf$ and $Tkey$ for the conference and keyword link-types respectively.

6.4 Dataset

We prepare two multi-relational networks from the DBLP bibliographic dataset. We consider the span of $1983 - 2013$ to prepare our dataset. The conference publications are maintained using $< inproceedings >$ tag. The author set, title of the paper, conference name, and the publication year for each of these publications are extracted using the tags $< author >$, $< title >$, $< booktitle >$, and $< year >$ respectively. The text inside the $< title >$ tag is stemmed; and stop-words are removed from it to get the set of keywords. The publication years make the set of time-stamps; and the publications, the conference names, and the set of keywords are used to make the links of three link-types *paper*, *conference*, and *keyword* respectively. Two networks `db-t-m` and `theory-t-m` are created from it by filtering selected conferences on the fields of Database system and Theory. A brief statistics of the two networks is presented in Table 6.3.

Division of training and test dataset is done following the procedure described in Appendix A. The parameter d in Fig. A.1 defines a particular split of training and test set. Experiments are performed by varying the value of d to examine the effect of different splits on the longitudinal bias.

6.5 Baseline studies

The prediction performance of the proposed features is compared with two previous studies [42,44]. Lichtenwalter et al. [42] have proposed the HPLP framework for supervised link prediction. The HPLP framework uses non-temporal and uni-relational topological features for link prediction. Along with the traditional link prediction features like CN, JC, AA, PA, they have used degree (*degree*) and volume (*volume*) of the target nodes, and a flow based measure PropFlow (*propflow*) in the HPLP framework. Yang et al. [44] have used temporal as well as multi-relational features for link prediction in bibliographic network. Its feature set can be classified into two. The first class contains *recency* and *activeness* of the target node. Recency gives the time elapsed since a node creates a new paper link, and activeness measures the number of new paper links made in last time-step. We denote these two features as $Yang_{Rec}$ and $Yang_{Act}$ respectively. Features of the second class are measured from the *preferential likelihood* of the target nodes based on five properties: *degree*, CN, JC, AA, and the number of conferences attended. We name these features as $Yang^{deg}_{pref}$, $Yang^{CN}_{pref}$, $Yang^{JC}_{pref}$, $Yang^{AA}_{pref}$ and $Yang^{conf}_{pref}$ respectively. Let us explain $Yang^{deg}_{pref}$ briefly. Each node keeps a vector, *degree preferential vector*. It depicts the historical degree of the neighbors of the node, at their connection time. Considering x and y as the target nodes, probability of x to be connected to y in future is measured by the distance of x's degree value with distribution of the values in degree preferential vector of y, and vice versa. This distance is measured using popular statistical measure *z-score*. Multiplication of these two probability values give $Yang^{deg}_{pref}$. Other features of Yang et al.are also calculated similarly.

6.6 Performance of individual features

Performance of individual features of the TMLP feature set are compared here in terms of their AUC values. Unsupervised scores given by each feature to the positive and negative examples are compared with all possible threshold values to get the performance of the feature. It allows us to inspect and compare the performance of the features

Table 6.4: Unsupervised AUC values of TMLP features.

Sl.		d	db-t-m				theory-t-m			
			target		*other*		*target*		*other*	
			train	test	train	test	train	test	train	test
1	*topo, unwt, static*	$d=1$	0.651 (AA)	0.727 (AA)	None	None	0.689 (AA)	0.713 (AA)	None	None
		$d=2$	0.661 (AA)	0.672 (AA)	None	None	0.715 (AA)	0.689 (AA)	None	None
		$d=3$	0.657 (AA)	0.657 (AA)	None	None	0.653 (AA)	0.702 (AA)	None	None
		$d=4$	0.627 (AA)	0.643 (AA)	None	None	0.674 (AA)	0.693 (AA)	None	None
2	*topo/dyad, wt, static*	$d=1$	0.743 (SCN_f)	0.771 (SAA_f)	0.714 (SAA_k)	0.766 (SAA_c)	0.731 (SAA_n)	0.741 (SAA_f)	0.706 (SRA_c)	0.713 (SRA_k)
		$d=2$	0.725 (SAA_f)	0.739 (SAA_f)	0.719 (SAA_k)	0.729 (SAA_k)	0.745 (SAA_f)	0.725 (SAA_f)	0.718 (SAA_k)	0.696 (SRA_k)
		$d=3$	0.726 (SAA_f)	0.736 (SAA_f)	0.713 (SAA_k)	0.735 (SAA_k)	0.697 (SAA_f)	0.739 (SAA_f)	0.675 (SAA_k)	0.718 (SAA_k)
		$d=4$	0.687 (SAA_f)	0.716 (SAA_f)	0.686 (SAA_k)	0.718 (SAA_k)	0.715 (SAA_f)	0.726 (SAA_f)	0.688 (SAA_c)	0.71 (SAA_c)
3	*topo/dyad, wt, temporal*	$d=1$	0.827 (TCN_n)	0.81 (TCN_n)	0.83 (TCN_k)	0.816 (TAA_c)	0.823 (TCN_n)	0.858 (TCN_f)	0.818 (TCN_k)	0.85 (TCN_k)
		$d=2$	0.82 (TCN_n)	0.814 (TCN_n)	0.806 (TCN_c)	0.81 (TCN_k)	0.838 (TCN_f)	0.83 (TCN_f)	0.837 (TCN_k)	0.829 (TCN_k)
		$d=3$	0.812 (TCN_n)	0.805 (TCN_n)	0.793 (TCN_c)	0.81 (TAA_c)	0.824 (TCN_f)	0.832 (TCN_f)	0.821 (TCN_c)	0.829 (TCN_k)
		$d=4$	0.794 (TCN_n)	0.8 (TCN_n)	0.794 (TCN_c)	0.789 (TCN_c)	0.811 (TCN_n)	0.826 (TCN_f)	0.805 (TCN_c)	0.819 (TCN_k)
4	*topo, unwt, temporal*	$d=1$	0.771 (TPA)	0.741 (TPA)	None	None	0.767 (TCN)	0.806 (TCN)	None	None
		$d=2$	0.759 (TCN)	0.759 (TPA)	None	None	0.779 (TPA)	0.78 (TCN)	None	None
		$d=3$	0.737 (TPA)	0.757 (TPA)	None	None	0.775 (TPA)	0.778 (TCN)	None	None
		$d=4$	0.727 (TPA)	0.736 (TPA)	None	None	0.752 (TPA)	0.766 (TPA)	None	None
5	*dyad, wt, static*	$d=1$	None	None	0.669 ($conf$)	0.694 ($conf$)	None	None	0.643 ($conf$)	0.643 ($conf$)
		$d=2$	None	None	0.629 ($conf$)	0.687 ($conf$)	None	None	0.629 ($conf$)	0.638 ($conf$)
		$d=3$	None	None	0.632 ($conf$)	0.684 ($conf$)	None	None	0.647 ($conf$)	0.65 ($conf$)
		$d=4$	None	None	0.62 ($conf$)	0.658 ($conf$)	None	None	0.643 ($conf$)	0.647 ($conf$)
6	*dyad, wt, temporal*	$d=1$	None	None	0.68 ($Tconf$)	0.676 ($Tconf$)	None	None	0.722 ($Tconf$)	0.697 ($Tconf$)
		$d=2$	None	None	0.684 ($Tconf$)	0.684 ($Tconf$)	None	None	0.716 ($Tconf$)	0.711 ($Tconf$)
		$d=3$	None	None	0.653 ($Tkey$)	0.69 ($Tconf$)	None	None	0.716 ($Tconf$)	0.71 ($Tconf$)
		$d=4$	None	None	0.637 ($Tconf$)	0.685 ($Tconf$)	None	None	0.703 ($Tconf$)	0.715 ($Tconf$)

during train and test interval. Table 6.4 presents AUC values of the best performing features of each subcategory for both the training and test interval over `db-t-m` and `theory-t-m`. It shows that the temporal features outperform their static counterparts by significant amount in terms of AUC score. Temporal features that use dyadic information of paper links (feature type $(topo/dyad, wt, temporal, target)$) perform better than the ones that do not use dyadic information (feature type $(topo, unwt, temporal, target)$). Features relating to the links of type other than target also perform nearly as good as their target paper-type counterparts. Adding temporality to such features improves their performance considerably. Even features that use only dyadic information of *other* link types (feature type $(dyad, wt, static, other)$) also show comparable performance with the baseline methods like CN, AA, etc. Adding temporality improves their performance further (see $Tconf$). Features related to Adamic/Adar dominate among the static topological features (Sl. 1 and 2 of Table 6.2). However, features related to common neighbor outperform others when temporal dynamics of dyadic property is considered (Sl. 3 of Table 6.2). $conf$ and $Tconf$ dominate among the features that use only the dyadic property of other link-types. All the aforementioned trends are visible in both train and test set over both the datasets. Performance of a few features varies with dataset. Such as, TPA clearly dominates in `db-t-m` among the features that correspond to its subcategory. Whereas, in `theory-t-m` there is no clear winner. This may be caused by the difference in average degree and its rate of change over time in the two networks. The two networks do not agree with the effect of $conference$ and *keyword* link-types when Sl. 3 features are considered. This may be caused by the difference in ratio between the number of keywords and the number of conferences in the two networks (see Table 6.3). We find that Jaccard's coefficient and its variants perform worst among the topological measures, which is expected because, JC has an anti-preferential characteristics [42].

In subsequent subsections we present a comprehensive analysis on the effect of time-span considered for training and testing, time and multi-relationality over the TMLP features. Effect of longitudinal bias over the proposed features is also presented. We further discuss the performance shown by the features considered in baseline studies.

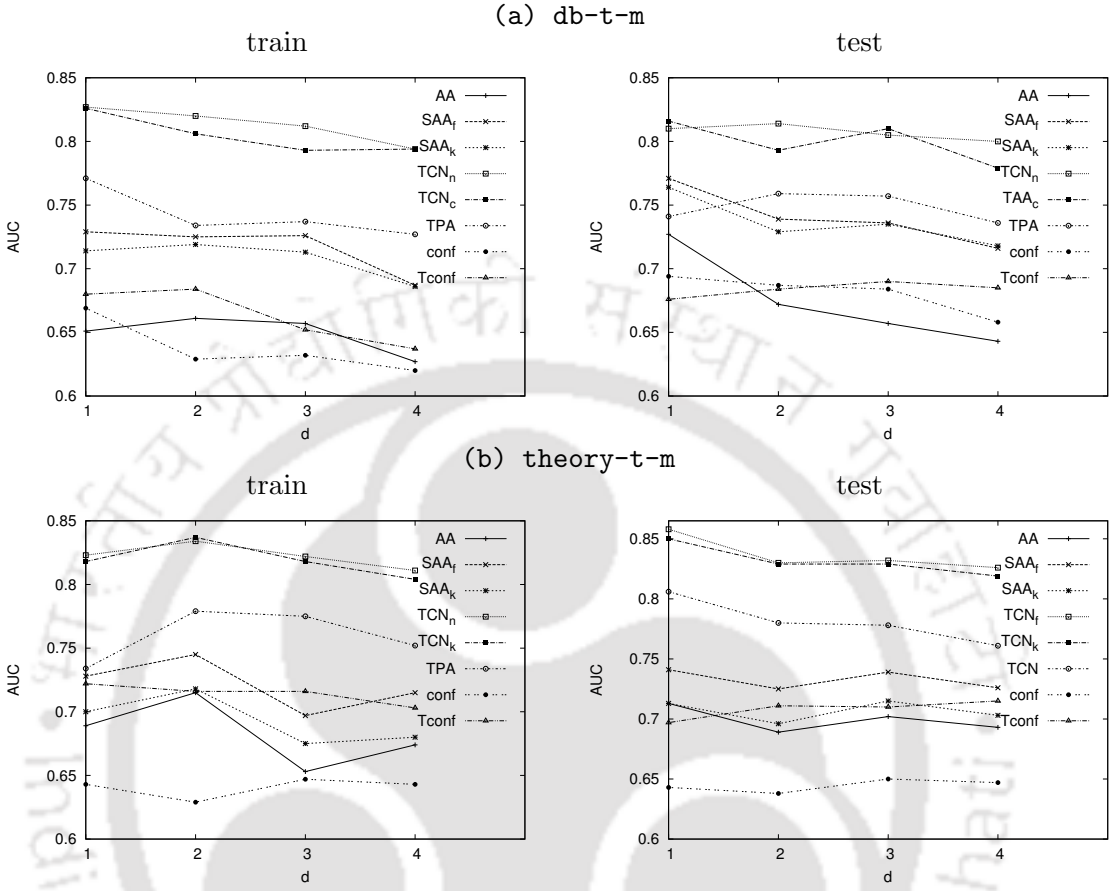


Figure 6.2: Performance of TMLP features for different d values.

6.6.1 Effect of time-span considered for training and testing

In this subsection, we discuss effect of the time-span used to define the training and test set on the TMLP features. In Fig. 6.2, the parameter d defines the span from which class labels are collected (for both training and test). As the value of d increases, time-span for preparing train and test multi-graphs decreases, and subsequently, true future links are collected during larger period of time. Fig. 6.2 shows the effect of d on individual features. It includes the best performing feature from each subcategory for train and test set over the two datasets. It reveals that performance of almost all features (except for

`theory-t-m`, train) degrade when the value of d is increased. This fact is justified because predicting more volume of data by observing less is difficult. This trend is sometimes violated when $d = 1$ (such as, for `theory-t-m`, train). Bibliographic collaboration is a slow process, and collecting future links during one year increases *false positives* affecting the AUC score. For other networks which grow faster, the window-size is usually kept smaller than a year (a month or a week), and similar effect may be observed for lower values of d. The temporal features behave more consistently with the aforementioned trend than the non-temporal ones. It is expected because, temporal methods get less number of windows to analyze when the value of d is high, and therefore, are prone to over-fitting in such case.

6.6.2 Effect of time, multi-relationality

Effectiveness of the temporal features and multi-relational properties in link prediction are demonstrated here. Fig. 6.3 shows that the temporal features outperform their non-temporal counterparts substantially in both of training and test interval over `db-t-m` and `theory-t-m`. The comparison is demonstrated for two subsets of features: the ones that make use of dyadic history, and the features that do not use dyadic history. CN and its variants are chosen to represent the first subset, as it is the mostly affected baseline topological measure in this context. For same reason, PA and its variants are selected as the baseline measure for the second subset. Fig. 6.3 shows that, TCN_n outperforms CN and SCN_n, and TPA outperforms PA significantly in all the occasions. Fig. 6.4 demonstrates how the dyadic properties of the links of other types, i.e., conference and keyword links help in link prediction. We select Adamic/Adar as the baseline topological measure and *keyword* as the link-type to show this effect. The figure shows that even the static weighted measure SAA_k outperforms AA in almost all cases. The temporal method TAA_k performs much better than its static counterparts in all cases. `db-t-m` is affected more than `theory-t-m` by the dyadic history of links of types other than paper.

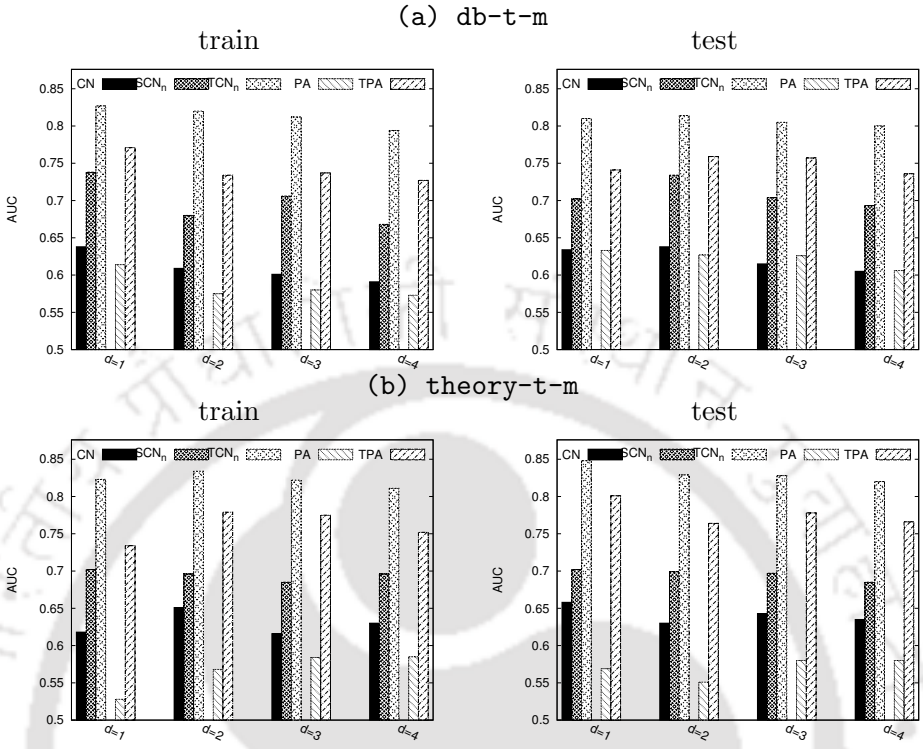


Figure 6.3: Effect of time.

6.6.3 Overcoming longitudinal bias

Performance of features vary from training to test set due to longitudinal bias. Changing characteristics of the underlying network causes the longitudinal bias. However, its effect varies over features. Here we investigate how the performance of TMLP features vary from the training period to the testing period for getting a view on the longitudinal bias created on datasets. Fig. 6.5 shows that the temporal features are much robust than their non-temporal counterparts as far as the longitudinal bias is concerned. Adamic/Adar is chosen as the baseline topological measure, and compares the bias of its static weighted version with the temporally weighted version for the target as well as other link-types. The bar chart draws the absolute percentage change in performance of a measure from training to test. It shows that in almost all cases such changes in performance of TAA_f

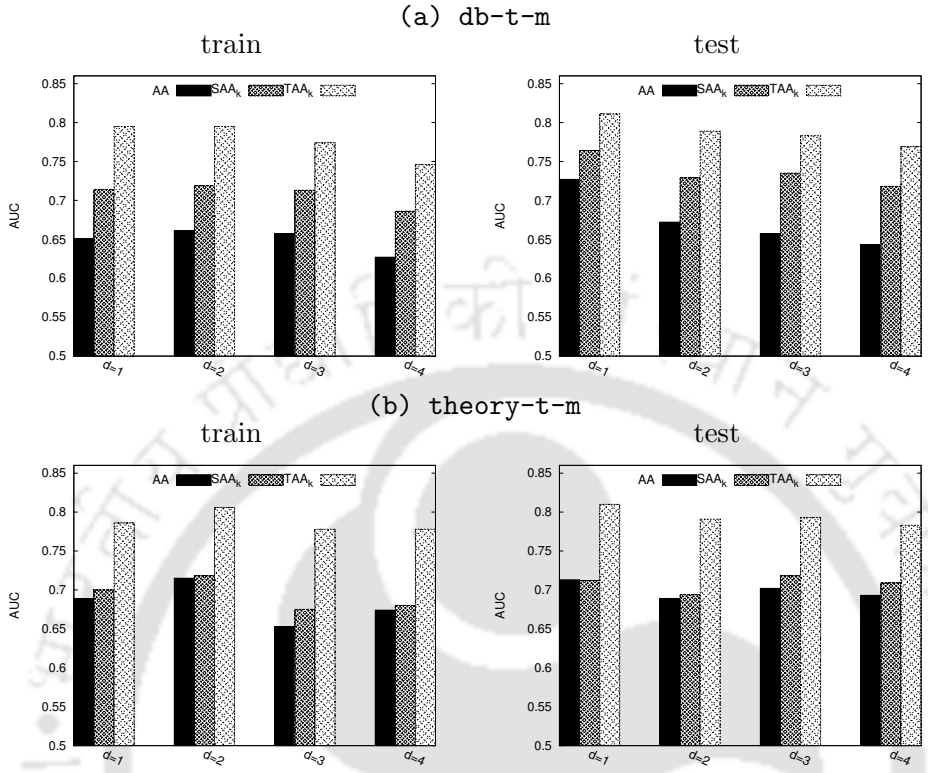


Figure 6.4: Effect of relation-type other than target.

and TAA_k are much less as compared to SAA_f and SAA_k respectively.

6.6.4 Comparing with other baseline features

This subsection compares performance of the baseline features presented in Section 6.5 with that of the proposed TMLP features. Features used in the HPLP framework are the static target link-type features present in the TMLP features. We have already shown that the temporal and other relationship-type features outperform the baseline static ones. The *propflow* feature of the HPLP framework also does not perform well because flow-based measures are not suitable for undirected networks [42]. Simple temporal method *recency* of YANG framework performs well. However, the preferential likelihood based features perform badly compared to the temporal ones proposed in the TMLP. One reason for this

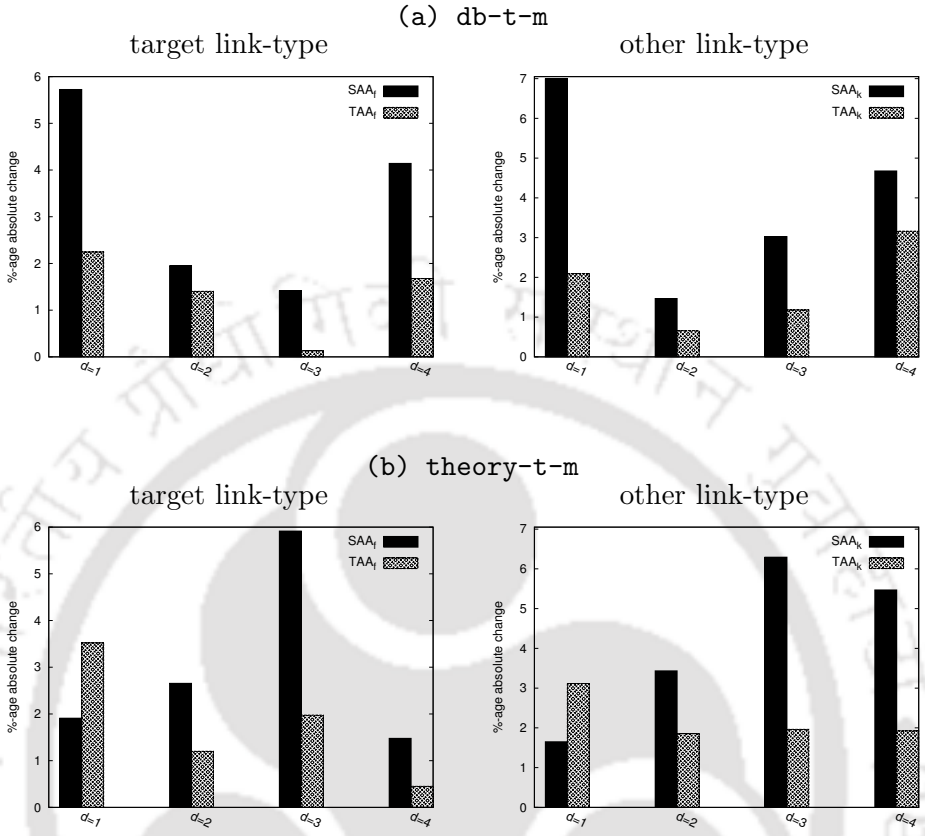


Figure 6.5: Percentage absolute change in performance from training to test.

is that, features require enough temporal history associated with target nodes; otherwise it assigns zero to the preferential likelihood based features. The temporal TMLP features deal with such over-fitting problem by considering *mean*[4] as the forecasting method instead of fitting the exponential smoothing model where the length of the time-series under consideration is small. Another problem with the preferential likelihood based features is, it uses *z-score* (see Subsection 6.5) for calculating the preferential likelihood. In bibliographic networks, number of links decreases rapidly with the number of common conferences attended by the end nodes, at their connection time; and the distribution follows power law (see Fig. 6.6). Selecting this kind of skewed distribution as *null*

[4]*mean* method forecasts the average value of data present in all windows of a time-series.

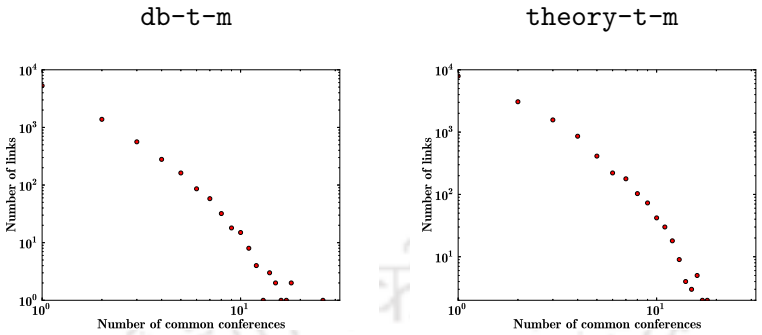


Figure 6.6: Log-log plot of the histogram of the number of common conferences attended by two authors at their connection time.

hypothesis may not be appropriate in this scenario. The values of mean and variance of this distribution is very small compared to the maximum of the independent variable. It penalizes the links for which, number of common conferences is greater than the mean. However, it contradicts with the intuition that probability of collaboration increases with the number of common conferences attended. Similar justifications can be provided for other preferential likelihood based measures.

6.7 Supervised Prediction

Motivated by [42], we use bagging with random forests to perform supervised classification as it works well in class imbalance scenario. It involves two levels of bootstrapping with replacement, which overcomes over-fitting. 15 bootstraps are used for bagging at the topmost level, and each bootstrap is fed to a random forest. In our experiments, each random forest contains 500 decision-trees. The whole experiment is run with 10 different seeds, and the average of their performance in terms of their AUC score is presented here. Comparison of the AUC values of the proposed TMLP model with the HPLP and YANG is presented in Table 6.5. $(-,-)$ in the column TMLP gives the percentage improvement of TMLP over HPLP and YANG respectively. It shows that the proposed model outperforms others significantly in both of the networks for all values of d. As HPLP contains only static and uni-relational features, it performs worst in all cases. Improvement in performance of

Table 6.5: Supervised AUC values for HPLP, YANG and TMLP feature set, and a combination of the three.

Network	d	HPLP	YANG	TMLP	Combined
db-t-m	$d = 1$	0.727	0.822	0.857 (17.9%, 4.3%)	0.863
	$d = 2$	0.746	0.804	0.868 (16.4%, 8.0%)	0.871
	$d = 3$	0.734	0.813	0.869 (18.4%, 6.9%)	0.874
	$d = 4$	0.697	0.795	0.853 (22.4%, 7.3%)	0.856
theory-t-m	$d = 1$	0.717	0.838	0.886 (23.6%, 5.7%)	0.888
	$d = 2$	0.715	0.818	0.873 (22.1%, 6.7%)	0.875
	$d = 3$	0.700	0.828	0.876 (25.1%, 5.8%)	0.878
	$d = 4$	0.690	0.819	0.864 (25.2%, 5.5%)	0.866

TMLP over HPLP increases with the increase in d, because non-temporal features are more vulnerable than the temporal ones against longitudinal bias (see Subsection 6.6.3), and the longitudinal bias increases as the value of d increases. TMLP is able to achieve upto 8% and 6.7% improvement over YANG in `db-t-m` and `theory-t-m` respectively. As AUC score is not always sufficient to evaluate a system, we also provide the corresponding ROC plots in Fig. 6.7. These plots also show that, TMLP can achieve very high true positive rate at a cost of very few false positives. The true positive rate increases rapidly against the false positive rate in the initial stage of the curves corresponding to TMLP, and slows down only when true positive rate reaches a very high value. TMLP achieves much higher true positive rate as compared to HPLP and YANG for lower values of false positive rate. We have also experimented with all the features combining the three models. However, it gives negligible performance improvement over TMLP (see column named Combined in Table 6.5). To demonstrate usefulness of temporality, dyadic history, and multi-relationality independently, we have performed our experiment with different subsets of the TMLP feature set. Some poorly performing features like JC and its variants do not contribute in the supervised learning. Moreover, some features are highly correlated with others. So, we reduce the TMLP feature set by selecting the best performing feature from each subcategory, and name this feature set as $TMLP_S$. We choose the feature subsets

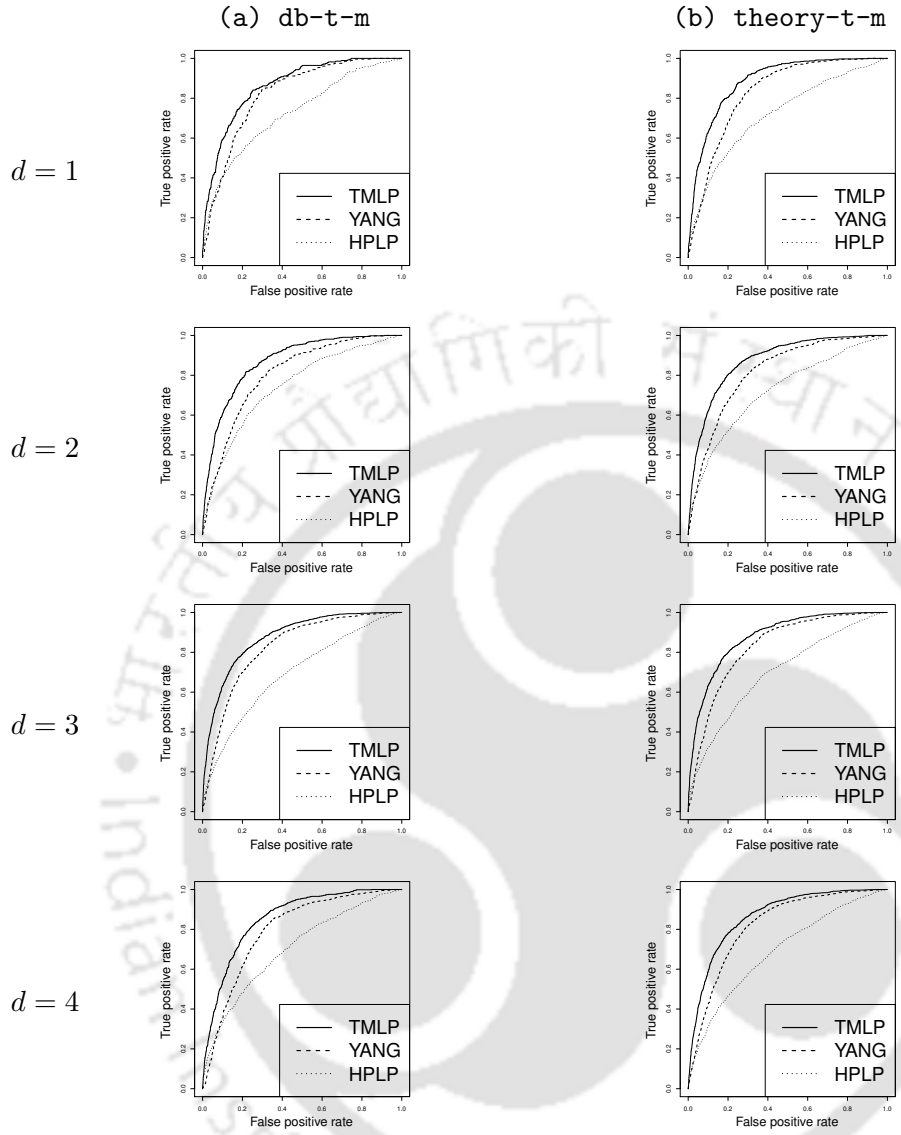


Figure 6.7: ROC curves for HPLP, YANG and TMLP framework.

Table 6.6: List of subsets of TMLP features.

Feature set	AA	SAA_f	SAA_k	SAA_c	TCN_n	TCN_k	TCN_c	TCN	TPA	$conf$	$Tconf$
S_1	✓	✓									
S_2	✓	✓	✓	✓						✓	
S_3	✓							✓	✓		
S_4	✓	✓			✓						
S_5	✓	✓	✓	✓	✓	✓	✓			✓	✓
$TMLP_S$	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 6.7: Comparison of different variants of TMLP.

Network	d	S_1	S_2	S_3	S_4	S_5	$TMLP_S$
db-t-m	$d=1$	0.701	0.759	0.779	0.810	0.848	0.853
	$d=2$	0.711	0.758	0.773	0.820	0.853	0.855
	$d=3$	0.708	0.751	0.782	0.807	0.852	0.862
	$d=4$	0.694	0.730	0.761	0.812	0.837	0.843
theory-t-m	$d=1$	0.709	0.742	0.836	0.833	0.875	0.880
	$d=2$	0.699	0.720	0.816	0.827	0.861	0.867
	$d=3$	0.698	0.728	0.815	0.825	0.865	0.871
	$d=4$	0.694	0.727	0.806	0.809	0.850	0.861

from this reduced feature set. Each subset and the features it considers is tabulated in Table 6.6. S_1 comprises of the subset representing the static and target-link features; S_2 adds static dyadic features of other link-type to S_1; S_3 considers purely topological features (with no dyadic information, which are essentially uni-relational) and adds time component; S_4 adds uni-relational dyadic temporal component to S_1; and S_4 considers all features of $TMLP_S$ except the temporal features which are based on graph topology only. In Table 6.7, we present results with different subsets of the $TMLP_S$ features to demonstrate the usefulness of the multi-relational property as well as the dyadic and topological versions of the temporal dimension. It shows that, introduction of multi-relational features to the static model increases the prediction performance significantly (compare S_1 and S_2). All of the temporal models outperform their static counterparts. The dyadic version of the temporal model (S_4) works much better than its topological counterpart (S_3). Robustness of the temporal features against the longitudinal bias is observed for both of the temporal versions.

6.8 Summary

This chapter proposed the TMLP feature set to predict future links in dynamic bibliographic networks using temporal and multi-relational information. It proposed

new temporal features based on time-series data collected from topological and dyadic history present in links of multiple link-types. To model the time-series data, it used the Exponential smoothing model. A comparative study on unsupervised as well as supervised prediction has been presented to demonstrate the effectiveness of the proposed features over the existing ones. From the supervised learning performance of different variants of the proposed framework, it has been observed that introducing multi-relational and temporal features can overcome the longitudinal bias.

In the next chapter, we propose two aggregation methods for the dyadic time-series, which overcome the inherent sparsity issue in the time-series, and allows us to apply complex time-series forecasting models like ARIMA for modeling the aggregated time-series.

Chapter 7

Addressing sparsity problem in dyadic time-series

7.1 Overview

In previous two chapters, time-series forecasting method has been exploited to enhance link prediction performance. Exponential smoothing model has been chosen to model several time-series extracted from node and link properties of networks. Reason behind the selection of exponential smoothing have been its simplicity and efficiency in estimating model parameter, and an assumption that network dynamics is driven by its recent behavior, which is inherent in exponential smoothing (one extreme case of exponential smoothing is *recency* [44, 51]). On the other hand, researchers have used other forecasting methods to model network dynamics. Auto-regressive models have taken foremost place in this race. Huang and Lin [34] has used auto-regressive integrated moving average (ARIMA) model to predict whether two connected individuals will interact again in future or not. Richard et al. [53] have also adapted auto-regressive models for modeling the evolution of networks. Although different time-series forecasting methods have been used individually towards temporal link-prediction, a thorough understanding of applicability of these methods is absent in literature. As observed in the previous chapters of this thesis, directed and undirected networks respond differently subject to certain network

parameters while predicting future links. Chapter 3 has revealed that frequency of interaction based link weight effects link prediction positively in undirected networks, whereas prediction performance worsens when the same is applied to directed networks. Results presented in Chapter 5 has shown that, effect of temporal preprocessing on link prediction is higher in `dblp` (undirected network) as compared to `fb` (directed network), where the temporal features have been generated using exponential smoothing model. This chapter investigates the effect of different time-series methods in temporal link prediction over networks of diverse characteristics, and propose robust and efficient temporal features for link prediction.

The time-series, which have been building block of the temporal features proposed in the previous two chapters often contains insufficient data to fit with complex forecasting models, and sometimes are very sparse in nature. For an example, in `dblp` there are few co-author pairs who frequently write papers together, or who collaborate for longer time-period. Let us explain the effect of sparseness using a toy example. Let infrequent interactions between two nodes in a social network results in a time series $\langle 1, 0, 0, 0, 0, 0, 0, 0, 1 \rangle$. While modeling this time series with the *simple exponential smoothing model*, the estimated parameter under-fits, and becomes nearly equivalent to the *average* method. It spoils the purpose of modeling with simple exponential smoothing, which is expected to give exponentially higher weightage to the more recent windows than the past ones. Moreover, interactions with the newly added nodes tend to form short length time series, which can not be modeled at all with forecasting models. In Chapter 6, this limitation has been dealt by applying *avge* unsupervised method on short length time-series, instead of exponential smoothing. It might affect prediction performance. On the other hand, there may be similarities in temporal characteristics of all dyads in a network. This assumption encourages us to prepare a single aggregate time-series with sufficient data to represent the dynamics of all dyadic time-series, which is used to estimate the model parameters. These parameters are used to produce the forecast values for all dyadic time-series. Here we propose two such aggregation methods. Advantages of the proposed aggregation methods are as follows.

- It reduces the job of estimating model parameters for every time series to estimating the same for a single time series. Considering low overhead of preparing the aggregate time series, it increases the efficiency of link prediction.

- It reduces the adverse effect of data sparsity in link prediction.

In this chapter, we propose two aggregation methods for the dyadic time series. The aggregate time series are modeled using three time series models: $ARIMA$, $ETS(A, N, N)$ and $ETS(A, A, N)$, and the model parameters are estimated. These parameters are used to forecast every dyadic time series of the network. New link prediction features are proposed by exploiting the forecast values in terms of link weight. Prediction performance of the proposed features (in unsupervised as well as supervised manner) are compared with baseline methods to show that aggregation of dyadic time series enhances prediction performance.

7.2 Aggregation techniques for dyadic time-series

This section proposes two methodologies to find an aggregate time-series as the representative of all dyadic time-series of a network. Let an evolving networks up-to time instant t be represented by an undirected graph $G^t = (V, E, T, \mathcal{Q})$, where V is the set of vertices representing actor nodes; $E \subset V \times V$ is the set of edges representing links; T is the set of time-stamps of interactions occurred in the network till t; $\mathcal{Q}$ is a function $\mathcal{Q} : E \rightarrow 2^T$, which returns the historical time-stamps of all interactions associated with a link. For notational simplicity, we map each edge in E with e_i, where $i \in \mathbb{N} \leq |E|$. Like in Chapter 5, the dyadic time-series $\mathscr{D}(i)$ of link e_i is prepared:

$$\mathscr{D}(i) =< \mathscr{D}_{-q}(i), \mathscr{D}_{-q+1}(i), \mathscr{D}_{-q+2}(i), ..., \mathscr{D}_{-1}(i), \mathscr{D}_0(i) >,$$

for windows $w_{-q}, w_{-q+1}, w_{-q+2}, ..., w_{-1}, w_0$, where $w_{-q} : q \in \mathbb{N}$ represents the window when the link e_i appeared. Two aggregation techniques proposed here are described next.

7.2.1 Singular value decomposition based

We first propose a sampling technique for choosing a fraction of dyadic time series. The chosen time series are represented in a matrix, and then *singular value decomposition* is applied. We describe the sampling technique next.

The sampling technique imposes certain conditions on the time series to qualify. Firstly, to encounter over-fitting, the selected time series should have large span, bounded by n, where w_{-n+1} is the earliest time window in the network. To overcome under-fitting due to sparseness, we impose a secondary condition, which ensures that most of the windows of the chosen time series be populated with non-zero values. So, we first sort all time series in decreasing order of their span, and then select top $m << |E|$ number of time series satisfying the following condition. Let π be the distribution of interaction counts for links (i.e., the sum of the values in respective time series) in the network. The total interaction count of the chosen time series must fall in the largest *decile*, *i.e.* above 90 *percentile* of π. *decile* is preferred to represent the threshold over other statistical averaging methods, because the target distribution is skewed towards the lower values. A matrix $\mathbf{A}$ with dimension $m \times n$ is prepared with the selected time series. The rows corresponding to time series having data points less than n will have missing (or *zero*) value(s) in left column(s). These entries are padded using the earliest valid data point. We present an example matrix of dimension 4×10 as follows. The matrix is formed with the time series of the links $e_1, e_3, e_4,$ and e_9 which have $10, 10, 9,$ *and* 8 number of valid data points respectively.

$$\begin{array}{c} \\ e_1 \\ e_3 \\ e_4 \\ e_9 \end{array} \begin{array}{c} \begin{array}{cccccc} w_{-9} & w_{-8} & w_{-7} & \dots & w_{-1} & w_0 \end{array} \\ \begin{pmatrix} \mathscr{D}_{-9}(1) & \mathscr{D}_{-8}(1) & \mathscr{D}_{-7}(1) & \dots & \mathscr{D}_{-1}(1) & \mathscr{D}_0(1) \\ \mathscr{D}_{-9}(3) & \mathscr{D}_{-8}(3) & \mathscr{D}_{-7}(3) & \dots & \mathscr{D}_{-1}(3) & \mathscr{D}_0(3) \\ \mathscr{D}_{-8}(4) & \mathscr{D}_{-8}(4) & \mathscr{D}_{-7}(4) & \dots & \mathscr{D}_{-1}(4) & \mathscr{D}_0(4) \\ \mathscr{D}_{-7}(9) & \mathscr{D}_{-7}(9) & \mathscr{D}_{-7}(9) & \dots & \mathscr{D}_{-1}(9) & \mathscr{D}_0(9) \end{pmatrix} \end{array}$$

Singular value decomposition (SVD) is applied on $\mathbf{A}$ to get

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathbf{T}},$$

where

- $\mathbf{U}$ is an $m \times n$ orthogonal matrix containing the *left singular vectors* of $\mathbf{A}$ in its columns,
- $\mathbf{\Sigma}$ is an $n \times n$ diagonal matrix containing the *singular values* of $\mathbf{A}$ in decreasing order,
- $\mathbf{V}$ is an $n \times n$ orthogonal matrix containing the *right singular vectors* of $\mathbf{A}$ in its columns.

The leading r (rank of matrix $\mathbf{A}$) right singular vectors represent row space of $\mathbf{A}$, and the first right singular vector retains maximum variance in the row space. We consider this as the aggregate time-series, and denote it as $\hat{\mathscr{D}}_{SVD}$.

7.2.2 Average based

This method averages the total number of interactions in each time-window over all links to get the aggregate time series. We define the aggregate time-series as:

$$\hat{\mathscr{D}}_{avg} =< \frac{\sum_{e_i \in |E_{-n+1}|} \mathscr{D}_{-n+1}(i)}{|E_{-n+1}|}, \frac{\sum_{e_i \in |E_{-n+2}|} \mathscr{D}_{-n+2}(i)}{|E_{-n+2}|}, \ldots, \frac{\sum_{e_i \in |E_0|} \mathscr{D}_0(i)}{|E_0|} >,$$

where E_{-n} gives the set of all edges in $G^{t-\Delta n}$.

7.3 Temporal link weight

The aggregate time-series $\hat{\mathscr{D}}_{SVD}$ and $\hat{\mathscr{D}}_{avg}$ are modeled using several forecast models, and the parameters are estimated. These parameters are used to form the forecast equation for time-series forecasting models. All dyadic time-series of the network are fit with the unique forecast equation (for a particular model), and the forecast values are exploited to assign link weights. The forecasting models used by us to model the time series are given below.

- Auto-regressive integrated moving average ($ARIMA(p, d, q)$) Model

- Exponential smoothing model
  1. Simple exponential smoothing model ($ETS(A, N, N)$)
  2. Holt's linear trend model ($ETS(A, A, N)$)

7.3.1 Forecast value and link weight

In Chapters 5 and 6, we have modeled each of the dyadic time-series with $ETS(A, N, N)$, and estimated the parameter α for each time-series. Instead here we model the aggregate time-series with $ETS(A, N, N)$ along with $ETS(A, A, N)$ and $ARIMA(p, d, q)$ to get unique set of parameters and forecast equation (for each model) as representative for all dyadic time series. Like Chapter 6, the forecast values give the link weight.

7.4 Feature set

Let us take the reference of Table 6.2 of Chapter 6 to make the list of the features proposed here. The features represented by the quadruple $(topo/dyad, wt, temporal, target)$ lists the features generated by the weighted versions of five local proximity based link prediction methods, where the weight of each link is generated by modeling its dyadic time-series using $ETS(A, N, N)$. In this chapter also we exploit those five weighted link prediction methods, but the temporal weights are calculated using aggregate time-series. So, each of the ten features present in the mentioned category of Table 6.2 has six counterparts here, because we have proposed two aggregation methods, and the resultant time-series for each aggregation method is modeled using three forecasting models. We denote each combination of aggregation method and forecasting model as a tuple: $\langle -, - \rangle$, first one of which takes two values SVD and avg representing the aggregation methods, and second one takes values $ETS(A, N, N)$, $ETS(A, A, N)$ and $ARIMA(p, d, q)$ representing the forecasting models. We attach these tuples with the notations of each feature in the mentioned category of Table 6.2 to denote their counterparts depicting the features proposed here. So, the counterparts of TCN_f are denoted as: $TCN_f^{\langle SVD, ETS(A,N,N) \rangle}$, $TCN_f^{\langle SVD, ETS(A,A,N) \rangle}$, $TCN_f^{\langle SVD, ARIMA(p,d,q) \rangle}$,

$TCN_f^{\langle avg,ETS(A,N,N)\rangle}$, $TCN_f^{\langle avg,ETS(A,A,N)\rangle}$, $TCN_f^{\langle avg,ARIMA(p,d,q)\rangle}$. Other 54 features are also denoted in similar fashion.

To compare with baseline, we consider the features represented by the quadruple $(topo/dyad, wt, temporal, target)$ of Table 6.2, Chapter 6. In Chapter 6, we have used to model the dyadic time-series with $ETS(A, N, N)$. Here we consider three models. So, we attach model names with the features of the quadruple $(topo/dyad, wt, temporal, target)$ of Table 6.2 to denote the baseline features used here. The counterparts of TCN_f are denoted as: $TCN_f^{\langle ETS(A,N,N)\rangle}$, $TCN_f^{\langle ETS(A,A,N)\rangle}$, $TCN_f^{\langle ARIMA(p,d,q)\rangle}$. Note that, $TCN_f^{\langle ETS(A,N,N)\rangle}$ is equivalent to TCN_f of Chapter 6. Other 27 features representing baseline are also denoted in similar fashion.

7.5 Datasets

We use four datasets in our experiments: `db-t, theory-t, fb-t and enron-t`. `db-t and theory-t` datasets are the uni-relational version of `db-t-m and theory-t-m` datasets used in Chapter 6. `fb-t` dataset is the same one used in Chapter 5. `enron-t` is temporal version of the `enron` dataset used in Chapters 3 and Chapter 4. These four datasets cover a wide range of social networks like bibliographic network (`db-t and theory-t`), on-line social network (`fb-t`) and e-mail network (`enron-t`).

7.6 Experimental results

The division of training and testing span for the datasets follows the procedure described in Appendix A. For `db-t` and `theory-t`, we choose $d = 3$ in our experiment, because of its consistency observed in prediction performance in Chapter 6. Like in Chapter 4, for `enron-t` and `fb-t`, we choose $d = 4$, and $d = 3$ respectively.

7.6.1 Performance of individual features

This subsection demonstrates the performance of individual features proposed in Section 7.4 in terms of their AUC scores. Table 7.1 provides a comparison of the pro-

Table 7.1: Comparison of the AUC values of proposed features with baseline.

		$ETS(A,N,N)$			$ETS(A,A,N)$			$ARIMA(p,d,q)$		
		baseline	*avg*	*SVD*	*baseline*	*avg*	*SVD*	*baseline*	*avg*	*SVD*
enron-t	*CN*	0.783	**0.852** (8.81%)	0.8508 (8.66%)	0.7834	0.846 (7.99%)	0.8507 (8.59%)	0.8333	0.8488 (1.86%)	0.8493 (1.92%)
	JC	0.6087	0.6916 (13.62%)	0.7217 (18.56%)	0.7299	0.7418 (1.63%)	0.7478 (2.45%)	0.7362	0.7533 (2.32%)	**0.7554** (2.61%)
	AA	0.8415	0.8616 (2.39%)	0.8628 (2.53%)	0.786	0.8521 (8.41%)	0.8591 (9.3%)	0.8149	**0.8633** (5.94%)	0.8623 (5.82%)
	RA	0.8445	0.8646 (2.38%)	0.8646 (2.38%)	0.7923	0.8638 (9.02%)	0.864 (9.05%)	0.8304	**0.8648** (4.14%)	**0.8648** (4.14%)
	PA	0.8487	0.866 (2.04%)	**0.8759** (3.2%)	0.7447	0.8642 (16.05%)	0.8753 (17.54%)	0.8241	0.8661 (5.1%)	0.8726 (5.89%)
fb-t	*CN*	0.6484	0.665 (2.56%)	**0.7147** (10.23%)	0.6034	0.6331 (4.92%)	0.6486 (7.49%)	0.6495	0.6941 (6.87%)	0.6946 (6.94%)
	JC	0.6206	**0.6751** (8.78%)	0.6505 (4.82%)	0.6113	0.6343 (3.76%)	0.6382 (4.4%)	0.6315	0.6405 (1.43%)	0.6414 (1.57%)
	AA	0.7163	0.7241 (1.09%)	**0.728** (1.63%)	0.6283	0.6772 (7.78%)	0.6857 (9.14%)	0.6583	0.7241 (10%)	0.7247 (10.09%)
	RA	0.7223	0.7279 (0.78%)	0.7241 (0.25%)	0.6251	0.6815 (9.02%)	0.6916 (10.64%)	0.6576	0.7303 (11.06%)	**0.7305** (11.09%)
db-t	*CN*	0.8049	0.8089 (0.5%)	0.8091 (0.52%)	0.7473	**0.8198** (9.7%)	0.8197 (9.69%)	0.7502	0.8124 (8.29%)	0.8117 (8.2%)
	JC	0.5946	0.7397 (24.4%)	**0.7435** (25.04%)	0.6271	0.6513 (3.86%)	0.6856 (9.33%)	0.6945	0.7128 (2.63%)	0.7079 (1.93%)
	AA	0.7982	0.8228 (3.08%)	0.8233 (3.14%)	0.7482	0.8254 (10.32%)	**0.8262** (10.43%)	0.7543	0.8081 (7.13%)	0.8174 (8.37%)
	RA	0.7469	0.7969 (6.69%)	**0.7975** (6.77%)	0.7326	0.7787 (6.29%)	0.774 (5.65%)	0.7412	0.7819 (5.49%)	0.7822 (5.53%)
	PA	0.8044	**0.8352** (3.83%)	0.8351 (3.82%)	0.6897	0.8184 (18.66%)	0.8248 (19.59%)	0.7898	0.8112 (2.71%)	0.8119 (2.8%)
theory-t	*CN*	0.832	0.8348 (0.34%)	**0.8436** (1.39%)	0.7806	0.8412 (7.76%)	0.8421 (7.88%)	0.7894	0.8363 (5.94%)	0.8408 (6.51%)
	JC	0.6768	0.768 (13.48%)	**0.7686** (13.56%)	0.7426	0.7658 (3.12%)	0.7623 (2.65%)	0.7371	0.7562 (2.59%)	0.7556 (2.51%)
	AA	0.8252	0.8267 (0.18%)	0.8276 (0.29%)	0.774	0.8344 (7.8%)	**0.8356** (7.96%)	0.786	0.8325 (5.92%)	0.8274 (5.27%)
	RA	0.771	0.8047 (4.37%)	**0.806** (4.54%)	0.7547	0.7978 (5.71%)	0.8002 (6.03%)	0.7714	0.7984 (3.5%)	0.7988 (3.55%)
	PA	0.7855	0.8283 (5.45%)	**0.8287** (5.5%)	0.6737	0.827 (22.75%)	0.8179 (21.4%)	0.802	0.7963 (-0.71%)	0.802 (0%)

posed features with the baselines. Each row summarizes the performance of the proposed features corresponding to a particular link prediction method and dataset. The columns with heading *baseline* stand for the baseline features, which model individual dyadic time-series. For an example, the row corresponding to *CN* link prediction method for `enron-t` dataset gives the performance of the features $TCN_f^{\langle ETS(A,N,N)\rangle}$, $TCN_f^{\langle avg,ETS(A,N,N)\rangle}$, $TCN_f^{\langle SVD,ETS(A,N,N)\rangle}$, $TCN_f^{\langle ETS(A,A,N)\rangle}$, $TCN_f^{\langle avg,ETS(A,A,N)\rangle}$, $TCN_f^{\langle SVD,ETS(A,A,N)\rangle}$, $TCN_f^{\langle ARIMA(p,d,q)\rangle}$, $TCN_f^{\langle avg,ARIMA(p,d,q)\rangle}$, $TCN_f^{\langle SVD,ARIMA(p,d,q)\rangle}$ respectively in `enron-t`. For the bibliographic datasets `db-t` and `theory-t`, each of such features has another counterpart that uses newman method rather than frequency to populate the time-series. In such cases we present results for the best one among the two. In the columns with headings *avg* and *SVD*, the values inside parenthesis indicates the improvement over baselines. The bold-faced values indicate the best performing feature in each row.

Some of the observations from the table are summarized below.

Table 7.2: Performance of supervised models in terms of AUC values

Network	$B.ARIMA(p,d,q)$	$P.ARIMA(p,d,q)$	$B.ETS(A,N,N)$	$P.ETS(A,N,N)$	$B.ETS(A,A,N)$	$P.ETS(A,A,N)$	$best.perform$
enron-t	0.8603	0.8737	0.8569	**0.8788**	0.8558	0.8772	0.8995
fb-t	0.7444	**0.7576**	0.7548	0.7561	0.7441	0.7496	0.7856
db-t	0.7884	0.8389	0.8325	**0.8664**	0.7986	0.8621	0.8697
theory-t	0.8130	**0.8675**	0.8432	0.8670	0.7810	0.8645	0.8724

- The proposed features outperform the baselines in almost all cases.
- Features which exploit $ETS(A,N,N)$ model, dominate the others in all datasets, except enron-t. In enron-t, features exploiting $ARIMA(p,d,q)$ dominates. However, the best performing feature for enron-t: $TPA_f^{\langle SVD,ETS(A,N,N)\rangle}$, also exploits $ETS(A,N,N)$ model.
- Features which use SVD aggregation method performs better than their avg counterparts on an average. Superior performance of SVD may be justified as: it applies two phase noise reduction (sampling and applying singular value decomposition), which reduces over-fitting in the process of aggregation.
- Best performing features for each of the datasets are: $TPA_f^{\langle SVD,ETS(A,N,N)\rangle}$ in enron, $TRA_f^{\langle SVD,ARIMA(p,d,q)\rangle}$ in fb-t, $TPA_n^{\langle avg,ETS(A,N,N)\rangle}$ in db-t, and $TCN_n^{\langle SVD,ETS(A,N,N)\rangle}$ in db-t.

7.6.2 Supervised prediction

We combine the features to build several supervised models towards supervised prediction. We use the supervised framework as described in Appendix A. We summarize the supervised models used in our experiments below.

- $B.ARIMA(p,d,q)$: Combination of baseline features, where $ARIMA(p,d,q)$ model is used to model the dyadic time-series plus the static features.
- $B.ETS(A,N,N)$: Combination of baseline features, where $ETS(A,N,N)$ model is used to model the dyadic time-series plus the static features.

- $B.ETS(A, A, N)$: Combination of baseline features, where $ETS(A, A, N)$ model is used to model the dyadic time-series plus the static features.
- $P.ARIMA(p, d, q)$: Combination of proposed features, where $ARIMA(p, d, q)$ model is used to model the dyadic time-series plus the static features.
- $P.ETS(A, N, N)$: Combination of proposed features, where $ETS(A, N, N)$ model is used to model the dyadic time-series plus the static features.
- $P.ETS(A, A, N)$: Combination of proposed features, where $ETS(A, A, N)$ model is used to model the dyadic time-series plus the static features.
- $best.perform$: Combination of the best performing features of each prediction method, i.e., each row of Table 7.1.

Table 7.2 summarizes the performance of the proposed models. It demonstrates that the models which are built using combination of the proposed features outperform their baseline counterparts in all the cases. The AUC values of best performing model (other than $best.perform$) for datasets are made bold-faced. The result shows that, difference of performance among the models which combines proposed features are less. However, the improvement in performance in the models which combines proposed features over their baseline counterparts is substantial. This observation shows that by using aggregation method for dyadic time-series the link prediction performance is improved.

7.7 Summary

This chapter has proposed two aggregation methods for the dyadic time-series to encounter data sparsity. The aggregated time-series acts as the representative dyadic time-series for the whole network. These aggregated time-series have been modeled using several forecasting models. Using the unique parameter set obtained by modeling each forecasting model, link-weights have been quantified. Several features have been proposed with the new link -weighting schemes. Unsupervised performance of the proposed features in four datasets has shown that they outperform their baseline counterparts in significant amount.

Supervised learning experiments have also been performed by combining multiple features. The experimental results have shown that the aggregation methods successfully handle data sparsity problem in dyadic time-series.

Chapter 8

Conclusion and future directions

8.1 Conclusion

In this thesis we exploited dyadic and structural similarity of node-pairs in several ways to predict future links in social networks. We encoded dyadic similarity in link weights, and used traditional proximity based link prediction methods as the building block of structural similarity. We proposed several new link prediction methods, which embed inherent properties of social networks, such as: reciprocative nature of relationships, temporal dynamics, heterogeneity, etc., in dyadic and structural similarity.

In Chapter 3, we first empirically investigated (using ten real datasets) the effect of traditional frequency of interaction based link weight on weighted link prediction methods, and found that the effect depends on characteristics of the network. The observations indicated that prediction methods are positively effected by to link weight in undirected networks, whereas directed networks are effected negatively. We also performed degree odd ratio analysis. It revealed that when the target node pairs' neighborhoods are populated by strong links, weighted link prediction methods perform better than their unweighted counterparts, for directed network also. We further proposed an index to speculate whether in a network weighted prediction methods may outperform their unweighted counterpart or not.

Chapter 4 effectively made link weight useful for link prediction in directed networks.

It first demonstrated empirical evidence of the importance of reciprocal links in triangle closing, using a null model. Then reciprocity aware link-weighting technique was proposed for directed networks, which successfully made the weighted prediction methods respond positively. It also proposed new link prediction methods for directed networks, and showed that they can be effective when combined with other prediction methods in supervised learning.

We gradually moved towards temporal link prediction in Chapter 5. To model temporal dynamics of social network (dyadic as well as topological), we exploited simple exponential smoothing time-series forecasting model. We first devised a time-aware method to predict dull nodes and links in an early stage, and cleaned them to enhance link prediction performance. Due to unavailability of ground truth information for the dull nodes and links in the datasets, we proposed a novel method to label he dull nodes and links.

In Chapter 6 we combined temporal dynamics of dyadic interaction (as link weights) and topology in multi-relational networks, and proposed several methods for link prediction. Like in Chapter 5, simple exponential smoothing time-series forecasting model was used to model time-series. Proposed methods were considered as features, and combined to build several models towards supervised prediction. We performed experiments with real datasets, and showed that our framework outperforms the methods proposed in recent studies. We also showed that the proposed temporal methods reduce longitudinal bias.

We further proposed two aggregation methods of dyadic time-series to encounter data sparsity problem in Chapter 7. We modeled the aggregated time-series using several time-series forecasting methods to estimate the parameters. Aggregated time-series acted as the representative time-series for all links, and all dyadic time-series were forecast using the same parameters of the aggregated time-series. Several experiments were performed on real datasets to show that aggregation improves prediction performance.

8.2 Future directions

Most of the earlier studies on link prediction consider homogeneous networks. However many of the real world networks are heterogeneous in nature (multiple types of nodes and multiple types of relationships between nodes). Few recent studies in complex network analysis have considered heterogeneous networks [43–48]. In Chapter 6, we have also considered heterogeneous networks by exploiting multi-relational information, and observed that it boosts prediction performance. Our method exploits multi-relational information from single source, such as DBLP dataset. However, multi-relational information may also be collected from multiple sources, such as, the authors' activities in OSNs like Facebook, twitter, the authors' website, etc. Recently researchers have started working on integration of network data [129] available in multiple sources. Cross-platform multi-relational information may better link prediction performance, which is important to explore.

Existing studies on evolution of social network [20, 28] usually deal with addition of new nodes and links in the network, and ignore disappearance of nodes and links with time. In chapter 5, we have considered disappearance of nodes and links in network evolution. We referred them as dull. We have observed improvement in link prediction performance by cleaning potential dull nodes and links from the network. In absence of ground truth about disappearing nodes and links, we considered the nodes and links which are inactive for long as dull. In reality, nodes and links disappear when people change their social circles, relocates, changes job, die, etc. Moreover, social ties are also of various nature. So, change in other behavioral characteristics of nodes and links such as (i) nodes' community, (ii) location, occupation, etc., (iii) type of link, etc., may also be investigated in this regard, and analyzing their effect in link prediction may be an important future extension.

Appendix A

Supervised Framework

This chapter discusses the general supervised framework adopted in this thesis for link prediction task. Supervised learning is used for classifying data points or samples into several categories. It first trains a statistical model from labeled samples, and then use that model to predict the class labels for unknown samples. Each sample is described by several features, also known as variables. Each feature can take a certain range of values, which are used to discriminate between samples. A sample is represented by a vector, called as feature vector, whose dimensions are given by each of the features. Therefore, each sample can be thought of as a unique vector in n-dimensional vector space where n is the number of features considered. In supervised learning approach, initially the class labels of some samples are known beforehand. They are called training examples. These training examples, i.e., their feature vectors are used to learn a model that can predict the class of a unknown sample summarized by its feature vector. Logistic regression, support vector machine (SVM), decision tree are some examples of supervised models widely used in data mining. Link prediction is a binary classification problem that classifies a node-pair into one of the two categories: *node-pairs that will be connected in future*, and *node-pairs that will never form any link*. The unsupervised methods like the ones presented in Section 2.1 can be used as features.

A.1 Preparing training and test set from longitudinal data

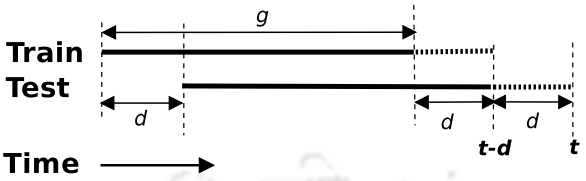


Figure A.1: Schematic view of training and test set preparation.

Dynamic networks evolve over time. To train a supervised link prediction framework, a time instant needs to be fixed upto which network data should be considered to train the model. Collection of labeled examples, i.e. future links, also depends on time. The whole training time-span is divided into two parts, where the first part is used for preparing the graph, and the second part is used for collecting the new links that appear between node pairs present in the graph. This kind of framework, where the specifications of data points are dependent on time is known as *longitudinal framework*. Testing these kind of framework is a bit tricky. A schematic view of dividing the training and test dataset is shown in Figure A.1. Solid lines (of time-span g) represent the time-line that is used for preparing the graph, and the dashed line is used to generate the class labels. First, the classifier is trained with network data upto time $t-d$, where the train graph is constructed with network data during $t-g-2d$ to $t-2d$, and the true future links (i.e., true class labels) are collected during next d amount of time. Given the trained classifier at time $t-d$, it is evaluated with the labels collected from next d amount of time. The time-spans during which training and test intervals for constructing graphs, and collecting class labels, are kept equal to reduce longitudinal bias [42]. Both of training and testing class labels are collected from same span of time d due to same reason. Longitudinal bias is a type of bias incurred on the classifier in supervised longitudinal framework due to change in trends during time. It affects prediction accuracy, because the trained model may not exactly match with the required model to test, due to shift in time. Longitudinal bias is usually reduced by considering either same time-span or same volume of data for training and

testing. It is not possible to maintain both the criteria, because the real networks under consideration do not evolve linearly (in terms of number of nodes and links) with time. Moreover, it is also not possible to maintain same volume of data for training and testing, because available dataset often does not contain time-stamps with granularity required for serving the purpose. Thus, in this thesis, we maintain the same time-span criterion to reduce longitudinal bias. We use the parameter d to define a particular split of training and test set.

Unsupervised link prediction has no training phase. Thus, dividing the time-span into two parts is sufficient. Any of the two divisions: *Train* and *Test* in Figure A.1 may be considered for testing the performance of such methods.

A.2 Bagging with random forests

Link prediction suffers from class imbalance problem as the number of future links is much lesser than the links that never appear. Bagging is a widely used method in data mining that deals with the class imbalance problem, and gets rid of over-fitting by reducing variance. Bagging derives multiple sample subsets from the training sample set using *bootstrapping with replacement*. Each of the sample subsets is used to train a supervised model. While testing, each of the trained model is used for classification, and the class which gets maximum vote is chosen. Week classifiers such as decision tree gets maximum benefit through bagging. Lichtenwalter et al. [42] have shown that bagging with random forests performs best among other contemporary supervised frameworks for link prediction. In this framework, the training examples are bagged to get several sample subsets, and then each of such sample subsets is used to learn a random forest. Random forests themselves are bagged decision trees, where a random combination of feature is considered to take decision in each node of a decision tree.

Appendix B

Link Prediction Evaluation Metric

Precision, recall, AUC score are some of the evaluation metrics used in link prediction task. However, considering the inherent class imbalanced problem present in link prediction, AUC is widely considered as better evaluation metric. This section briefly discusses about AUC score.

Given a test sample, a classification model for link prediction usually returns a value that serves as the sample's likelihood (or score) of being a future link. The samples, whose scores are above a threshold, are predicted as future links. Given a threshold, the test samples are grouped into four:

- **True positive (TP):** The samples which are correctly predicted as future links.
- **False positive (FP):** The samples which are incorrectly predicted as future links.
- **True negative (TN):** The samples which are correctly not predicted as future links.
- **False negative (TN):** The samples which are incorrectly not predicted as future links.

True positive rate (TPR) and false positive rate (FPR) are defined as:

$$TPR = \frac{\text{number of correctly predicted future links}}{\text{number of actual future links}} = \frac{|TP|}{|TP| + |FN|}, and$$

$$FPR = \frac{\text{number of incorrectly predicted future links}}{\text{number of actual negative links}} = \frac{|FP|}{|FP| + |TN|}.$$

Receiver operating characteristic curve (ROC curve) plots TPR vs FPR for all possible range of thresholds. AUC (area under ROC curve) score gives the area under this curve. Its value varies in $[0, 1]$. AUC score is equivalent to the probability of a randomly chosen positive example, i.e., actual future links having higher prediction score than a randomly chosen negative example. A pure random prediction model gets the AUC value of 0.5, and a perfect model with no prediction error will achieve the AUC value 1.0. To evaluate link prediction, AUC score is calculated as follows [56]: Let p be the number of positive examples; n be the number of negative examples; c_i and d_i denote the number of negative examples that have less and equal score respectively, compared to positive example i. Then,

$$AUC = \frac{\sum_i c_i + 0.5 \times \sum_i d_i}{n * p}.$$

References

[1] M. E. Newman, "The structure and function of complex networks," *SIAM review*, vol. 45, no. 2, pp. 167–256, 2003.

[2] D. J. Watts and S. H. Strogatz, "Collective dynamics of small-worldnetworks," *nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[3] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *science*, vol. 286, no. 5439, pp. 509–512, 1999.

[4] M. E. Newman, "Modularity and community structure in networks," *Proceedings of the national academy of sciences*, vol. 103, no. 23, pp. 8577–8582, 2006.

[5] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, "Network motifs: simple building blocks of complex networks," *Science*, vol. 298, no. 5594, pp. 824–827, 2002.

[6] A. S. Klovdahl, "Social networks and the spread of infectious diseases: the aids example," *Social science & medicine*, vol. 21, no. 11, pp. 1203–1216, 1985.

[7] E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai, and A.-L. Barabási, "Hierarchical organization of modularity in metabolic networks," *science*, vol. 297, no. 5586, pp. 1551–1555, 2002.

[8] O. L. Petchey, P. T. McPhearson, T. M. Casey, and P. J. Morin, "Environmental warming alters food-web structure and ecosystem function," *Nature*, vol. 402, no. 6757, pp. 69–72, 1999.

[9] C. Andersson, K. Frenken, and A. Hellervik, "A complex network approach to urban growth," *Environment and Planning A*, vol. 38, no. 10, pp. 1941–1964, 2006.

[10] S. Arianos, E. Bompard, A. Carbone, and F. Xue, "Power grid vulnerability: A complex network approach," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 19, no. 1, p. 013119, 2009.

[11] P.-A. Chirita, J. Diederich, and W. Nejdl, "Mailrank: Using ranking for spam detection," in *CIKM '05*. ACM, 2005, pp. 373–380.

[12] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," in *CIKM '03*, 2003, pp. 556–559.

[13] J. A. Barnes, *Class and committees in a Norwegian island parish*. Plenum New York, 1954.

[14] W. W. Zachary, "An information flow model for conflict and fission in small groups," *Journal of anthropological research*, pp. 452–473, 1977.

[15] R. L. Breiger and P. E. Pattison, "Cumulated social roles: The duality of persons and their algebras," *Social networks*, vol. 8, no. 3, pp. 215–256, 1986.

[16] M. E. J. Newman, "Scientific collaboration networks. i. network construction and fundamental results," *Phys. Rev. E*, vol. 64, p. 016131, Jun 2001.

[17] N. P. Hummon and P. Dereian, "Connectivity in a citation network: The development of dna theory," *Social networks*, vol. 11, no. 1, pp. 39–63, 1989.

[18] A. V. Papachristos, "Murder by structure: Dominance relations and the social structure of gang homicide1," *American Journal of Sociology*, vol. 115, no. 1, pp. 74–128, 2009.

[19] M. Granovetter, "The Strength of Weak Ties," *The American Journal of Sociology*, vol. 78, no. 6, pp. 1360–1380, 1973.

[20] A.-L. Barabâsi, H. Jeong, Z. Néda, E. Ravasz, A. Schubert, and T. Vicsek, "Evolution of the social network of scientific collaborations," *Physica A: Statistical mechanics and its applications*, vol. 311, no. 3, pp. 590–614, 2002.

[21] A. Guille, H. Hacid, C. Favre, and D. A. Zighed, "Information diffusion in online social networks: A survey," *SIGMOD Rec.*, vol. 42, no. 2, pp. 17–28, Jul. 2013.

[22] D. Cai, Z. Shao, X. He, X. Yan, and J. Han, "Mining hidden community in heterogeneous social networks," in *WLD '05*. ACM, 2005, pp. 58–65.

[23] L. C. Freeman, "Centrality in social networks conceptual clarification," *Social networks*, vol. 1, no. 3, pp. 215–239, 1978.

[24] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *SIGKDD '03*. ACM, 2003, pp. 137–146.

[25] S. Fortunato, "Community detection in graphs," *Physics reports*, vol. 486, no. 3, pp. 75–174, 2010.

[26] M. Al Hasan and M. J. Zaki, "A survey of link prediction in social networks," in *Social network data analytics*. Springer, 2011, pp. 243–275.

[27] M. Kimura and K. Saito, "Tractable models for information diffusion in social networks," in *ECML/PKDD '06*. Springer, 2006, pp. 259–271.

[28] G. Kossinets and D. J. Watts, "Empirical analysis of an evolving social network," *Science*, vol. 311, no. 5757, pp. 88–90, 2006.

[29] J. Leskovec, L. Backstrom, R. Kumar, and A. Tomkins, "Microscopic evolution of social networks," in *SIGKDD '08*. ACM, 2008, pp. 462–470.

[30] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan, "Group formation in large social networks: membership, growth, and evolution," in *SIGKDD '06*. ACM, 2006, pp. 44–54.

[31] L. M. Aiello, A. Barrat, R. Schifanella, C. Cattuto, B. Markines, and F. Menczer, "Friendship prediction and homophily in social media," *ACM Trans. Web*, vol. 6, no. 2, pp. 9:1–9:33, Jun. 2012.

[32] G. Salton, *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley Longman Publishing Co., Inc., 1989.

[33] V. E. Krebs, "Mapping networks of terrorist cells," *Connections*, vol. 24, no. 3, pp. 43–52, 2002.

[34] Z. Huang and D. K. J. Lin, "The time-series link prediction problem with applications in communication surveillance," *INFORMS Journal on Computing*, vol. 21, no. 2, pp. 286–303, 2009.

[35] B. Barzel and A.-L. Barabási, "Network link prediction by global silencing of indirect correlations," *Nature biotechnology*, vol. 31, no. 8, pp. 720–725, 2013.

[36] Z. Huang, X. Li, and H. Chen, "Link prediction approach to collaborative filtering," in *JCDL '05*. ACM, 2005, pp. 141–142.

[37] M. McPherson, L. Smith-Lovin, and J. M. Cook, "Birds of a feather: Homophily in social networks," *Annual review of sociology*, pp. 415–444, 2001.

[38] T. Zhou, L. Lü, and Y.-C. Zhang, "Predicting missing links via local information," *The European Physical Journal B - Condensed Matter and Complex Systems*, vol. 71, no. 4, pp. 623–630, 2009.

[39] M. E. J. Newman, "Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality," *Physical Review E*, vol. 64, p. 016132, 2001.

[40] T. Murata and S. Moriyasu, "Link prediction of social networks based on weighted proximity measures," in *WI '07*. IEEE Computer Society, 2007, pp. 85–88.

[41] L. Lü and T. Zhou, "Link prediction in weighted networks: The role of weak ties," *EPL (Europhysics Letters)*, vol. 89, p. 18001, 2010.

[42] R. N. Lichtenwalter, J. T. Lussier, and N. V. Chawla, "New perspectives and methods in link prediction," in *SIGKDD '12*. ACM, 2010, pp. 243–252.

[43] Y. Sun, J. Han, C. C. Aggarwal, and N. V. Chawla, "When will it happen?: relationship prediction in heterogeneous information networks," in *WSDM '12*. ACM, 2012, pp. 663–672.

[44] Y. Yang, N. V. Chawla, Y. Sun, and J. Han, "Predicting links in multi-relational and heterogeneous networks." in *ICDM '12*, vol. 12, 2012, pp. 755–764.

[45] Y. Dong, J. Tang, S. Wu, J. Tian, N. V. Chawla, J. Rao, and H. Cao, "Link prediction and recommendation across heterogeneous social networks," in *ICDM '12*. IEEE, 2012, pp. 181–190.

[46] D. Davis, R. Lichtenwalter, and N. V. Chawla, "Multi-relational link prediction in heterogeneous information networks," in *ASONAM '11*. IEEE, 2011, pp. 281–288.

[47] Y. Sun, R. Barber, M. Gupta, C. C. Aggarwal, and J. Han, "Co-author relationship prediction in heterogeneous bibliographic networks," in *ASONAM '11*. IEEE, 2011, pp. 121–128.

[48] B. Ermi, E. Acar, and A. Cemgil, "Link prediction in heterogeneous data via generalized coupled tensor factorization," *Data Mining and Knowledge Discovery*, vol. 29, no. 1, pp. 203–236, 2015.

[49] L. Lü and T. Zhou, "Link prediction in complex networks: A survey," *Physica A: Statistical Mechanics and its Applications*, vol. 390, no. 6, pp. 1150–1170, 2011.

[50] T. Tylenda, R. Angelova, and S. Bedathur, "Towards time-aware link prediction in evolving social networks," in *SNA-KDD '09*. ACM, 2009, pp. 9:1–9:10.

[51] A. Potgieter, K. A. April, R. J. Cooke, and I. O. Osunmakinde, "Temporality in link prediction: Understanding social complexity," *Emergence: Complexity & Organization (E: CO)*, vol. 11, no. 1, pp. 69–83, 2009.

[52] D. M. Dunlavy, T. G. Kolda, and E. Acar, "Temporal link prediction using matrix and tensor factorizations," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 5, no. 2, p. 10, 2011.

[53] E. Richard, S. Gaïffas, and N. Vayatis, "Link prediction in graphs with autoregressive features," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 565–593, Jan. 2014.

[54] R. Hyndman, A. Koehler, J. Ord, and R. Snyder, *Forecasting with Exponential Smoothing: The State Space Approach*, ser. Springer Series in Statistics. Springer, 2008.

[55] M. J.-H. Lim, M. Negnevitsky, and J. Hartnett, "Artificial intelligence applications for analysis of e-mail communication activities," 2004.

[56] L. Lü and T. Zhou, "Link prediction in complex networks: A survey," *Physica A: Statistical Mechanics and its Applications*, vol. 390, no. 6, pp. 1150–1170, 2011.

[57] Z. Liu, Q.-M. Zhang, L. Lü, and T. Zhou, "Link prediction in complex networks: A local naïve bayes model," *EPL (Europhysics Letters)*, vol. 96, no. 4, p. 48007, 2011.

[58] P. Symeonidis, E. Tiakas, and Y. Manolopoulos, "Transitive node similarity for link prediction in social networks with positive and negative links," in *RecSys '10*. ACM, 2010, pp. 183–190.

[59] S. Scellato, A. Noulas, and C. Mascolo, "Exploiting place features in link prediction on location-based social networks," in *SIGKDD '11*. ACM, 2011, pp. 1046–1054.

[60] V. Leroy, B. B. Cambazoglu, and F. Bonchi, "Cold start link prediction," in *SIGKDD '10*. ACM, 2010, pp. 393–402.

[61] R. N. Lichtenwalter, J. T. Lussier, and N. V. Chawla, "New perspectives and methods in link prediction," in *SIGKDD '10*. ACM, 2010, pp. 243–252.

[62] P. Jaccard, "Distribution de la flore alpine dans le bassin des Dranses et dans quelques régions voisines," *Bulletin de la Société Vaudoise des Sciences Naturelles*, vol. 37, pp. 241–272, 1901.

[63] L. A. Adamic and E. Adar, "Friends and neighbors on the web," *Social Networks*, vol. 25, no. 3, pp. 211–230, 2003.

[64] M. T. Gastner and M. E. Newman, "The spatial structure of networks," *The European Physical Journal B-Condensed Matter and Complex Systems*, vol. 49, no. 2, pp. 247–252, 2006.

[65] M. E. Newman, "Clustering and preferential attachment in growing networks," *Physical Review E*, vol. 64, no. 2, p. 025102, 2001.

[66] L. Katz, "A new status index derived from sociometric analysis," *Psychometrika*, vol. 18, no. 1, pp. 39–43, 1953.

[67] S. Brin and L. Page, "Reprint of: The anatomy of a large-scale hypertextual web search engine," *Computer networks*, vol. 56, no. 18, pp. 3825–3833, 2012.

[68] G. Jeh and J. Widom, "Simrank: a measure of structural-context similarity," in *SIGKDD '02*. ACM, 2002, pp. 538–543.

[69] A. Clauset, C. Moore, and M. E. Newman, "Hierarchical structure and the prediction of missing links in networks," *Nature*, vol. 453, no. 7191, pp. 98–101, 2008.

[70] R. Guimerà and M. Sales-Pardo, "Missing and spurious interactions and the reconstruction of complex networks," *Proceedings of the National Academy of Sciences*, vol. 106, no. 52, pp. 22 073–22 078, 2009.

[71] P. W. Holland, K. B. Laskey, and S. Leinhardt, "Stochastic blockmodels: First steps," *Social networks*, vol. 5, no. 2, pp. 109–137, 1983.

[72] N. Friedman, L. Getoor, D. Koller, and A. Pfeffer, "Learning probabilistic relational models," in *IJCAI '99*, vol. 99, 1999, pp. 1300–1309.

[73] D. Heckerman, C. Meek, and D. Koller, "Probabilistic entity-relationship models, prms, and plate models," *Introduction to statistical relational learning*, pp. 201–238, 2007.

[74] K. Yu, W. Chu, S. Yu, V. Tresp, and Z. Xu, "Stochastic relational models for discriminative link prediction," in *NIPS '06*, 2006, pp. 1553–1560.

[75] K. Miller, M. I. Jordan, and T. L. Griffiths, "Nonparametric latent feature models for link prediction," in *ANIPS '09*, 2009, pp. 1276–1284.

[76] M. Zhou, "Infinite edge partition models for overlapping community detection and link prediction." in *AISTATS '15*, 2015.

[77] P. Sarkar, D. Chakrabarti, and M. Jordan, "Nonparametric link prediction in dynamic networks," in *ICML '12*, 2012.

[78] J. Kunegis and A. Lommatzsch, "Learning spectral graph transformations for link prediction," in *ICML '09*. ACM, 2009, pp. 561–568.

[79] J. Kunegis, E. W. De Luca, and S. Albayrak, "The link prediction problem in bipartite networks," in *International Conference on Information Processing and Management of Uncertainty in Knowledge-based Systems*. Springer, 2010, pp. 380–389.

[80] J. Kunegis, S. Schmidt, A. Lommatzsch, J. Lerner, E. W. De Luca, and S. Albayrak, "Spectral analysis of signed graphs for clustering, prediction and visualization." in *SDM '10*, vol. 10. SIAM, 2010, pp. 559–559.

[81] K. M. Borgwardt, "Graph kernels," Ph.D. dissertation, LMU, 2007.

[82] Y. Saad, *Numerical methods for large eigenvalue problems*. SIAM, 1992, vol. 158.

[83] M. A. Hasan, V. Chaoji, S. Salem, and M. Zaki, "Link prediction using supervised learning," in *SDM '06*, 2006.

[84] D. Wang, D. Pedreschi, C. Song, F. Giannotti, and A.-L. Barabasi, "Human mobility, social ties, and link prediction," in *SIGKDD '11*. ACM, 2011, pp. 1100–1108.

[85] N. Benchettara, R. Kanawati, and C. Rouveirol, "Supervised machine learning applied to link prediction in bipartite social networks," in *ASONAM '10*. IEEE, 2010, pp. 326–330.

[86] H. R. De Sá and R. B. Prudêncio, "Supervised link prediction in weighted networks," in *IJCNN '11*. IEEE, 2011, pp. 2281–2288.

[87] C. Ahmed, A. ElKorany, and R. Bahgat, "A supervised learning approach to link prediction in twitter," *Social Network Analysis and Mining*, vol. 6, no. 1, pp. 1–11, 2016.

[88] J. G. De Gooijer and R. J. Hyndman, "25 years of time series forecasting," *International journal of forecasting*, vol. 22, no. 3, pp. 443–473, 2006.

[89] G. E. Box, G. M. Jenkins, and G. C. Reinsel, *Time series analysis: forecasting and control*. John Wiley & Sons, 2013.

[90] D. Peña, G. C. Tiao, and R. S. Tsay, *A course in time series analysis*. John Wiley & Sons, 2011, vol. 322.

[91] C. Wang, V. Satuluri, and S. Parthasarathy, "Local probabilistic models for link prediction," in *ICDM '07*. IEEE, 2007, pp. 322–331.

[92] J.-P. Onnela, J. Saramäki, J. Hyvönen, G. Szabó, M. A. De Menezes, K. Kaski, A.-L. Barabási, and J. Kertész, "Analysis of a large-scale weighted network of one-to-one human communication," *New Journal of Physics*, vol. 9, no. 6, p. 179, 2007.

[93] J.-P. Onnela, J. Saramäki, J. Hyvönen, G. Szabó, D. Lazer, K. Kaski, J. Kertész, and A.-L. Barabási, "Structure and tie strengths in mobile communication networks," *Proceedings of the National Academy of Sciences*, vol. 104, no. 18, pp. 7332–7336, 2007.

[94] H. Ogata, Y. Yano, N. Furugori, and Q. Jin, "Computer supported social networking for augmenting cooperation," *Computer Supported Cooperative Work (CSCW)*, vol. 10, no. 2, pp. 189–209, 2001.

[95] E. Gilbert and K. Karahalios, "Predicting tie strength with social media," in *SIGCHI '09*. ACM, 2009, pp. 211–220.

[96] I. Kahanda and J. Neville, "Using transactional information to predict link strength in online social networks." *ICWSM*, vol. 9, pp. 74–81, 2009.

[97] C. Scholz, M. Atzmueller, and G. Stumme, "On the predictability of human contacts: Influence factors and the strength of stronger ties," in *SOCIALCOM-PASSAT '12*. IEEE Computer Society, 2012, pp. 312–321.

[98] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Phys. Rev. E*, vol. 74, p. 036104, Sep 2006.

[99] T. Opsahl and P. Panzarasa, "Clustering in weighted networks," *Social networks*, vol. 31, no. 2, pp. 155–163, 2009.

[100] T. Herman, M. Monsalve, S. Pemmaraju, P. Polgreen, A. M. Segre, D. Sharma, and G. Thomas, "Inferring realistic intra-hospital contact networks using link prediction and computer logins," in *SOCIALCOM-PASSAT '12*. IEEE Computer Society, 2012, pp. 572–578.

[101] M. McPherson, L. Smith-Lovin, and J. M. Cook, "Birds of a feather: Homophily in social networks," *Annual review of sociology*, pp. 415–444, 2001.

[102] D. J. Watts and S. H. Strogatz, "Collective dynamics of small-worldnetworks," *nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[103] A. Barrat, M. Barthelemy, R. Pastor-Satorras, and A. Vespignani, "The architecture of complex weighted networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 11, pp. 3747–3752, 2004.

[104] J. Saramäki, M. Kivelä, J.-P. Onnela, K. Kaski, and J. Kertesz, "Generalizations of the clustering coefficient to weighted complex networks," *Physical Review E*, vol. 75, no. 2, p. 027105, 2007.

[105] R. N. Lichtenwalter, *Network analysis and link prediction: effective and meaningful modeling and evaluation*. University of Notre Dame, 2012.

[106] D. M. Romero and J. M. Kleinberg, "The directed closure process in hybrid social-information networks, with an analysis of link formation on twitter." in *ICWSM '10*, 2010.

[107] T. Lou, J. Tang, J. Hopcroft, Z. Fang, and X. Ding, "Learning to predict reciprocity and triadic closure in social networks," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 7, no. 2, p. 5, 2013.

[108] V. Zlatić and H. Štefančić, "Model of wikipedia growth based on information exchange via reciprocal arcs," *EPL (Europhysics Letters)*, vol. 93, no. 5, p. 58005, 2011.

[109] R. Kumar, J. Novak, and A. Tomkins, "Structure and evolution of online social networks," in *Link mining: models, algorithms, and applications*. Springer, 2010, pp. 337–357.

[110] N. Sett, S. R. Singh, and S. Nandi, "Influence of edge weight on node proximity based link prediction methods: An empirical analysis," *Neurocomputing*, vol. 172, pp. 71–83, 2016.

[111] B. Viswanath, A. Mislove, M. Cha, and K. P. Gummadi, "On the evolution of user interaction in facebook," in *WOSN '09*. ACM, 2009, pp. 37–42.

[112] M. E. Newman, S. Forrest, and J. Balthrop, "Email networks and the spread of computer viruses," *Physical Review E*, vol. 66, no. 3, p. 035101, 2002.

[113] D. Chakrabarti, R. Kumar, and A. Tomkins, "Evolutionary clustering," in *SIGKDD '06*. ACM, 2006, pp. 554–560.

[114] Y. Chi, X. Song, D. Zhou, K. Hino, and B. L. Tseng, "On evolutionary spectral clustering," *ACM Trans. Knowl. Discov. Data*, vol. 3, no. 4, pp. 17:1–17:30, dec 2009.

[115] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[116] J. Zhang, Y. Wang, and J. Vassileva, "Socconnect: A personalized social network aggregator and recommender," *Information Processing & Management*, vol. 49, no. 3, pp. 721–737, 2013.

[117] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida, "Detecting spammers on twitter," in *CEAS*, vol. 6, 2010, p. 12.

[118] S. Bhagat, G. Cormode, B. Krishnamurthy, and D. Srivastava, "Prediction promotes privacy in dynamic social networks," in *OSN '10*, 2010, pp. 6–6.

[119] A. A. Ferreira, M. A. Gonçalves, and A. H. Laender, "A brief survey of automatic methods for author name disambiguation," *Acm Sigmod Record*, vol. 41, no. 2, pp. 15–26, 2012.

[120] M. Huisman and C. Steglich, "Treatment of non-response in longitudinal network studies," *Social networks*, vol. 30, no. 4, pp. 297–308, 2008.

[121] C. Hernández and G. Navarro, "Compressed representations for web and social graphs," *Knowledge and information systems*, vol. 40, no. 2, pp. 279–313, 2014.

[122] S. A. Macskassy, "Mining dynamic networks: The importance of pre-processing on downstream analytics," *COMMPER 2012*, p. 2, 2011.

[123] K. Y. Kamath and J. Caverlee, "Transient crowd discovery on the real-time social web," in *WSDM '11*. ACM, 2011, pp. 585–594.

[124] T. Raeder, O. Lizardo, D. Hachen, and N. V. Chawla, "Predictors of short-term decay of cell phone contacts in a large scale communication network," *Social Networks*, vol. 33, no. 4, pp. 245–257, 2011.

[125] G. Miritello, "Predicting tie creation and decay," in *Temporal Patterns of Communication in Social Networks*. Springer, 2013, pp. 85–106.

[126] M. Karsai, K. Kaski, A.-L. Barabási, and J. Kertész, "Universal features of correlated bursty behaviour," *Scientific reports*, vol. 2, 2012.

[127] J. A. Hanley and B. J. Mcneil, "The meaning and use of the area under a receiver operating characteristic (ROC) curve." *Radiology*, vol. 143, no. 1, pp. 29–36, april 1982.

[128] R. N. Lichtenwalter and N. V. Chawla, "Vertex collocation profiles: subgraph counting for link analysis and prediction," in *WWW '12*. ACM, 2012, pp. 1019–1028.

[129] G.-J. Qi, C. C. Aggarwal, and T. S. Huang, "Breaking the barrier to transferring link information across networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 7, pp. 1741–1753, 2015.

Brief Biography of the Author

Niladri Sett was born in Sripur, Balagarh, West Bengal, India on 21 December, 1983. After completing his schooling in Balagarh and Chandannagar, he has completed the Bachelor of Engineering (BE) degree from the *Department of Information Technology, National Institute of Technology (NIT), Durgapur*, India in the year 2005. After graduating from NIT Durgapur, he has briefly served as a lecturer in the Department of Information Technology, BCET, Durgapur. He completed his M.Tech. degree from the *Department of Computer Science and Engineering, NIT, Durgapur* in 2009, and thereafter joined *Department of Computer Science and Engineering, Indian Institute of Technology (IIT) Guwahati* as a PhD research scholar. During his PhD, he has been supervised by Dr. Sanasam Ranbir Singh and Prof. Sukumar Nandi. His research interests include social network analysis, information retrieval, data mining, etc. He has also worked in other fields, such as delay tolerant networks, software defined network, network security, etc., in cooperation with other research groups at IIT Guwahati.

Publications related to thesis

Journals

Published

1. Niladri Sett, Sanasam Ranbir Singh and Sukumar Nandi, "Influence of edge weight on node proximity based link prediction methods: an empirical analysis", Neurocomputing 172: 71-83 (2016), Elsevier

Conditionally accepted

2. Niladri Sett, Saptarshi Basu, Sukumar Nandi and Sanasam Ranbir Singh, "Temporal Link prediction in Multi-relational Network", World Wide Web Journal, Springer

3. Niladri Sett, Devesh, Sanasam Ranbir Singh and Sukumar Nandi, "Exploiting reciprocity towards link prediction", Knowledge and Information Systems Journal, Springer

To be submitted

4. Extended version of the paper published in WI 2016 proceedings, Web Intelligence Journal, IOS Press (invited)

Conferences

Published

1. Niladri Sett, Subhrendu Chattopadhyay, Sanasam Ranbir Singh and Sukumar Nandi, "A time aware method for predicting dull nodes and links in evolving networks for data

cleaning", in IEEE/WIC/ACM WI 2016

Under review

2. Niladri Sett, Devesh, Sanasam Ranbir Singh and Sukumar Nandi, "Addressing data sparsity in time series link prediction", in ACM SIGIR 2017 (short paper)

Publications outside thesis

Conferences

Published / Accepted

1. Subhrendu Chattopadhyay, Niladri sett, Sukumar Nandi and Sandip Chakraborty, "Flipper: fault-tolerant distributed network management and control", in IFIP/IEEE IM 2017 (Accepted)

2. Akash Anil, Niladri Sett and Sanasam Ranbir Singh, "Modeling evolution of a social network using temporal graph kernels", in ACM SIGIR 2014 (short paper)

3. Deepak Mangal, Niladri Sett, Sanasam Ranbir Singh and Sukumar Nandi, "Link prediction on evolving social network using spectral analysis", in IEEE ANTS 2013

Manuscript under preparation

4. "Exploiting seasonality in social based DTN routing", with Akhil GV, Amrita Bose Paul, Sukumar Nandi and Santosh Biswas

Department of Computer Science and Engineering

Indian Institute of Technology Guwahati

Guwahati 781039, India