Time-aware methods for Link Prediction in Social Networks

Lankeshwara Munasinghe

Department of Informatics
School of Multidisciplinary Sciences
The Graduate University for Advanced Studies (SOKENDAI)
2013

A dissertation submitted to the Department of Informatics,
School of Multidisciplinary Sciences,
The Graduate University for Advanced Studies(SOKENDAI)
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Informatics

Advisory committee

Associate Professor Ryutaro Ichise (SOKENDAI)
Professor Ken Satoh (SOKENDAI)
Associate Professor Asanobu Kitamoto (SOKENDAI)
Professor Akiko Aizawa (National Institute of Informatics)
Associate Professor Satoshi Oyama (Hokkaido University)

Dedication

To my family

Acknowledgment

I would like to express my sincere gratitude to my supervisor Professor Ryutaro Ichise for his excellent support and supervision extended to me throughout my study, research activity and life in Japan. My special thanks goes to other professors of my advisory panel for their invaluable feedback and constructive comments to improve and enhance my research. I should acknowledge all the staff members of The Graduate University for Advanced Studies and the National Institute of Informatics. I thank all my lab members for their encouragements and sharing all the hard times with me. I would like to express my gratitude to Dr. Chammika Mannakara for his support extended to me throughout my stay in Japan. I thank to people of Japan and the government for providing me a golden opportunity to come here and do my PhD.

I thank my family for their support extended to me with love and care by staying with me all the hard times. I pay my gratitude to my parents, brothers, the only sister and my teachers. Finally, I acknowledge all of my friends in Japan who are always with me, giving all the support and encouragements.

Abstract

Online social network services has become one of the most influential and key source of service providing, information/knowledge sharing and many other Internet based activities. The rapid growth of social networks shows the increasing popularity of these services among the users. The growth of social networks occurs as a result of adding new users and new links between users. The emergence of new links has primacy in the study of social network evolution. Thus, predicting/recommending future links in social networks has attracted a great deal of attention. Link prediction has many applications and, it offers many benefits to the users of social networking services such as providing fast and accurate recommendations or suggestions to the users. However, highly structured massive real-world networks involving heterogeneous entities with complex associations have added new challenges to link prediction research due to different factors such as sparsity, complexity, size, time-dependent nature of the networks.

There have been numerous attempts to address the problem of link prediction through diverse approaches. Most common way is to measure the closeness/similarity of nodes to each other in terms of different social aspects. These similarities change over time due to highly dynamic behavior of social networks. The existing static similarity measures have not been able to cope with rapidly evolving social networks thus, are not sufficient for accurate link prediction. In order to alleviate this problem, we contributed by introducing two novel time-aware features, 1) *Time score* which is capable of dealing with temporality of common neighbors and, 2) *T_Flow* computes information flow between nodes by considering link activeness which vary over time. We used the latest timestamps of interactions/links to compute them. The novel features used in conjunction with supervised machine learning method for link prediction. Both methods tested on real world social networks namely, *facebook* friendship network data and coauthorship data extracted from *ePrint archives*. The results revealed a significant improvement in link prediction accuracy for both features comparing with the existing features.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Summary: *This chapter briefly discuss about social networks and link prediction. Section 1.1 presents the general definition of social networks and their aspects. Link prediction is an inseparable part of social networks. Section 1.2 discuss an overview of link prediction in social networks and the benefit of link prediction. In the following sections we presented our motivation behind this research work.*

1.1 Social networks

A social network* is a structure consist of entities which can be individuals, groups or organizations, and the relations or associations among them. With the emergence of the Internet, the online social networks have been gained increasing popularity. Online social networks has become one of the most influential and key source of service providing, information/knowledge sharing and many other Internet based activities. Social networks are composed of users (nodes) and associations (edges) among them. The users can be individuals, groups, organizations, etc. Users join a social network, publish their own content, profile and create links to other users in the network by making "friendships". The lexical meaning of a "friendship" depends on the network. It can be a ordinary friendship, scientific collaboration, business relationship, etc. The growth of social networks occurs as

* In this thesis we used "network" and "graph" to refer the same entity while "vertices" and "nodes" also the same. The terms "links" and "edges" interchangeably used to refer the associations or relationships between nodes.

a result of adding new users and adding new links.

Social networks serve a range of benefits to its users:

Support for organizing & sharing contents to make friendships: Most social networking services provide platforms for users to create, share and organize their own profiles. These services has become extremely popular due to availability of user oriented, enhanced methods to interact with other users. Social networking sites such as facebook[†] (over 1 billion users), Twitter[‡] (over 200 million users), are examples of wildly popular networks used to share and organize the contents, finding friends. Social networks such as Flickr[§], YouTube[¶], are examples for social networks for sharing multimedia content such as photos, videos.

Support for sharing knowledge, learning & collaboration: Social networks enhance informal learning and support social connections within users or organizations for sharing their profiles for academic and business purposes. The users of such service can find the suitable candidates who match the personal or organizational interests. LinkedIn[‖], a social network made up over 200 million professionals is an example for academic as well as business oriented social networks.

Support for communication: E-mail networks are an example of communication social networks. The modern e-mail system has integrated state-of-art communication technologies such as dialing, chatting, video conferencing in order to empower the users. It has permitted complex heterogeneous social connections between users.

Modern multi-relational, heterogeneous social networks has been analyzed using different approaches such as graph theory, graph mining [98]. Social Network Analysis (SNA) is the study of relations between individuals including the analysis of social structures, social position, role analysis, and many others. Normally, the relationship between individuals, e.g., kinship, friends, neighbors, etc. are presented as a network. Traditional social science involves the circulation of questionnaires, asking respondents to detail their interaction with others. Then a network can be constructed based on the response, with nodes representing individuals and edges the interaction between them. This type of data collection confines traditional SNA to a limited scale, typically at most hundreds of actors in one

[†] www.facebook.com [‡] www.twitter.com [§] www.flickr.com [¶] www.youtube.com
[‖] www.linkedin.com

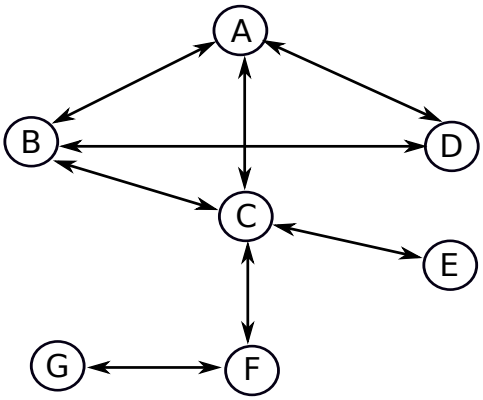


Figure 1.1: Example of a social network depicting the structure of links between nodes A through F. It has links both bi-directional and one-directional

study. With the prosperity of Internet, many social networking and social media sites are emerging, and people can easily connect to each other in the cyber space. This also facilitates SNA to a much larger scale — millions of users or even more in a network; Examples include email communication networks, instant messenger networks, mobile call networks, friendship networks. Other forms of complex network, like coauthorship or citation networks, biological networks, metabolic pathways, genetic regulatory networks, food web and neural networks, are also examine and demonstrate similar patterns. These large scale networks of various entities yield patterns that are normally not observed in small networks. In addition, they also pose computational challenges as well as new tasks and problems for the SNA.

An example of a social network has shown in Figure 1.1. Node A to F denote users and edges represent the relationships between them. Nodes and edges are associated with attributes such as age, gender, time of link creation, etc. In this Figure, edges have direction, bi-directional and one-directional, but node attributes are not shown. Extreme popularity and rapid growth of these online social networks has opened a unique opportunity to study and understand the dynamics of the evolution of such networks. On the other hand, the availability social network data and the analytical methods developed on them has made it easy and interesting to do research on social networks. This ability of data has also broadened the variety of disciplines contributing to the advance of social network research.

Social network analysis has different perspectives such as sociological, business, theoretical, etc. The research in this thesis mainly focused the evolution of social networks. Past research have been extensively studied the growth, shrink and other characteristics of modern social networks [16]. Although those research have done tremendous effort to develop methods to understand the dynamics of the social networks, identifying the mechanisms by which they evolve is a fundamental question that is still not well understood, and it forms the motivation for our work here. We specially investigated the time-related mechanisms in order to predict the future potential links of a given network. To best of our knowledge, correlations between link evolution and temporal behavior of nodes/links still largely open for research.

1.2 Link prediction

Link prediction is the most fundamental problem that attempts to infer which new links are likely to occur in the near future based on the topological,node and edge properties in a given network [52]. That is, if we are presented with a snapshot of a network at the current time, the goal is predicting links that will occur in the next time step. As part of the recent surge of research on large, complex social networks and their properties, a considerable amount of attention has been devoted to the computational analysis of social network evolution. In social networks nodes represent people or other entities embedded in a social context, and whose edges represent interaction, collaboration, or influence between entities.

Link prediction problem has interpreted and defined in many ways. We discuss few of them briefly. A detailed discussion of related work presented in Chapter 2. In data mining perspective, link prediction problem as a link mining task because many real-world networks composed of variety of entity types linked via multiple types of relations. An emerging challenge for link mining is the problem of mining richly linked datasets to explore the knowledge behind the links or relationships. This knowledge provide additional advantage that can be helpful for many data mining tasks. Yet multi-relational data violates the traditional assumption of independent, identically distributed data instances that provides the basis for many statistical machine learning algorithms. Therefore, new approaches are needed

that can exploit the dependencies across the attribute and link structure [27]. Link prediction can be divided in to two cases: (1) predicting entirely new links which means those links are never appeared in the network. New links emerge in between existing nodes as well as by adding new nodes. Predicting links added by latter case is extremely hard problem. Thus, most of the research has been attempting to find methods to predict links among the existing nodes. (2) predicting repeating links, that is, some links are not visible in the network during the observed period of time but they appeared either before or after the observed prod of time [102]. However, if time is a part of the predictive model, then repeating link prediction refer to the same task which is to predict the evolution of a network in terms of new edges that will be added in the future. According to the probabilistic perspective, Link prediction is an estimate of the likelihood or probability of the future occurrence of a link in a network or estimating the probability of whole network taking a particular form by adding set of new links. In both cases the complex dependencies among the links are required to address using probabilistic and statistical models [25].

Past research have been introduced lots of algorithms and methods to solve the long-standing problem of link prediction. Those worthy research have proved that link prediction has many applications and, it offers many benefits to the users of social networking services. Individual users of these services can find their friends, colleagues, or people whom they wish to meet efficiently and accurately [26]. For example, online social networking services such as facebook, linkedin could use link prediction to provide fast, accurate service and precise recommendations or suggestions to their users. Organizations such as security agencies and business organizations can find more accurate information regarding unseen relationships among people or organizations and operate accordingly. Link prediction in scientific collaboration networks has been a fundamental research area. Many researchers have addressed this problem with different approaches because of its utmost importance for the development of research. The effective systems enable researchers to find experts, other individuals in the same research field and research organizations in a more productive manner [84, 88]. The evolution of biological networks such as gene networks, protein-protein networks also have studied as link prediction task [40, 39]. This has been a great privilege to researchers

who are working in field of bioinformatics because biological networks are not easy to observe and understand the microscopic and macroscopic properties of evolution. Once the properties are revealed it can be used to predict the missing links, which is referred as the *network completion problem* [42]. In network completion problem knowledge extracted from an observed part of a network is use to estimate the unseen parts of the network. Link evolution and group or cluster formation are correlated. They can't be considered as independent mechanisms. Therefore, combined approaches of graph clustering and link prediction methods have been used for community detection and proved their consistent accuracy and effectiveness over the graph clustering methods itself [99]. Predicting links among the documents such as research publications, web documents, are also a part of link prediction research. Although the domain is different from the traditional social network the methods and technology used for link prediction are similar. Some research have devised models to predict the document connectivity using semantic information of the documents. Link prediction approaches has been used in other domains such as email anomaly detection, collaborative filtering, and health-care. Application of link prediction methods have shown highly productive results in those domains.

Highly structured massive real-world networks involving heterogeneous entities with complex associations have added new challenges to link prediction research [25]. Besides that, the dynamic behavior of social networks has added an immense challenge to the ink prediction research. Thus, we set our goal to make use of the knowledge extracted from dynamic/temporal behavior of networks to formalize methods which leads to accurate link predictions.

1.3 Motivation

Vast majority of past link prediction research have been used static features or attributes of nodes, links and topological features to predict the future links. Only few research have been considered the temporal behavior of nodes and links. Although most of the static features provide a worthy knowledge about general social phenomenon which could use to predict the potential links, temporal features have huge impact on link evolution. The static features assume that they never

changes over time which is not true every time. It is worthwhile to study how to use the knowledge gained from temporal behavior of nodes and links to predict future potential links. Hence we set the goal of this thesis to study and understand temporal behavior of node and links and how it can be used to predict future link evolution. To this end, we investigated the factors which make nodes and links to become strong/active periodically. Stronger/active links and nodes have greater influence over link evolution than weaker links and nodes. We found that temporality can be caused by various factors depending on the nature of the network. We focused on finding factors which are common across most networks. Our studies revealed that timestamps of links or interactions provide the essential knowledge of temporal behavior. Further, timestamps of links/interactions are useful to study the temporality in most online social networks. With this back ground knowledge we focused on finding new methods, which incorporate the temporality using timestamps, for predicting future links in social networks. To our knowledge, this scenario has not been discussed sufficiently in the context of link prediction. The main contribution of this piece of study is determining the impact of the relationship between the time stamps of the interactions and the link strength for future links. To this end, we introduce two time-aware features which are significantly improved the link prediction accuracy in rapidly changing social networks.

1.4 Organization of the thesis

The remainder of this thesis is organized as follows.

Chapter 2: Related work discuss the past research related to the research presented in this thesis. The discussion begins with the introduction of link prediction problem and how it has been addressed by various approaches. We conclude this chapter by summarizing common disadvantages which drew us towards the presented research.

Chapter 3: Machine learning for link prediction starts with the discussion of methods which have been used for link prediction. It particularly focused on machine learning methods used in the past research. We describe the key aspects of supervised and unsupervised learning methods with reasoning why we selected

supervised learning method in this research.

Chapter 4: Time score introduce a novel time-aware feature defined on common neighbors. We have shown the effectiveness of *Time score* in link prediction with experimental evaluation using real world data.

Chapter 5: T_Flow algorithm presents a novel extension of a random walk algorithm, *PropFlow* [57], which have used for link prediction. The novelty of *T_Flow* algorithm is that it is sensitive to the dynamic behavior of links. The experimental results confirms that *T_Flow* outperform the previous algorithm. Besides that, we argue that this method is applicable to any flow-based algorithm.

Chapter 6: Conclusion and Future works summarize and discuss the contributions of this research work. It also discuss the limitations of the features introduced in this research work, and and future directions of our research.

The sole purpose of this thesis is to provide in detail description of the time-aware features we invented in our research to the research community who are interested in using them. The novel methods introduced here is generally applicable to any sort of social network.

Chapter 2

Related work

Summary: *Link prediction for social network data is a fundamental data mining task in various application domains, including social network analysis, information retrieval, recommendation systems, record linkage, marketing and bioinformatics. Link prediction research has been attracted great deal of attention with the surge of online social networking services. In this chapter, we review some of the state-of-the-art link prediction research focused on social networks. We summarized recent progress about link prediction algorithms, emphasizing the contributions from different perspectives and approaches, such as graph theoretic approaches, probabilistic approaches, similarity-based approaches, ect. Those methods/algorithms have been used to extract knowledge regarding the evolution mechanisms of social networks which then can be used to infer the future potential links. Finally, we outline the incompetency of handling the dynamic/temporal behavior of networks by many prediction methods discussed in this chapter.*

2.1 Graph theoretic approaches

Graph theory or network theory is a mathematical approach to study and model the structure of graphs or networks. In the mathematical literature, network is a collection of nodes joined by links. Mathematical models has been extensively used in link prediction research to foresee the future form of a current network. Graph theory has been built upon structural patterns of networks which is referred

as graph topology. Topological properties such as clustering coefficient, shortest paths, average path length, betweenness centrality, closeness centrality, degree distribution, etc. of a network can be used to derive the principles of network evolution models. The descriptions of some fundamental topological features mentioned above is as follows:

- *Clustering coefficient C* is defined as:

$$C = \frac{3 * number\ of\ triangles\ in\ the\ network}{Number\ of\ connect\ triples\ of\ vertices} \tag{2.1}$$

- *Shortest paths* is a fundamental concept in graph theory is the geodesic distance or shortest path of edges that links two given vertices. There may not be a unique geodesic distance between two vertices. A node pair can have two or more shortest paths.

- *Average path length* is one of the three most robust measures of network topology, along with its clustering coefficient and its degree distribution. Consider an unweighted graph G with n nodes. Let $d(i, j)$ denotes the shortest path between nodes i and j. If $i \neq j$ then the average path length l_G is defined as;

$$l_G = \frac{1}{n(n-1)} \sum_{i,j} d(i, j) \tag{2.2}$$

- *Betweenness centrality* quantifies the number of shortest paths pass through a node.

- *Closeness centrality* can be regarded as a measure of how much a node close to the other nodes in a network. Let G be a graph and $i, j \in G$ are nodes. Let $d_s(i, j)$ be the shortest path between them. The *Closeness centrality* C_c of i defined as;

$$C_c = \frac{1}{\sum_{j \in G} d_s(i, j)} \tag{2.3}$$

- *Degree distribution* is the probability distribution of these degrees over the whole network.

Except the above topological features, there are various structural features have been introduced in the past research. Most of them are variants or extensions of the fundamental topological features. Newman et. al. have extensively studied the social network evolution using scientific collaboration networks [78]. They proposed models to incorporate topological patterns such as clustering coefficient to the random graph models to make robust graph generation models which are used to study the evolution of modern social networks [80]. The results of such investigations show that evolution of most scientific collaboration networks obey the above mentioned principle graph theories [77]. Further, they devised a method to construct weighted graphs using coauthorships. The weights used as the strength of collaborative links. These new methods lead to build more robust yet complicated models for network evolution.

Recently, after a surge in interest in network structure among researchers as a result of research on the Internet and the online social networks, another branch of research has investigated the statistical properties of networks and methods for modeling networks either analytically or numerically. One important and fundamental result that has emerged from these studies concerns the numbers of links that nodes have to other nodes, their so-called "degrees". It has been found that in many networks, the distribution of node degree is highly skewed, with a small number of nodes having an unusually large number of links [79]. Empirical studies have proposed number of random graph models based on degree distribution. Erdös and Rènyi model is, arguably the most famous one of them. This random graph model is simple to define. One takes some number N of nodes or vertices and places links or edges between them, such that each pair of vertices i, j has a connecting edge with independent probability p. Consider a node i in a random graph with n number of nodes. It is connected with equal probability p with each of the $n - 1$ other vertices in the graph, and hence the probability p_k that i has degree exactly k is given by the binomial distribution:

$$p_k = \binom{n-1}{k} p^k (1-p)^{n-1-k} \tag{2.4}$$

However, as a model of a real-world network, it has some serious shortcomings. Perhaps the most serious is its degree distribution, which is quite unlike those

seen in most real-world networks due to the independence assumption is not true in general.

The well-known Barabȁi-Albert model [6], which is based on preferential attachment, is an algorithm for generating random scale-free networks using a preferential attachment mechanism. Scale-free networks are widely observed in natural and human-made systems, including the Internet, the world wide web, citation networks, and some social networks. In the Barabȁi-Albert model, new links are attached to nodes using a probability distribution weighted by node degree, resulting in linear preferential attachment. New nodes are added to the network one at a time. Each new node is connected to existing nodes with a probability that is proportional to the number of links that the existing nodes already have. Formally, the probability p_i that the new node is connected to node i is:

$$p_i = \frac{k_i}{\sum_j k_j} \tag{2.5}$$

where k_i is the degree of node i and the sum is made over degree of all pre-existing nodes j (i.e. the denominator results in the current number of edges in the network). Higher degree tend to quickly accumulate even more links, while nodes with only a few links or lower degree are unlikely to be chosen as the destination for a new link. The new nodes have a "preference" to attach themselves to the already heavily linked or higher degree nodes.

Another model introduced by Clauset et. al. based on power-law distribution [17]. This model is widely used to model scale-free networks. A power law is a type of probability distribution if the frequency (with which an event occurs) varies as a power of some attribute of that event (e.g. its size), the frequency is said to follow a power law. In the context of networks, the frequency is the number of nodes and the attribute is the degree of the nodes. The frequency of nodes decrease according to the power law as node degree increases. A scale-free network is one with a power-law degree distribution. Scale-free networks are a type of network characterized by the presence of large hubs, that is, there exist few nodes which are highly connected. For an undirected network, we can just

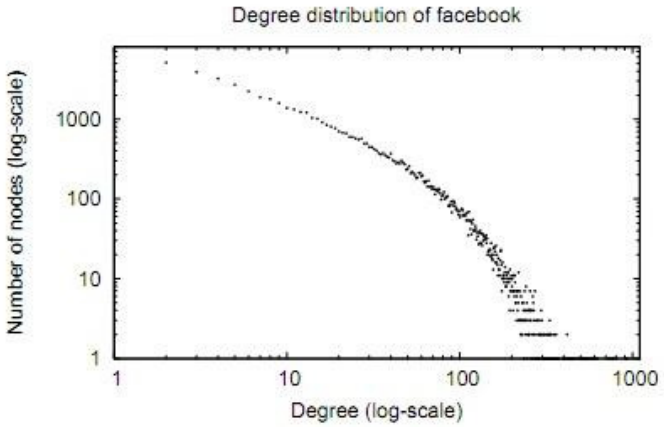


Figure 2.1: Power-law degree distribution of facebook

write the degree distribution as;

$$P_{deg}(k) \propto k^{-\gamma} \tag{2.6}$$

where γ is some exponent. This form of $P_{deg}(k)$ decays slowly as the degree k increases, increasing the likelihood of finding a node with a very large degree. Figure 2.1 shows the degree distribution of facebook which as an example of power-law distributions in the modern social networks.

The most of graph theoretic approaches have been tried to model the graph evolution mechanism at macroscopic level. In contrast, some other methods such as probabilistic and similarity based methods have been introduced to model the network evolution at microscopic level. We discuss those methods in the following sections.

2.2 Probabilistic approaches

In principle, probabilistic approaches try to estimate the likelihood of potential links. The potential links with higher probabilities are more likely to happen than the links with lower probabilities. In contrast, the models such as exponential random graphs are used to estimate probabilistic models for the whole network. Thus, the probabilistic approaches can be put into two groups which models esti-

mate probabilities of individual potential links and models estimate probabilities of potential structures of a current network. Besides that, we have to note that the probabilistic methods are mostly based on the graph theoretical approaches described in the proceeding section.

In recent years, there has been growing interest in exponential random graph models for social networks, commonly called the P^* class of models. The exponential random graph models(ERGMs) [87] are a popular approach to estimate probabilistic models for a whole network using global features of a network, nodes and edges. These models have been built upon statistical models which allow to inference about whether certain network substructures, often represented in the model by one or a small number of parameters, are more commonly observed in the network than might be expected by chance. We can then develop hypotheses about the social processes that might produce these structural properties. Exponential random graph models is an attempt to build plausible models for networks by overcoming limitations of early graph theoretic approaches. The general form of the exponential random graph model for an observed graph Y is:

$$P_r(Y = y) = \frac{1}{k} exp\Big\{ \sum_A \eta_A g_A(y) \Big\} \tag{2.7}$$

where (1) $P_r(Y = y)$ is the probability of Y taking the form y, (2) η_A is the parameter corresponding to the configuration A such as triangle, connected triple, etc. (η_A is non-zero only if all pairs of variables in A are assumed to be conditionally dependent given the rest of the graph), (3) $g_A(y) = \prod_{y_{ij} \in A} Y_{ij}$ is the network statistic corresponding to configuration A; $g_A(y) = 1$ if the configuration is observed in the network y, and is 0 otherwise. y_{ij} is a random variable denote the existence of link between node i and j, (4) k is a normalizing quantity which ensures that Equation 2.7 is a proper probability distribution. In general, exponential random graph models are a good solution for study the evolution of small-world networks which have small number of nodes and links. For complex and large networks ERGMs are not the best models, and applying for large networks will have to pay high cost for computing.

Recent link prediction research focused mostly on temporal and local patterns of networks. It has been shown that temporal and local patterns has significant

impact on link evolution. In the recent probabilistic approaches, local probabilistic methods have been largely introduced and gained increasing popularity due to their flexibility and effectiveness. Wang et. al. proposed a local probabilistic model using probabilistic graph models to estimate the cooccurrence probability of two nodes who resides within a local proximity to each other [104]. Specifically, they have used *Markov Random Fields (MRF)* to model the local neighborhood of a node. The local proximity is defined on the path length. There are two main stages in their approach to use graphical models in this context: (a) First, given the candidate link (say between nodes x and y) whose probability is to be estimated, identifying the central neighborhood set (say w, x, y, z), which are the nodes that are deemed germane to the estimation procedure. The identification of the central neighborhood set is governed by the local topology of the social network as viewed from the perspective of the two nodes whose link probability is to be estimated. (b) once the central neighborhood set (w, x, y, z) is identified and learn a maximum entropy Markov random field model that estimates the joint probability of the nodes comprising the central neighborhood set, i.e., $p(w, x, y, z)$. In this context one can leverage the fact that most networks such as coauthorships are computed from an event log (an event corresponding to a publication). Multiway statistics (e.g. non-derivable frequent itemsets whose elements are drawn from (w, x, y, z)) on these event logs can be used to constrain and learn the model parameters efficiently . The resulting model can then be used to estimate the link probability between x and y which is henceforth denote as the cooccurrence probability. The experimental results have shown that this cooccurrence feature is quite effective for link prediction on coauthorship networks. When used in combination with existing topological and semantic features in conjunction with supervised learning methods, the resulting classification performance shows considerable improvements.

Tylenda et. al. investigated the value of incorporating the history information available on the interactions (or links) of the current social network state [102]. In particular, this work is an extension of the local probabilistic model proposed by Wang et al. [104], which described above incorporating time awareness. In this work, they have investigated the impact of considering the temporal evolution of social networks explicitly in link prediction tasks, and make following

contributions: (a) developed graph-based link prediction techniques that incorporate the temporal information contained in evolving social networks. The edge weights possibly derived from temporal features were incorporated into the state-of-the-art link prediction methods, such as the Adamic/Adar and rooted PageRank based techniques. Their results unequivocally show that timestamps of past interactions significantly improve the prediction accuracy of new and recurrent links over rather sophisticated methods proposed recently. One interesting point of this work is they assumed weight of a link in a network is a strictly increasing function of the time of its creation. The oldest and the latest link are assigned weights w_{min} and w_{max} respectively. Note that $w_{min} \geq 0$ and $w_{max} \geq w_{min}$. In the experiments they have used three functions. If t denotes the time of a link normalized in such way that the beginning of the data set corresponds to 0.0 and the end to 1.0, then the weighting functions are scaled and shifted variants of $exp(3t)$, t and $\sqrt{t}$. The experimental results showed that time of interactions between entities is a dominant feature for ranking neighboring nodes based on their probability of future interaction with the central node. The time-aware methods introduced in this work is used to rank the top k candidates which are likely linked with a given node V. The possible candidates are selected from a local neighborhood of node v. This unsupervised ranking method shows impressive results compared to the other unsupervised ranking methods such as Adamic/Ader, common neighbors, Jaccard's coefficient and rooted pagerank.

Kashima et. al. introduces a new approach to the problem of link prediction for network structured domains, such as the Web, social networks, and biological networks [39]. Their approach is based on the topological features of network structures, not on the node features. They have presented a novel parameterized probabilistic model of network evolution and derive an efficient incremental learning algorithm for such models, which is then used to predict links among the nodes. This method computes the probability of creating an edge from one node to another over time by assuming that state of an edge at time $t + 1$ depends on the state t, which is somewhat similar to markov assumption. In this model, probabilistic flips of the existence of edges are modeled by a certain "copy-and-paste" mechanism between the edges. This link prediction algorithm is derived by assuming that the network structure is in a stationary state of the network. This

allows one to formalize the inference of the stationary state as a transduction problem, and propose an Expectation-Maximization (EM)-based transduction method. The algorithm embodies a maximum likelihood estimation procedure using exponentiated gradient ascent. The basic idea behind this model is as follows; if you have a friend who has a strong influence on you, your association will be highly affected by the friend's association. Also, if a gene is duplicated in the course of genetic evolution, the copied gene will have similar characteristics to the original one. Assume that node k has a strong influence on node i, and there is an edge between node k and node j. Following the above hypothesis, there will likely be an edge between node i and node j. Similarly, if there are no edges between k and j, there will likely be no edge between i and j. In other words, node k can copy-and-paste one of its edge labels to i and j. An edge label $\phi(i, j)$ indicates the probability that an edge exists between any pair of nodes. In particular, $\phi(i, j) = 1$ if an edge exists between nodes i and j, and $\phi(i, j) = 0$ if an edge does not exist between nodes i and j. Note that ϕ is symmetric. This method has been tested on biological networks. The experimental results show a promising improvements in link prediction.

In some cases, the structural information of networks is completely missing or partly available while the node information available. Thus, link prediction task becomes more challenging because the topological features are no longer available. Leroy et.al. has proposed a two-phase method based on bootstrap probabilistic graph as a solution to the above problem [46]. In the first phase, this approach build a bootstrap probabilistic graph where its edges have probabilities which is computed using group memberships of nodes. In the second phase, the graph based features of probabilistic bootstrap graph is used to derive new probabilities of edges which is regarded as the final outcome. This method has been tested on Flickr data set. The idea behind this method is as follows: We are given a set U of users and a multiset G of groups of users. We denote the set of groups to which a user u belongs to, $m(u) = \{g \in G | u \in g, g \subseteq U\}$, as his/her membership set. Now the task is to reconstruct the links of a social graph $N = (U, A)$, where the nodes are the users and the arcs $A \subseteq U \times U$ represent a (one-way) relation between two users. Reconstructing the social network N means to predict which of the links in $U \times U$ actually exist in A, or in other terms, to build a function

$f : U \times U \rightarrow 0, 1$. Proposed solution is a two-phase method based on the bootstrap probabilistic graph for cold start link prediction. During the first phase, we predict the existence of links based only on the group membership information. The output of the first phase is the bootstrap probabilistic graph, i.e., a directed probabilistic graph $BPG = (U, E, p_1)$, where $E \subseteq U \times U$, and every link $(u, v) \in E$ is labeled with a probability $p_1(u, v) > 0$ representing the confidence (or uncertainty) about the link's existence, i.e., $p_1 : U \times U \rightarrow [0, 1]$. In particular, after the first phase, we have $p_1(u, v) = 0$ and $p_1(v, u) = 0$ for every user pair (u, v), where $m(u) \cap m(v) =$. This is because if two users have no groups in common, a prediction cannot be made about the existence of a link between them. Moreover, we have $p_1(u, v) > 0$ for every user pair (u, v) such that $m(u) \cap m(v) =$ (this will also hold for the reverse arc (v, u)). Links with null probabilities do not exist in BPG. The second phase takes as input the bootstrap probabilistic graph BPG, and it refines the probability distribution p_1 into a new probability distribution p_2 , by means of graph based features. Therefore, the output of the second phase is a probabilistic graph $PG = (U, E, p_2)$. After the second phase, some links that previously had $p_1(u, v) = 0$ can now possibly have a non-null score, $p_2(u, v) > 0$, thus extending the overall recall of the method .

Popescul et. al. used statistical relational learning method to predict citations in the domain of scientific publications [85]. Link prediction models in this domain can be used as citation recommender services. This service can potentially be deployed to recommend citations to users who provide the abstract, names of the authors and possibly a partial reference list of a paper in progress. This method composed of two main processes: generation of feature candidates from relational data and their selection with statistical model selection criteria. In addition to prediction, the learned features have an explanatory power, providing insights into the nature of the citation graph structure. A statistical relational model for a given database shows not only the correlations between attributes of each table, but also dependencies among attributes of different tables. For example, publication data can be put in a relational database tables using schema:

```
Citation(from:Document, to:Document),
Author(doc:Document, auth:Person),
PublishedIn(doc:Document, vn:Venue).
```

So, the learner has to learn a model from the relational data using relational algebra which is used for relational feature generation.

Krzysztof et. al. proposed a predictive model of structural changes in elementary subgraphs of social network based on mixture of Markov Chains [37]. The model is trained and verified on a dataset from a large corporate social network analyzed in short, one day-long time windows, and reveals distinctive patterns of evolution of connections on the level of local network topology. They claimed that the network investigated in such short timescales is highly dynamic and therefore immune to classic methods of link prediction and structural analysis, and show that in the case of complex networks, the dynamic subgraph mining may lead to better prediction accuracy . This research has been suggested that the accurate predictions for fast-changing social networks observed in short periods of time require the analysis of dependencies and correlations of the activity of the nodes which may be described in terms of temporal patterns of changes in local network topology. Therefore, they analyzed them from the level of the simplest of these patterns - the connections between triples of nodes. There are 64 different connection patterns in a directed network of labeled nodes has been identified in this analysis. A model based on mixture of markov chains has been used with expectation-maximization algorithm, and tested on real world network data. The results have shown that it is possible to predict the evolution of the links in node triads of fast-changing social network with a good accuracy.

Maximum likelihood methods also a popular probabilistic/statical method for modeling network evolution. Using a methodology based on the maximum likelihood principle, Leskovec et. al. investigated a wide variety of network formation strategies, and show that edge locality plays a critical role in evolution of networks. Their findings supplement earlier network models based on the inherently non-local preferential attachment. Leskovec et. al. presented a detailed study of network evolution by analyzing four large online social networks namely, delicious*, flickr, answers†, and LinkedIn, with full temporal information about node and edge arrivals [47]. For the first time at such a large scale, they have studied the individual node arrival and edge creation processes that collectively lead to macroscopic properties of networks. Based on their observations, they have develop a

* delicious.com † www.answers.com

complete model of network evolution, where nodes arrive at a pre-specified rate and select their lifetimes. Each node then independently initiates edges according to a "gap" process, selecting a destination for each edge according to a simple triangle-closing model free of any parameters. The authors have shown analytically that the combination of the gap distribution with the node lifetime leads to a power law out-degree distribution that accurately reflects the true network in all four cases. Finally, they have given model parameter settings that allow automatic evolution and generation of realistic synthetic networks of arbitrary scale.

The probabilistic methods discussed above have not been able to cover all the aspects of node similarities. Hence, some research has been done to explore other similarity measures which can effectively improve the link prediction accuracy.

2.3 Similarity based approaches

The prominent characteristic of similarity based methods is that they measure the similarity/dissimilarity of node pairs to assign a score or weight to them. More similar node pairs get higher scores or weights and are more likely to link in the future. Another characteristic of similarity base approaches is that they have used a set of similarity measures rather than isolated similarity measure. This is quite advantageous when supervised learning methods used for link prediction. There has been number of similarity measures have been introduced for link prediction. In general, the similarity based methods are adapted from techniques used in graph theory and in social-network analysis; in a number of cases, these techniques were not designed to measure node-to-node similarity and hence need to be modified for this purpose.

Link prediction in coauthorship networks has been a major focus of link prediction research. Consider a coauthorship network among scientists. There are many reasons exogenous to the network why two scientists who have never written an article together will do so in the next few years: For example, they may happen to become geographically close when one of them changes institutions. Such collaborations can be hard to predict. But one also senses that a large number of new collaborations are hinted at by the topology of the network: Two scientists who are "close" in the network will have colleagues in common and will travel in sim-

ilar circles, conferences; this social proximity suggests that they themselves are more likely to collaborate in the near future. Liben et. al. have done lot of work to make this intuitive notion precise and to understand which measures of "proximity" in a network lead to the most accurate link predictions [52]. They found that a number of proximity measures lead to predictions that outperform chance by factors of 40% to 50%, indicating that the network topology does indeed contain latent information from which to infer future interactions. Moreover, certain fairly subtle measures—involving infinite sums over paths in the network—often outperform more direct measures such as shortest-path distances and numbers of shared neighbors. Their similarity methods consist of neighborhood based methods, path based methods and some high level approaches. Neighborhood methods includes *Common neighbors, Jaccard's coefficient, Adamic/Adar and Preferential attachment* while the path based methods includes *Katz index, Hitting time, Page rank, SimRank* . Besides the above methods the authors have used some other high level methods such as *Low-rank approximation, Unseen bigrams and Clustering*. Most of the above methods are summarized in Table 2.1. For a node x, let $\Gamma(x)$ denote the set of neighbors of x in the given coauthorship network G_{collab}. These methods have been extensively used with other link prediction research which we discuss in this section. Link prediction performances of the methods shown in Table 2.1 have been tested using coauthorship networks extracted from publications of five ares of the physics *e-Print archives*[‡]. Although the results show that most of the above methods outperforms the random predictors, there is no method generally perform better on every coauthorship networks. It implies that different method(s) works better on different data sets.

For predicting coauthorships, semantic descriptions of authors or researchers might be very helpful for predicting links or collaborations in coauthorship networks. If one knew to what extent each researcher is an expert in each field, one could potentially use this knowledge to find researchers with compatible expertise and suggest collaborations. However, semantic descriptions are often unavailable due to lack of supporting vocabulary. Therefore, structural attributes from the graph of past collaborations/coauthorships can be used to train a set of predictors using supervised learning algorithms. These predictors can then be used to pre-

[‡] www.arxiv.org

Table 2.1: A list of similarity based methods

Method	**Formula**
Adamic/Adar	$\sum_{z \epsilon \Gamma(x) \bigcap \Gamma(y)} \frac{1}{log\|\Gamma(z)\|}$
Common neighbors	$\|\Gamma(x) \bigcap \Gamma(y)\|$
Jaccard's coefficient	$\frac{\|\Gamma(x) \bigcap \Gamma(y)\|}{\|\Gamma(x) \bigcup \Gamma(y)\|}$
Preferential attachment	$\|\Gamma(x)\|\|\Gamma(y)\|$
$Katz_\beta$	$\sum_{l=1}^{\infty} \beta^l . \|paths_{xy}^l\|$ where $\|paths_{xy}^l\|$ = paths of length exactly l from x to y weighted: $\|paths_{xy}^l\|$ = number of collaborations between x, y unweighted: $\|paths_{xy}^l\| = 1$ iff x and y collaborate
Hitting time Hitting time: stationary-normed Commute time Commute time: stationary-normed	$-H_{xy}$ $-H_{xy}.\pi_{xy}$ $-(H_{xy} + H_{yx})$ $-(H_{xy}.\pi_y + H_{yx}.\pi_x)$ where H_{xy} = expected time for random walk from x to reach y π_y = stationary-distribution weight of y (proportion of time the random walk is at node y)
Rooted PageRank_α	Stationary distribution weight of y under the following random walk: with probability α, jump to x. With probability $(1 - \alpha)$, go to a random neighbor of current node.
SimRank_γ	$\begin{cases} 1 & if\ x = y \\ \gamma . \frac{\sum_{a \in \Gamma x} \sum_{b \in \Gamma y} \textbf{score(a,b)}}{\Gamma x . \Gamma y} & otherwise \end{cases}$

dict future links between existing nodes in the graph. Based on this idea, Pavlov et. al. proposed a supervised learning framework for link prediction in coauthorship networks [84]. In this method, the authors have used many neighborhood based and path based similarity measures , which they termed as features. Once the features are calculated for each node pair, the node pairs are represented by a feature vector which is used as a predictor. The definition of a predictor is very similar to what is known in the machine learning community as a classifier. The results confirm that the appearance of new collaborations is dependent on past network structure and that supervised learning methods can exploit this dependence to make predictions with reasonable accuracy. Since the method itself relies solely on structural attributes of the underlying network and on general supervised learning algorithms, it can be easily extendable to any kinds of networks in which link prediction is desirable. Later, the link prediction in coauthorship networks was improved by introducing some semantic similarity features such as as keyword match count for paper topics and abstracts. The previous approach, only based on structural features implies that researchers are more likely to collaborate with people of there entourage. However, it happens that communities based on the same topic are not related at all, or by very few links, because of the real distance between the people or because of a non-existing partnership between the scientific institutions. In these many cases, having the structure of the graph is not enough to predict the best partner in a specific domain. Therefore, some research focused on combining the structural and non-structural attributes to resolve the problem of link prediction: the most obvious idea for linking researchers is to compare the topics, keywords and abstracts of their research papers. Thus, by counting the number of words in common between all the topics, keywords and abstracts of their previous papers, one can have a new feature based on the semantic and not on the network structure [88, 107]. The authors have conducted experiments on coauthorship networks combining machine learning methods and new features to test the link prediction accuracy The experimental shows that the introduction of semantic features have significantly improved the prediction accuracy.

In the domain of scientific collaborations, citations depict the similarity or relatedness of papers. Hence, the citation networks have most of the characteristics present in the social networks, and number of link prediction research has been

done on citation networks using semantic similarity methods. Shibata et. al. proposed models to predict the existence of citations among papers by formulating link predictions for five large-scale datasets of citation networks [96]. A supervised machine learning model, support vector machine (SVM), has been applied with structural as well as semantic similarity features. Three features in particular, link-based Jaccard coefficient , difference in betweenness centrality , and cosine similarity of term frequency-inverse document frequency vectors, largely affect the predictions of citations. The results also indicate that different models are required for different types of research areas-research fields with a single issue or research fields with multiple issues. In the case of research fields with multiple issues, there are barriers among research fields because the results indicate that papers tend to be cited in each research field locally. Therefore, one must consider the typology of targeted research areas when building models for link prediction in citation networks.

Modern social networking services provide users with options that enable them to share the contents in natural ways. Web 2.0 applications have attracted a considerable amount of attention because their open-ended nature allows users to create lightweight semantic scaffolding to organize and share content. For example, in facebook some one can express the feelings in terms of liking, tagging, commenting, chatting, following, posting, etc. Thus, the necessity of understanding and find similarities of the semantic meanings behind the expressions has become a major topic in link prediction research. Some of the recent research has been focused methods such as natural language processing, content analysis, sentiment analysis, etc. to make accurate predictions on link evolution. To date, the interplay of the social and semantic components of social media has been only partially explored. Schifanella et. al. focused on Flickr and Last.fm, two social media systems in which one can relate the tagging activity of the users with an explicit representation of their social network [91]. The authors have shown that a substantial level of local lexical and topical alignment is observable among users who lie close to each other in the social network. The null model introduced here preserves user activity while removing local correlations, allowing us to disentangle the actual local alignment between users from statistical effects due to the assortative mixing of user activity and centrality in the social network. The null

model has been built in following manner: (1) keep the social network unchanged; (1) built the global list of tags with their multiplicity, i.e. each tag appears the total number of times it has been used; (3) for each user with n_t tags $t_1, t_2, \ldots, t_{n_t}$, with respective frequencies $f_1, f_2, \ldots, f_{n_t}$, we extract n_t distinct tags at random from the global list of tags and assign them to user u with frequencies $f_1, f_2, \ldots, f_{n_t}$. This guarantees that the number of distinct tags and the total number of tag assignments for each user is the same as in the original data, and that the distribution of frequencies of tags is left unchanged. This analysis suggests that users with similar topical interests are more likely to be friends, and therefore semantic similarity measures among users based solely on their annotation metadata should be predictive of social links. The hypothesis was tested on the Last.fm data set, confirming that the social network constructed from semantic similarity captures actual friendship more accurately than Last.fm's suggestions based on listening patterns. The main contributions of this analysis are:

- Show that strong correlations exist across several measures of user activity, and characterize the mixing patterns that involve user activity and user centrality in the social network.

- Developed sound measures of tag overlap, and introduce appropriate null models to disentangle the actual local alignment between users from statistical effects due to the mixing properties of user activity and centrality in the social network. These measures were applied to the Flickr and Last.fm data sets. The resulting analysis shows that, despite neither Flickr nor Last.fm support globally-shared tag vocabularies, a substantial level of local lexical (shared tags) and topical (shared groups) alignment is observable among users who are close to each other in the social network. Also, it has found that some observables are more adequate than others to measure lexical and topical alignment, in the sense that they are less sensitive to purely statistical effects.

- Inquired if the observed correlations between annotation metadata and social proximity allow to use semantic similarity between user annotations as statistical predictors of friendship links. The evaluation of number of semantic similarity measures from the literature, based on Last.fm metadata

resulted that when consider the annotations of the most active users, almost all of the semantic similarity measures considered outperform the neighbor suggestions from the Last.fm system at predicting actual friendship relations. Scalable semantic similarity measures such as Maximum Information Path, proposed by some of the authors, are among those achieving the best predictive performance.

The results shows that using any of the tested social similarity measures were able to improve on the accuracy of the social link predictions provided by Last.fm, and the improvements were especially significant for users who are active taggers. Aiello et. al. further extended the above research by studying the homophily in three systems,namely Last.fm[§], Flickr.com and aNobii[¶], that combine tagging social media with online social networks [3]. They found a substantial level of topical similarity among users who are close to each other in the social network. Those recent research reveal that the relationships or links are formed according to the social science phenomenon, and semantic similarities play a vital for link formation in modern online social networks.

The problem of link prediction actually consists of a family of prediction problems. So far, the previous literature on link prediction is restricted to link prediction within the same network.The similarity measures are not limited to the node and edges in one network. Similarities exist between different social networks. Those similarities can be effectively used for link prediction across multiple social network, which is referred as heterogeneous link prediction. This is a new dimension of link prediction. With the surge of social networks it is difficult to investigate each and every network. Instead of that, one can use the knowledge extracted from similar social networks to study the evolution of the other network. This technology is called transfer learning. This is an emerging trend in link prediction research. Ahmad et. al. proposed a new problem, inter-network link prediction (INLP), which is the problem of predicting the formation of links across networks i.e., given networks G_1 and G_2 the task is to use information from G_1 to make predictions about G_2 and vice versa [2]. Link prediction techniques exploit various techniques like the attributes of the nodes, topological features of the graph or aggregate features of the nodes to make predictions about the links.

[§] www.last.fm [¶] www.anobii.com

The performance of some of these techniques can be enhanced by adding domain knowledge to these techniques. An often neglected source of domain knowledge is social science theories which link back to network processes that may be going on in various social networks. It has shown that the insights from theories of social communication can be effectively use to enhance the internetwork link prediction task. Many of these theories propose the existence of "Structural Signatures" (expected subgraphs) which are likely to be present in certain types of networks. The main idea structural signatures is to identify a set of substructures or subgraphs which are likely to be present in certain types of graphs which are known to be generated by certain social processes. In other words, If we see some transformation between two types subgraphs occurring in sufficiently large number of cases, as compared to such a transformation occurring in purely random graphs, then we can predict that this link is likely to form. It may be happen according to a particular social phenomenon such as theory of homophily, theory of social balance, etc. Since this scenario can be expected to happen across any social network, we can use the knowledge extracted from one network to predict links in other networks where the structural or node level information is not available.

Many important real-world systems, modeled naturally as complex networks, have heterogeneous interactions and complicated dependency structures. Link prediction in such networks must model the influences between heterogeneous relationships and distinguish the formation mechanisms of each link type, a task which is beyond the simple topological features commonly used to score potential links. Davis et. al. introduced a novel probabilistically weighted extension of the Adamic/Adar measure called multi-relational link prediction (MRLP) for heterogeneous information networks, which has used to demonstrate the potential benefits of diverse evidence, particularly in cases where homogeneous relationships are very sparse [21]. The authors also have exposed some fundamental flaws of traditional unsupervised link prediction. They have presented supervised learning approaches for link prediction in multi-relational networks, and demonstrate that a supervised approach to link prediction can enhance performance. The trends and tradeoffs of supervised and unsupervised link prediction in a multi-relational setting has been discussed with the evidence of results on three diverse, real-world heterogeneous information networks. In this research, they have considered the

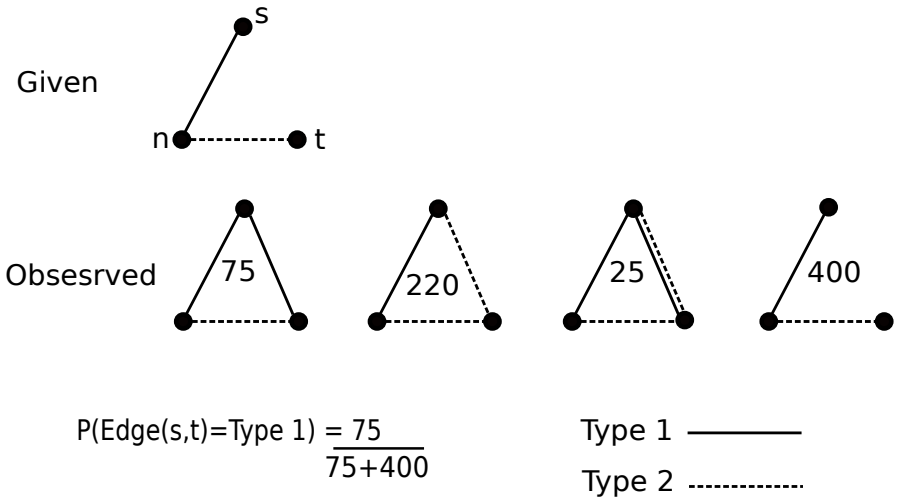


Figure 2.2: Example: calculating the probabilistic weights for MRLP

type dependency between different type links for link evolution. This approach somewhat similar to frequent pattern mining. This method mainly estimate the probabilities of different triads form by different combinations of link types in a given graph as shown in Figure 2.2. It assumes the observed patterns match with given the pattern for potential absent of target link. The most important component of the MRLP method is an appropriate weighting scheme for different edge type combinations. The weights are determined by counting the occurrence of each unique 3-node substructure in the network. The weighted triads is used to define a score for node pair (s, t) as shown in Equation 2.8

$$score_x(s,t) = \sum_{n \in N_s \cap N_t} w_n \tag{2.8}$$

This score has been used as a feature for supervised learning methods to learn a model which is used to predict future potential links as well as their types.

The availability of data such as individual mobility or geographical location allow to discover novel similarity measures and hence, can improve the prediction accuracy. The recent development of technologies to capture the geographical location based data have open a new way to develop more sophisticated features such as geographical distance, visiting co-location, location similarity, etc. Number of link prediction research work have been make use of these features for loca-

tion based link prediction. Even though human movement and mobility patterns have a high degree of freedom and variation, they also exhibit structural patterns due to geographic and social constraints. Using cell phone location data, as well as data from two online location-based social networks, Cho et. al. aimed to understand what basic laws govern human motion and dynamics. They have found that humans experience a combination of periodic movement that is geographically limited and seemingly random jumps correlated with their social networks. Short-ranged travel is periodic both spatially and temporally and not effected by the social network structure, while long-distance travel is more influenced by social network ties. Further analysis show that social relationships can explain about 10% to 30% of all human movement, while periodic behavior explains 50% to 70%. Based on these findings, they have develop a model of human mobility that combines periodic short range movements with travel due to the social network structure. The experimental results show that new model reliably predicts the locations and dynamics of future human movement and gives an order of magnitude better performance than present models of human mobility [15].

Except the link prediction approaches presented in the current section and the proceeding sections, following section presents some of the approaches which have used algebraic methods such as matrix alignment, weighted path based methods such as random walks, etc.

2.4 Other approaches

Except similarity based methods and probabilistic methods, the literature provides numerous diverse approaches stemming from different theoretical backgrounds. Some typical examples are frequent pattern mining, random walk, propagation methods, etc. We discuss some of them in this section.

Lin et. al. proposed a unsupervised link discovery method based on rarity analysis [58]. In this paper they focused on discovering "interesting" paths and nodes from data that can be represented as sets of entities connected by a set of binary relations. In other words, each object in the data set is treated as a separate entity and there are different types of binary relations connecting these entities. This kind of data can naturally be represented by a labeled graph where nodes

stand for entities and links for binary relations. For example, social network data or Web pages with proper classification on hyperlinks can be represented in this way. A key assumption of this work is that the data employs a rich vocabulary of relations where different link types represent different semantic relationships. For example, we could have different links representing that X wrote a letter to Y or that X is the brother of Y. Therefore, different graphs with identical structure will usually have very different meanings depending on the types of links involved. Given these assumptions, we can define the following three classes of novel link discovery problems addressed by in this approach: of multi-relational data . *Novel path discovery:* given an arbitrary pair of entities in a graph, find the most interesting or novel paths between them. *Novel loop discovery:* given an arbitrary entity in a graph, find the most interesting or novel loops starting and ending at it. *Significant node discovery:* given an arbitrary entity in a graph, find other entities that are most significantly connected to it. For example, given some person A, find the set of people that A is most significantly connected to A. In order to address the above problems various notions of *rarity* have been used to measure the "interestingness" of paths between nodes. To deal with novel path discovery problems, rarity analysis is important as it carries the information of interestingness. That is, an event that occurs infrequently compared to other events has the potential to be interesting and, thus, worth to consider it for predicting links. This approach has been used as unsupervised method to discover the different types of links in bibliography data sets.

Kashima et. al introduced a semi-supervised algorithm called "link propagation" [40]. This is similar to label propagation principle, that is, if two nodes have similar to each other they are likely to have the same label. Likewise, if two node pairs are similar to each other they are likely to have the same type of link. In this paper, they have applied label propagation method to pairs of nodes with multiple link types (i.e. (node, node, type)-triplets) and predict the relationships among the nodes. Since it need a triplet-wise similarity matrix to apply the label propagation idea to triplets, the Kronecker product and the Kronecker sum of the element-wise similarity matrices has used. To solve the resultant system of linear equations, the conjugate gradient method is has used. Since naive application of the conjugate gradient method causes serious scalability problems, they have used an acceler-

ation technique called "vec-trick" and its generalized versions for tensors, which significantly reduces the computation time and space requirements.

The information propagation in social networks has been a fundamental factor for link evolution. Particularly, it is important for evolution of social networks devoted for information exchange/sharing such as *Twitter*. Cha et. al. carried out an experimental analysis emphasizing the importance of information propagation for social network evolution using *Twitter* social network [10]. In this paper, they have collected and analyze large-scale traces of information dissemination in the Flickr social network. Their analysis is based on crawls of the favorite markings of 2.5 million users on 11 million photos. They showed empirical evidence that (a) social links are the dominant method of information propagation, accounting for more than 50% of the spread of favorite-marked pictures; (b) information spreading is limited to individuals who are within close proximity of the uploaders; and (c) spreading takes a long time at each hop. As a result, one can conclude that content popularity is often localized in the network and popularity of pictures steadily increases over many years. While the popularity pattern observed is natural for many personal photos, this analysis claimed similar trends for popular photos with hundreds of fans. The findings of this work differ from the common expectations about the quick and wide spread of word-of-mouth effect, and they need to be investigated thoroughly.

Random walk based approaches for link prediction are quite common and popular in the recent past. It has been widely use by recent research works. Backstrom et. al. introduced supervised random walk method combining topological features and node/edge level features for link prediction [5]. This supervised random walk method combines the network structure with the characteristics (attributes, features) of nodes and edges of the network into a unified link prediction algorithm. In this method, random walker learns how to bias a PageRank-like random walk on the network so that it visits given nodes (i.e., positive training examples) more often than the others. it achieves this by using node and edge features to learn edge strengths (i.e., random walk transition probabilities) such that the random walk on a such weighted network is more likely to visit "positive" than "negative" nodes. In the context of link prediction, positive nodes are nodes to which new edges will be created in the future, and negative are all other nodes. The supervised learn-

ing task has formulated as if we are given a source node s and training examples about which nodes s will create links to in the future. The goal is to then learn a function that assigns a strength (i.e., random walk transition probability) to each edge so that when computing the random walk scores in such a weighted network nodes to which s creates new links have higher scores to s than nodes to which s does not create links. To achieve this goal, the proposed page rank-like random walk method has used an optimization method to learn edge strengths to bias the random walk. One important point of this research is it has accounted the link age for determining the link strength. The authors have introduced a new feature $(T - t)^{\beta}$, where T is the current time, and t is the link creation time. The parameter was set to different values $\beta = 0.1, 0.3, 0.5$, and tested the link prediction performance of the proposed algorithm. This has shown a considerable impact for the performance and indicates the importance of time-related features which can capture the knowledge of temporal behavior of link strength.

Lichtenwalter et. al introduced a new unsupervised prediction method on networks, *PropFlow*, which corresponds to the probability that a restricted random walk starting at a node v_i ends at a node v_j in l steps or fewer using link weights as transition probabilities [57]. The restrictions are that the walk terminates upon reaching v_j, the destination node, or upon revisiting any node including , v_i, the starting node. The walk selects links based on their weights. This produces a score s_{ij} that can serve as an estimation of the likelihood of new links. *PropFlow* is somewhat similar to Rooted PageRank, but it is a more localized measure of propagation, and is insensitive to topological noise far from the source node. Unlike Rooted PageRank, the computation of *PropFlow* does not require walk restarts or convergence but simply employs a modified breadth-first search restricted to height l. It is thus much faster to compute. It may be used on weighted, unweighted, directed, or undirected networks. In the phone network, *PropFlow* outperforms baseline unsupervised methods such as by $> 15\%$ AUC on average. It outperforms Rooted PageRank by more than 8.75% AUC. The performances of this algorithm has been tested om two data sets; a phone network data set and a coauthorship data set. The baseline methods are common neighbors, Adamic/Adar, Jaccard's coefficient, preferential attachment, and the unweighted Katz measure with $\beta = 0.005$. Although *PropFlow* may be used

in any network, *PropFlow* has special intuitive significance as a link predictor in networks where some resource such as information flows, propagates, or cascades. In transportation networks, when a resource frequently travels from one node through neighbors to another, there is often some cost for the intermediaries. When the expected cost inherent in traveling through intermediaries overcomes the cost of establishing a new link, one can expect formation of that particular link. In transmission networks, the measure represents the link-weighted probability that a randomly outward-propagated transmission sent by one node will reach another. In coauthorship network *PropFlow* doesn't show significant effect for link prediction according to the experimental evaluation of *PropFlow* both as an individual predictor and as a feature in our supervised classification framework.

One of the challenging task in link prediction research is to determine which subset of attributes are important to establish the links observed in a network. It is very important for accurate link prediction. A link between two persons or entities may also be determined by examining their existing ties (e.g. do they have common friends or what kind of common interests they share?). Indeed, what may be appropriate for one data set may not be for another. Scripps et. al. presented a flexible framework that allows us to identify the relevant attributes or topological features that are most well-aligned with the link structure [92]. This work introduced a new discriminative learning technique for link prediction based on the matrix alignment approach. The algorithm automatically determines the most predictive features of the link structure by aligning the adjacency matrix of a network with weighted similarity matrices computed from node attributes and neighborhood topological features. If we are given an adjacency matrix $A = (a_{ij})_{nxn}$ of a network and data matrix $X = (a_{ik})_{nxd}$, where each x_{ik} represents the k^{th} attribute value of node i. In an ideal network, one can imagine perfect alignment between the links and the attributes - that is where $\forall i, j : sim(x_i, x_j) = a_{ij}$. However, in most networks, such perfect alignment will not exist. The proposed matrix alignment framework uses a set of weights to determine the important attributes for establishing links between nodes. More specifically, goal is to learn a set of weights $\vec{w} = \{w_1, \ldots, w_d\}$ that minimizes the objective function $L = \| A - XWX^T \|_F^2$, where the diagonal elements of W correspond to $\vec{w}$. Intuitively, the objective function aims to learn a set of weights that maximizes the degree of alignment between the

link structure and attribute similarity. To avoid overfitting, a regularization technique has been employed by adding a penalty term $\lambda \parallel W - I \parallel_F^2$ to the objective function, where I is the identity matrix:

$$L = \parallel A - XWX^T \parallel_F^2 + \lambda \parallel W - I \parallel_F^2 \tag{2.9}$$

This will coerce the weight vector w to ones for high values of λ, which is equivalent to assigning equal importance to all the attributes. Experimental results on a variety of network data have demonstrated the effectiveness of this approach.

In online social networks, where the notion of "friendship" is broader than what would generally be considered in sociological studies, the friendship links are denser but the links contain noisier information (i.e., some weaker relationships). However, the networks also contain additional transactional events among entities (e.g., commenting, chatting, tagging, liking, etc.) that can be used to infer the true underlying social network. To this end, Kahanda et. al. introduced a supervised learning approach to predict link strength from transactional information [38]. It is formulated this as a link prediction task and compare the utility of attribute-based, topological, and transactional features. The novelty of this approach is the use of transactional features. This has prime importance because the emerging online social networks are multi-relational. Therefore, it is worthwhile to study the effectiveness of transactions happens via each type of relation. Transactional features consider the transactional information between users (i.e., wall postings, picture postings, linking and group membership, etc.). These features only consider single edges in the transactional graphs; they do not consider the larger relational context of those transactions. For example, one feature counts the number of posts from node v_j on node v_i's wall; another counts the number of photos posted by node v_j and tagged as containing node vi . However, the features do not consider the other transactional activity of nodes v_i and v_j. This approach has been evaluated on public data from the Purdue facebook network and shows that it can accurately predict strong relationships. Moreover, it emphasis that transactional-network features are the most influential features for this task.

One of the most commonly used and successful recommendation algorithms

is collaborative filtering, which explores the correlations within user-item interactions to infer user interests and preferences. However, the recommendation quality of collaborative filtering approaches is greatly limited by the data sparsity problem. To alleviate this problem, Huang et.al have introduced an extension of graph-based algorithms by representing user-item interactions as graphs and employing link prediction approaches proposed in the recent network modeling literature for making collaborative filtering recommendations [33]. They have adapted a wide range of linkage measures for making recommendations. The preliminary experimental results based on a book recommendation dataset show that some of these measures achieved significantly better performance than standard collaborative filtering algorithms. In many security informatics applications, it is important to monitor traffic over various communication channels and efficiently identify those communications that are unusual for further investigation. Huang et. al. investigated such anomaly detection problems using a graph-theoretic link prediction approach [35]. Data from the publicly-available Enron email corpus were used to validate the proposed approach. An underlying assumption of biomedical informatics is that decisions can be more informed when professionals are assisted by analytical systems. For this purpose, Johnson et. al. proposed ALIVE, a multi-relational link prediction and visualization environment for the health-care domain [36]. ALIVE combines novel link prediction methods with a simple user interface and intuitive visualization of data to enhance the decision-making process for health-care professionals. It also includes a novel link prediction algorithm, MRPF, which outperforms many comparable algorithms on multiple networks in the biomedical domain.

The features and methods introduced in the proceeding section have used quite often with learning methods for link prediction. The use of learning methods in conjunction with probabilistic, similarity or any other features, has shown immense predictive power. Thus, it is worthwhile to review some machine learning methods used for link prediction.

Chapter 3

Machine learning for link prediction

Summary: *Machine learning methods have been extensively used in link prediction research. It has shown that machine learning methods are extremely reliable and easy to use tool for the binary classification task of existence or non-existence of links using set of features. The present chapter starts with our problem definition and the following sections discuss supervised and unsupervised learning methods, and their usage for link prediction. In our approach, we used supervised learning method for link prediction. Further, we presented the set of features we used in our experiments combining supervised machine learning methods.*

3.1 Problem definition

We consider the classical problem of link prediction where we are given a snapshot of a social network at time t, and we seek to accurately predict the edges that will be added to the network during the interval from time t to a given future time $t + 1$. More concretely, we are given a large network, say facebook, at time t and for each user we would like to predict what new edges (friendships) that user will create between t and some future time $t + 1$. In that case, we focused on activeness of nodes/links. Activeness of nodes/links has strong correlation with new link formation. We learned that timestamps of links or interactions are strongly correlated with the activeness of nodes/links. Thus, we determine to make use of them to build novel features which can effectively treat the activeness of node/links.

Our problem definition is as follows: *Suppose we are given a network G at time T with set of nodes* $V = \{v_1, v_2, v_3, \ldots, v_i, \ldots, v_n\}$ *and set of links* $E = \{e_{ij}/i, j \in V\}$ *with the most recent timestamps* $t_E = \{t_{ij}/i, j \in V \wedge e_{ij} \in E\}$ *of the links or interactions occurred via the existing links. Our goal is to predict the potential links of G at time* $T + 1$ *using existing features, and timestamps which is correlated with the temporality.* We tested link prediction performance of novel features in conjunction with supervised machine learning method which is described in the following sections.

3.2 Machine learning methods

Machine learning algorithms are described as either "supervised" or "unsupervised". The distinction between two methods drawn from how the learning method classifies data. Both supervised and unsupervised learning methods have been used in previous studies with different frameworks for link prediction. There are pros and cons in both supervised and unsupervised methods but supervised methods has been showed better performances than the unsupervised methods [61]. However, machine learning methods remain immense challenge due to the complexity and size of the networks as well as the temporal behaviors the networks. We now discuss the usage of supervised and unsupervised learning methods for link perdition.

3.2.1 Unsupervised learning for link prediction

Unsupervised learning methods used to learn models from unlabeled data. They are not provided with classes in advance. In fact, the basic task of unsupervised learning is to learn class labels automatically. In order to develop class labels, similarities or distance between data points are taken in to account by the unsupervised learning algorithms. Similar data is grouped in to "clusters" and labeled as a class. However, number of clusters may have to determined beforehand which is a difficult and arbitrary decision to make. In the context of link prediction it looks much simpler than the complex clustering problem because link prediction problem is basically a binary classification task. The main task is to find an effec-

tive unsupervised method to classify the instances in a way that either existence or non-existence of a link.

Almost every features which describe the similarity between node pairs can be used as unsupervised methods for link prediction. Most unsupervised methods either generate scores based on node neighborhoods or path information to assign scores to potential links [57]. It can be number of common neighbors, number of shortest paths, Jaccard's coefficient etc. Most of the link prediction research have used the unsupervised methods as the base lines methods for comparison. The classification task in link prediction is to determine whether a link will appear or not. Thus, a threshold is set for the scores where the links having scores higher than the threshold are classified as potential links while others are not. In general, this binary classification task is not been well achieved by unsupervised methods and hence these methods are not popular in link prediction research. However, Link prediction research based on local proximity or neighborhood also have effectively used unsupervised methods. In that case the task is to rank a limited number of candidate nodes within a local neighborhood, mostly 2-hops, and select the top k candidates as the future potential nodes to make links [102]. Lin et. al. proposed an unsupervised method for link prediction in multi-relational data [58]. In multi-relational environment each link has a class label such as coauthored, cited, published, etc. The task is to predict potential links with its classes. So, the unsupervised methods has become handy in this problem setting. The amount of research used only unsupervised methods is proportionately low. Most of the link prediction research rely on supervised methods rather than unsupervised methods because the limited applicability of unsupervised methods. Next section presents a discussion of supervised learning methods for link prediction.

3.2.2 Supervised learning for link prediction

Supervised classification is one of the tasks most frequently carried out by so called Intelligent Systems. Thus, a large number of techniques have been developed based on Artificial Intelligence (Logic-based techniques, Perceptron-based techniques) and Statistics (Bayesian Networks, Instance-based techniques). The goal of supervised learning is to build a concise model of the distribution of class

labels in terms of predictor features. The resulting classifier is then used to assign class labels to the testing instances where the values of the predictor features are known, but the value of the class label is unknown [44].

In supervised algorithms, the classes are predetermined and the examples are labeled with the corresponding classes. These classes can be a finite set. The supervised machine learner's task is to search for patterns and construct mathematical models. These models then are evaluated on the basis of their predictive capacity in relation to measures of variance in the data itself. The available methods such as decision tree induction, naive Bayes, support vector machine [19], logistic regression, etc. are examples of supervised learning techniques.

Unlike unsupervised methods, supervised methods has been atop among the link prediction methods. Some of the past research has developed supervised algorithms-particularly for link prediction. Doppa et. al. proposed a learning algorithm for link prediction based on chance constraints [23]. The accuracy of current prediction methods is quite low due to the extreme class skew and the large number of potential links. Hence they proposed learning algorithms based on chance constrained programs and show that they exhibit all the properties needed for a good link predictor, namely, they allow preferential bias to positive or negative class; handle skewness in the data; and scale to large networks. Their experimental results on three real world domains coauthorship networks, biological networks and citation networks show significant performance improvement over base line algorithms. Backstrom et. al. introduced a supervised random walk algorithm for link prediction [5]. Random walker use edge strengths as transition probabilities. Edge strengths are learned using page rank scores of nodes and gradient based optimization technique. Lu et. al. proposed a novel and general framework for supervised link prediction. Their model can effectively and efficiently learn the network dynamics from a time series of network snapshots, and therefore improve the link prediction accuracy. In addition, multiple graphs over the same set of vertices but from different sources can be naturally incorporated into this framework. They have performed extensive set of experiments on real world data sets. The experimental results confirm that prediction accuracy is improved using supervision and multiple sources of information [61].

The prominent feature of supervised learning is feature construction and col-

lective classification using a learned model. Once the features are computed for a particular node pair, we obtain a vector of values referred to as a feature vector, which may be correlated with the future possible link between that node pair. We train the learning system with the set of *feature vectors* computed for training data. Then the model is used to predict the future links [84]. The training feature vectors, are labeled with a binary label which denotes the node pair is linked or not. The feature vectors are composed of existing features such a number of Adamic/Adar similarity measure, common neighbors, Jaccard's coefficient, preferential attachment as well as novel time-aware features introduced in this thesis. we have discussed the existing features in detail in Section 3.4 and we discuss the novel features in Chapter 4 and Chapter 5.

Most of the previous studies have used decision tree supervised learning method for link prediction, and has shown its consistency as a binary classification method. Hence, we adopt the same strategy by using decision tree supervised learning method in our experiments. In the following section we discuss the overview of decision tree algorithm with J48 implementation of decision tree algorithm which we used in our experimental evaluation.

3.3 Decision tree method

Information produced by data mining techniques can be represented in many different ways. Decision tree structures are a common way to organize classification schemes. In classifying tasks, decision trees visualize what steps are taken to arrive at a classification. Every decision tree begins with what is termed a root node, considered to be the "parent" of every other node. Each node in the tree evaluates an attribute in the data and determines which path it should follow. Typically, the decision test is based on comparing a value against some constant. Classification using a decision tree is performed by routing from the root node until arriving at a leaf node. The illustration in Figure 3.1 provided here is a canonical example in data mining, involving the decision to play or not play based on climate conditions. In this case, outlook is in the position of the root node. The degrees of the node are attribute values. In this example, the child nodes are tests of humidity and windy, leading to the leaf nodes which are the actual classifications. This example

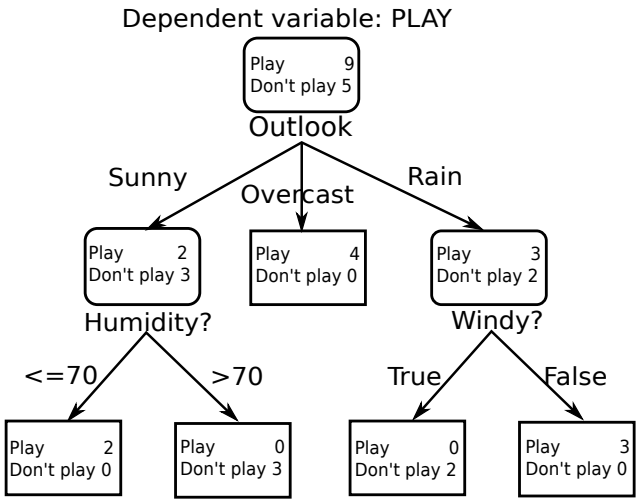


Figure 3.1: Example of a decision tree

also includes the corresponding data, also referred to as instances. In our example, there are 9 "play" days and 5 "no play" days. Decision trees can represent diverse types of data. The simplest and most familiar is numerical data. It is often desirable to organize nominal data as well. Nominal quantities are formally described by a discrete set of symbols. For example, weather can be described in either numeric or nominal fashion. We can quantify the temperature by saying that it is 11 degrees Celsius or 52 degrees Fahrenheit. We could also say that it is cold, cool, mild, warm or hot. The former is an example of numeric data, and the latter is a type of nominal data. More accurately, the example of cold, cool, mild, warm and hot is a special type of nominal data, described as ordinal data. Ordinal data has an implicit assumption of ordered relationships between the values. Continuing with the weather example, we could also have a purely nominal description like sunny, overcast and rainy. These values have no relationships or distance measures. The type of data organized by a tree is important for understanding how the tree works at the node level. Recalling that each node is effectively a test, numeric data is often evaluated in terms of simple mathematical inequality. For example, numeric weather data could be tested by finding if it is greater than 10 degrees Fahrenheit. Nominal data is tested in Boolean fashion; in other words, whether or not it has a particular value. The illustration shows both types of tests. In the weather example, outlook is a nominal data type. The test simply asks which attribute value

is represented and routes accordingly. The humidity node reflects numeric tests, with an inequality of less than or equal to 70, or greater than 70. Decision tree induction algorithms function recursively. First, an attribute must be selected as the root node. In order to create the most efficient (i.e, smallest) tree, the root node must effectively split the data. Each split attempts to pare down a set of instances (the actual data) until they all have the same classification. The best split is the one that provides what is termed the most information gain. Information in this context comes from the concept of entropy from information theory, as developed by Claude Shannon. Although "information" has many contexts, it has a very specific mathematical meaning relating to certainty in decision making. Ideally, each split in the decision tree should bring us closer to a classification. One way to conceptualize this is to see each step along the tree as removing randomness or entropy. Information, expressed as a mathematical quantity, reflects this. For example, consider a very simple classification problem that requires creating a decision tree to decide yes or no based on some data. This is exactly the scenario visualized in the decision tree. Each attributes values will have a certain number of yes or no classifications. If there are equal numbers of "yes"s and "no"s, then there is a great deal of entropy in that value. In this situation, information reaches a maximum. Conversely, if there are only "yes"s or only "no"s the information is also zero. The entropy is low, and the attribute value is very useful for making a decision. The formula for calculating intermediate entropy values is as follows for a random variable with m outcomes $\{1, ..., m\}$;

$$Info = -\sum_{i=1}^{m} p_i log_2 p_i \tag{3.1}$$

We can break this down. Consider trying to calculate the information gain for three variables for one attribute. The attribute as a whole has a total of nine "yes"s and five "no"s. The first variable has two "yes"s and three "no"s. The second has four yeses and zero "no"'s. The final has three "yes"s and two "no"s. Our first step is to calculate the information for each of the variables. Starting with the first, our formula leads us to $info([2, 3])$ being equal to $-(2/5\ log\ 2/5) - (3/5\ log\ 3/5)$. This comes to 0.971 bits. Our second variable is easy to calculate. It only has "yes"s, so it has a value of 0 bits. The final variable is just the reverse of the first

and the value is also 0.971 bits. Having found the information for the variables, we need to calculate the information for the attribute as a whole: 9 yeses and 5 no's. The calculation is $info([9,5]) = -(9/14\ log\ 9/14) - (5/14\ log\ 5/14)$. This comes to 0.940 bits. In decision tree induction, our objective is to find the overall information gain. This is found by averaging the information value of the attribute values. In our case, this is equivalent to finding the information of all the attributes together. We would use the formula $info([2,3],[4,0],[3,2]) = (5/14)\ 0.971 + (4/14)\ 0 + (5/14)\ 0.971$. This comes to 0.6931 bits. The final step is to calculate the overall information gain. Information gain is found by subtracting the information value average by the raw total information of the attribute. Mathematically, we would calculate information gain as follows;

$$Gain = info([9,5]) - info([2,3],[4,0],[3,2]) = 0.940 - 0.693 = 0.247 \quad (3.2)$$

The decision tree induction algorithm will compute this sum for every attribute, and select the one with the highest information gain as the root node, and continue the calculation recursively until the data is completely classified. This approach is one of the fundamental techniques used for decision tree induction. It has a number of possible shortcomings. One common issue arises when an attribute has a large number of uniquely identifying values. An example of this could be social security numbers, or other types of personal identification numbers. In this case, there is an artificially high decision-value to the information, the ID classifies each and every person, and distorts the algorithm by overfitting the data. One solution is to use an information gain ratio that biases attributes with large numbers of distinct values [29].

Decision trees are a classic way to represent information from a machine learning algorithm, and offer a fast and powerful way to express structures in data. One of the questions that arises in a decision tree algorithm is the optimal size of the final tree. A tree that is too large risks overfitting the training data and poorly generalizing to new samples. A common strategy is to grow the tree until each node contains a small number of instances then use pruning to remove nodes that do not provide additional information. The J48 algorithm Weka [24] project gives several options related to tree pruning. J48 is a version of an earlier algorithm

developed by J. Ross Quinlan [86], the very popular C4.5. It is important to understand the variety of options available when using this algorithm, as they can make a significant difference in the quality of results. In many cases, the default settings will prove adequate, but in others, each choice may require some consideration. Many algorithms attempt to "prune", or simplify, their results. Pruning produces fewer, more easily interpreted results. More importantly, pruning can be used as a tool to correct for potential overfitting. The basic algorithm described above recursively classifies until each leaf is pure, meaning that the data has been categorized as close to perfectly as possible. This process ensures maximum accuracy on the training data, but it may create excessive rules that only describe particular idiosyncrasies of that data. When tested on new data, the rules may be less effective. Pruning always reduces the accuracy of a model on training data. This is because pruning employs various means to relax the specificity of the decision tree, hopefully improving its performance on test data. The overall concept is to gradually generalize a decision tree until it gains a balance of flexibility and accuracy.

J48 employs two pruning methods. The first is known as subtree replacement. This means that nodes in a decision tree may be replaced with a leaf, basically reducing the number of tests along a certain path. This process starts from the leaves of the fully formed tree, and works backwards toward the root. The second type of pruning used in J48 is termed subtree raising. In this case, a node may be moved upwards towards the root of the tree, replacing other nodes along the way. Subtree raising often has a negligible effect on decision tree models. There is often no clear way to predict the utility of the option, though it may be advisable to try turning it off if the induction process is taking a long time. This is due to the fact that subtree raising can be somewhat computationally complex. Error rates are used to make actual decisions about which parts of the tree to replace or raise. There are multiple ways to do this. The simplest is to reserve a portion of the training data to test on the decision tree. The reserved portion can then be used as test data for the decision tree, helping to overcome potential overfitting. This approach is known as reduced-error pruning. Though the method is straightforward, it also reduces the overall amount of data available for training the model. For particularly small datasets, it may be advisable to avoid using reduced error

pruning. Other error rate methods statistically analyze the training data and estimate the amount of error inherent in it. The mathematics are somewhat complex, but this approach seeks to forecast the natural variance of the data, and to account for that variance in the decision tree. This approach requires a confidence threshold, which by default is set to 25 percent. This option is important for determining how specific or general the model should be. If the training data is expected to conform fairly closely to the data we would like to test the model on, this figure can be lowered. The reverse is true if the model performs poorly on new data; try decreasing the rate in order to produce a more pruned (i.e, more generalized)tree.

There are several other options that determine the specificity of the model. The minimum number of instances per leaf is one powerful option. This allows you to dictate the lowest number of instances that can constitute a leaf. The higher the number, the more general the tree. Lowering the number will produce more specific trees, as the leaves become more granular. The binary split option is used with numerical data. If turned on, this option will take any numeric attribute and split it into two ranges using an inequality. This greatly limits the number of possible decision points. Rather than allowing for multiple splits based on numeric ranges, this option effectively treats the data as a nominal value. Turning this encourages more generalized trees. There is also an option available for using Laplace smoothing for predicted probabilities. Laplace smoothing is used to prevent probabilities from ever being calculated as zero. This is mainly to avoid possible complications that can arise from zero probabilities. The most basic parameter is the tree pruning option. If we decide to employ tree pruning, we will need to consider the options above. Note that depending on how the training and test data have been defined that the performance of an unpruned tree may superficially appear better than a pruned one. As described above, this can be a result of overfitting. It is important to experiment with models by intelligently adjusting these parameters. Often, only repeated experiments and familiarity with the data will tease out the best set of options [106].

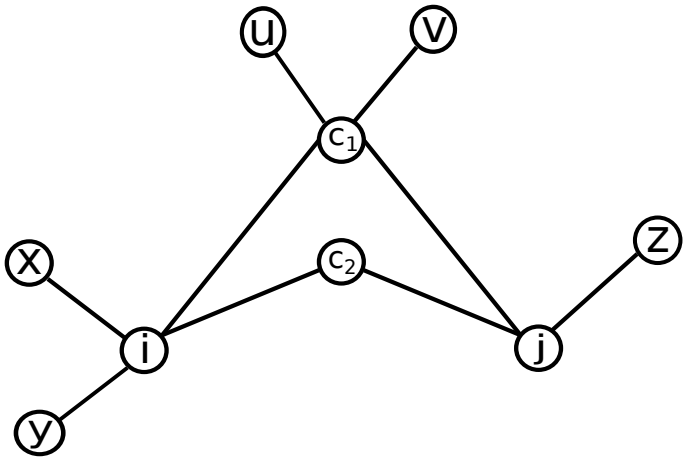


Figure 3.2: Example of a social network

3.4 Features used for link prediction

In this section, we introduce the set of baseline features we used in this research work. This set of features has been consistently used by the previous research and have proven their eminence in social network analysis. In our experiments, we referred follow set of features as *baseline (BC)* combination. Let i, j and k are nodes, and $\Gamma(i)$, $\Gamma(j)$ and $\Gamma(k)$denote the sets of neighbors of i, j and k respectively. We have shown the definitions of the baseline features below and demonstrated their computations using the example social network shown in Figure 3.2.

Adamic/Adar[1] measure indicates if a node pair has a common neighbor which is not common to several other nodes, then the similarity of that particular node pair is higher than the node pairs having neighbors that are common to several other nodes. This measure assigns higher weights to common neighbors that are not common to several other nodes.

$$\sum_{k \epsilon \Gamma(i) \bigcap \Gamma(j)} \frac{1}{log|\Gamma(k)|} \tag{3.3}$$

We can compute *Adamic/Adar* measure for the nodes i and j in Figure 3.2. They have two common neighbors c_1 and c_2. Node c_1 has four neighbors whereas c_2

has two neighbors.

$$Adamic/Adar_{ij} = \frac{1}{log\ 4} + \frac{1}{log\ 2} = 4.983 \tag{3.4}$$

Common neighbors is one of the simplest similarity measure which counts the number of common neighbors of a node pair.

$$|\Gamma(i) \bigcap \Gamma(j)| \tag{3.5}$$

Nodes i and j in Figure 3.2 have two common neighbors c_1 and c_2. Thus;

$$Common\ neighbors_{ij} = 2 \tag{3.6}$$

Jaccard's coefficient[62] is a commonly used similarity metric in information retrieval. In social network analysis it is the normalized measure of common neighbors.

$$\frac{|\Gamma(i) \bigcap \Gamma(j)|}{|\Gamma(i) \bigcup \Gamma(j)|} \tag{3.7}$$

In Figure 3.2 node i has four neighbors $\{x, y, c_1, c_2\}$. Node j has three neighbors $\{z, c_1, c_2\}$. The nodes c_1 and c_2 are common for both i and j. We can compute;

$$Jaccard's\ coefficient_{ij} = \frac{2}{4+3} = 0.286 \tag{3.8}$$

Preferential attachment indicates that new links are more likely to be formed with nodes of higher degree, or nodes that are popular in the network. This has received considerable attention as a model of the growth of networks. The basic premise is that the probability that a new edge has node i as an endpoint is proportional to $|\Gamma i|$, the current number of neighbors of i. on the basis of empirical evidence, Newman and Barabs̀i et. al. further proposed that the probability of link between nodes i and j, is correlated with the product of the number of their degrees [79].

$$|\Gamma(i)||\Gamma(j)| \tag{3.9}$$

The degree of nodes i and j is 4 and 3 respectively. Thus;

$$Preferential\ attachment_{ij} = 4 * 3 = 12 \tag{3.10}$$

The above features have been predominantly used with supervised learning and unsupervised methods for link prediction across many kinds of social networks, and has shown there consistency. However, these methods are static methods which means that none of them considered the dynamic behavior of the links or node. The activeness of nodes vary over time hence, the effectiveness of above features vary over time. For example, the influence of common neighbors is not static over the time. So, by just taking number of common neighbors between a node pair may draw into a miscalculation of similarity between the nodes unless if the activeness is considered. Thus, we determined to develop new time-aware features which can overcome the drawbacks of the baseline features explained here. In the next chapters, we have discussed the new time-aware feature we introduced in this research.

Chapter 4

Time score

Summary: *In this chapter, we introduce a novel time-aware feature, referred to as Time score, that captures the important aspects of time stamps of interactions and the temporality of link strengths of common neighbors. We also analyze the effectiveness of Time score with different parameter settings for different network data sets. The results of the analysis revealed that the Time score was sensitive to different networks and different time measures. We applied Time score to two social network data sets, namely, facebook friendship network data set and a coauthorship network data set. The results revealed a significant improvement in predicting future links.*

4.1 Neighborhood-based features and time-awareness

Social networks are dynamic. They evolve rapidly over time by adding new nodes and new links. Appearance of new links indicate new interactions between nodes in the network. The frequency of interactions and time it happen has strong correlation with link strength/activeness. Therefore, is worthwhile to investigate the how link strength/activeness change over time in social networks. To this end, Burt et. al. has done as extensive analysis of decay of social links. They have used several data sets representing different social networks such as family relations, work/job related, etc. the study has reported some impressive findings regarding the decay of link strengths. In this work, the tendency for relationships

to weaken and disappear is referred as *decay*, and functions describe the rate of decay over time referred as *decaying functions* [7]. The summary of the findings is:

1. The strength of social links depends on social scientific aspects such as homophily, social status/popularity, etc.

2. Long term relationships have slower decay as they are well established links.

3. Decay has pattern over time similar to the population ecology "liability of newness". In other words, people tends to have connections with newly emerging persons in terms of social aspects. Thus, Decay is a power function of time in which the probability of decay decreases with link age and node age. However, notice that there is no specific form for decaying functions, and has to design and estimate parameters according to the facts used to determine the link strengths.

Above implications tells us the use of static features for link prediction is not enough to make accurate predictions due to rapid changes happen in the networks. Particularly, the links become active for a certain time and then fade away. The active links more influential than the weaker or inactive links. However, in static methods assumes that activeness* of links, and hence the activeness of nodes, doesn't change over time. It leads to inaccurate results. Thus, most of the recent research focused on developing time-aware link prediction methods in evolving social networks which can improve the link prediction accuracy as well as overcome the lapses of the static methods.

Tylenda et. al. proposed a time-aware method extending the local probabilistic model for link prediction by [104]. In this work, they have followed similar approach to Burt et. al. by assuming the link strength is a power function of time to assign weights for the links. The oldest and the latest link are assigned weights w_{min} and w_{max} respectively. Note that $w_{min} \geq 0$ and $w_{max} \geq w_{min}$. In the experiments they have used three functions. If t denotes the time of a link normalized in such way that the beginning of the data set corresponds to 0.0 and the end to 1.0,

* We used terms "activeness" and "strength" to refer the same intuitive meaning.

then the weighting functions are scaled and shifted variants of $exp(3t)$, t and $\sqrt{t}$ [102]. However, is bit ambiguous to use a normalized time because the elapsed time with the current time is a critical factor for estimating the link strength. In another research by Backstrom et. al. introduced a new feature $(T - t)^{\beta}$ for assigning weight for links. T is the current time, and t is the link creation time. The parameter was set to different values $\beta = 0.1, 0.3, 0.5$, and tested the link prediction performance of a supervised random walk algorithm [5]. In this case also, the authors have assumed the link strength is a power function of time.

Above mentioned research works are few examples for recent attempts to introduce time-aware methods for link prediction along with some of the well-known features in Chapter 3, Section 3.4. All of these features defined on common neighbors except preferential attachment. The eminence of common neighbors in the realm of social network analysis have been largely discussed and, has introduced many features based on them in the past research. However, the temporality of common neighbors were discussed in a fairly small number of them. It has resulted the common neighbors based features less competent. Thus, we study the correlation between temporality of common neighbors and link evolution and, contributed by introducing a new time-aware feature which is referred as *Time score*, computes a score for common neighbors in terms of link strength and link weights. The link strength is determined by elapsed time from the current time and differences of the most recent timestamps of the interactions or links.

4.2 Time score

Most of the recent link prediction research are being focused on temporal and local patterns of the networks. Number of research works have been introduced time-related features and methods to deal with temporal behavior of node and links. Those features or methods have been defined using social scientific aspects such as *decay* of relationships or social links over time. This phenomenon has been studied extensively in the empirical studies, and revealed that the *decay* of social links is a power function of time. The strength/activeness of social links associated with various factors depending on the network. In most cases, link strength/activeness is strongly correlated with time-related factors but in some

others are not. However, generally, strength of social links strongly correlated with time-related factors. A simplest yet vital factor is *link age*. Link age can be interpreted in two ways: (1) elapsed time since the creation of link (2) elapsed time since the last interaction/transaction[†], with respect to the current time. According to our perception, the second factor is strongly correlated with link activeness. Interactions keep relationships alive and active. Interactions between nodes are very important for link evolution. If transactions or interactions happen frequently and recently the links become active and strong. Active links has an utmost importance for link evolution. Thus, we started from this point and introduced a robust feature to incorporate the effectiveness of common neighbors and their temporality using the activeness of links. In the context of social networks, the effectiveness of the common neighbors depends not only on the cooccurrence frequency, or number of common neighbors, but also on how long the neighbors have been in contact. The time stamps of the interactions are useful in finding such information. This information provides a far better view of the importance of common neighbors than considering only the number of common neighbors. To this end, we designed a new feature based on the following concepts.

1. If a node pair interacted with each other recently with respect to the current time, then the link between them becomes active. This scenario is illuminated in our approach as the decay of a link activeness as a power function of elapsed time since the last interaction with respect to the current time.

2. If a node interacted with its neighbors within a closer proximity of time, the neighbors are more likely to become linked. This scenario is illuminated in our approach as influence of a common neighbor decays as a power function of time difference between latest interaction with its neighbors.

Compiling the above considerations, we introduced a new feature, *Time score(TS)* which can treat the temporal behavior of common neighbors [68]. We use the new feature in conjunction with supervised machine learning methods in order to predict links in network data sets. *Time score* for the node pair a and b that has n common neighbors is defined as follows:

[†] We used terms interaction and transaction interchangeably with the same meaning

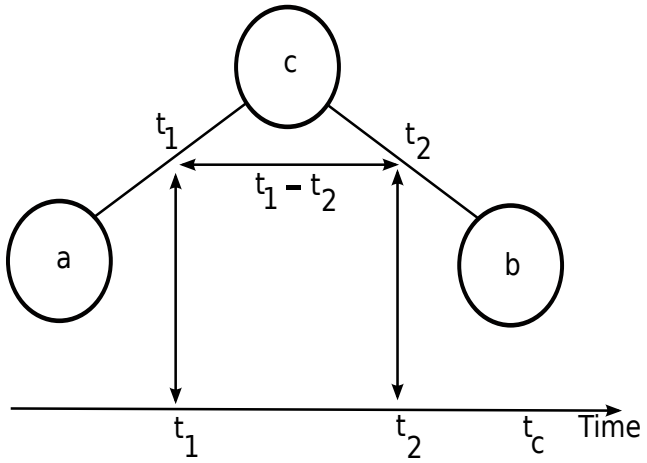


Figure 4.1: Concept of *Time score*

$$TS(a,b) = \sum_{n} \frac{H_{m_n}(1-\alpha)^{k_n}}{|t_{1_n} - t_{2_n}| + 1} \tag{4.1}$$

This concept is illustrated in Fig. 4.1. Nodes a and b have common neighbor c. Here, t_1 is the most recent time stamp of the interactions between a and c, and t_2 is the most recent time stamp of the interactions between b and c. In addition, H_m is the harmonic mean of the cooccurrence frequencies of a and b with the common neighbor c. In Equation 4.1, the term $(1-\alpha)^{k_n}$ derived from the first assumption and the term $\frac{1}{|t_{1_n} - t_{2_n}|+1}$ derived from the second assumption. Here, α is called the decaying factor and $(1-\alpha)$[‡] is a decaying function ($0 < \alpha < 1$). Decaying function describes the rate of decay of link activeness over time. k is the difference between current time t_c and the most recent timestamp from t_1 and t_2, and k is defined as follows:

$$k = t_c - max(t_1, t_2) \tag{4.2}$$

The number of interactions or cooccurrences, referred to as link weight, of a node pair is also important in determining the link strength. Therefore, we used the harmonic mean of the link values of each node in a node pair with their common

[‡] In our journal paper, we used parameter β which is equal to $(1-\alpha)$

neighbor. The harmonic mean, H_m, of numbers $x_1, ..., x_j$ is defined as follows:

$$H_m = \frac{1}{\frac{1}{j}\sum_{i=1}^{j}\frac{1}{x_i}} \tag{4.3}$$

Typically, the harmonic mean is appropriate for situations in which an average of rates is desired. In Equation 4.3, x_i ($i = 1, ..., j$) denote the rates. In the present case, $j = 2$ because we used the link values of each node in a node pair with their common neighbors as the rates.

In Equation 4.1, the term $(1 - \alpha)^{k_n}$ increases as k_n decreases. We assume that there is a certain decay in the influence of common neighbors with respect the current time. The decaying factor represents the decay, which has to be determine for a given network. The other assumption is that the decay $\alpha = 0$ only if the common neighbors interact with their sharing nodes at the current time. Therefore, no need to test the equation for $\alpha = 0$ because the decay doesn't count when $t_{1_n} = t_{2_n} = t_c$. At this point Equation 4.1 reduces to the summation of harmonic means of weights of links between common neighbors and its sharing nodes. In the other term, we use the reciprocal of $|t_{1_n} - t_{2_n}| + 1$, where t_{1_n} and t_{2_n} are the time stamps of the most recent interactions of the node pair with the common neighbor. This term becomes larger when the difference between t_{1_n} and t_{2_n} becomes larger. The meaning of this term is if common neighbors interacted with their common neighbors within a closer proximity of time the influence of common neighbors increase. The addition of one in the term is in order to avoid the *Time score* from becoming infinite when the two time stamps are equal.

Compiling all, the new feature *Time score* can be used as a feature, which is used for predicting future possible links. In order to show how to calculate *Time score*, let us assume that two authors, a and b, have common neighbor author c. If a and c published two papers in 2005 and 2006 and authors b and c published one paper in 2008, then the harmonic mean of two publications and one publication is obtained as follows:

$$H_m = \frac{1}{\frac{1}{2}(\frac{1}{2} + \frac{1}{1})} = 1.3333 \tag{4.4}$$

If the current year is assumed to be 2011, then the *Time score* for a future possible link between a and b can be calculated as follows:

$$TS(a,b) = \left(\frac{1.3333 * 0.5^3}{|2008 - 2006| + 1}\right) \approx 0.05555 \tag{4.5}$$

In this case, k = 2011-2008 = 3, because the latest time stamp is 2008, and the current year is 2011. The number of common neighbors, n, is 1, and we assume that $\alpha = 0.5$.

4.3 Experimental setting

In this section we discuss the experiments carried out to test the effectiveness of *Time score* for link prediction. In order to test the effectiveness of *Time score*, we performed two experiments using two real-world social network data sets. First experiment tested the link prediction performance of *Time score* by varying α from 0.1 to 0.9 [69]. The purpose of this experiment was to provide guidelines for choosing values of the decaying factor α, particularly for different time units k and different data. The second experiment, the link prediction performances of *Time score* compared using two feature combinations.

1. Baseline combination *(BC)* which includes only the existing features

2. *Time score combination (TSC)* which includes *Time score* and the existing features

we used the α values corresponding to the highest F-measure for each data set in the first experiment to compute *Time score*. Table 4.1 lists the details of the features used in each combination. The real-world networks we used for our experiments are very sparse, and so the rate of positive examples is very low. On average, the percentages of positive examples in the facebook data and the coauthorship data were 0.05% and 0.08%, respectively. In order to solve this problem, we used SMOT oversampling algorithm with default parameters [12]. After oversampling, the percentages of positive examples in the facebook data and the coauthorship data were 0.3% and 0.5%, respectively. In the experiments, we used J48

Table 4.1: Features used in Time score combination and baseline combination

Feature	Formula	Baseline combination(BC)	Time score combination(TSC)
Adamic/Adar	$\sum_{k \epsilon \Gamma(i) \bigcap \Gamma(j)} \frac{1}{log\|\Gamma(k)\|}$	✓	✓
Common neighbors	$\|\Gamma(i) \bigcap \Gamma(j)\|$	✓	✓
Jaccard's coefficient	$\frac{\|\Gamma(i) \bigcap \Gamma(j)\|}{\|\Gamma(i) \bigcup \Gamma(j)\|}$	✓	✓
Preferential attachment	$\|\Gamma(i)\|\|\Gamma(j)\|$	✓	✓
Time score	$\sum_n \frac{H_{m_n}(1-\alpha)^{k_n}}{\|t_{1_n}-t_{2_n}\|+1}$	-	✓

weka implementation [24] of C4.5 decision tree algorithm [86] as our supervised learning method. We tested the effectiveness of *T_Flow* algorithm for a data set extracted from facebook social network and coauthorship data sets extracted from *e-print archive*[§].

4.4 Experiment using facebook data

The first data set was facebook friendship network data from [103], which were collected from the regional facebook network of New Orleans. The facebook data was collected for 60,290 users who are connected by 1,545,686 links. We extracted a snapshot of the data from October 2007 to January 2009. Table 4.2 shows the statistics of the facebook network data sets used in the experiments. We created twelve network data sets from the extracted snapshot in order to use with supervised machine learning method. Supervised machine learning algorithms required training data to train the learner. Therefore, we used user interactions (wall postings) within three consecutive months to predict the links of the following month because social networks such as facebook show drastic changes within short periods of time. For example, to predict the links that emerged during January 2009, we trained the decision tree algorithm using the data from September 2008 to December 2008. Features were computed using the network data from September 2008 to November 2008, and the links that emerged during December

[§] http://arxiv.org/

2008 were considered to be the positive examples for training data. The trained model was applied for the test data features computed for the data from October 2008 to December 2008 in order to predict the links that emerged during January 2009.

At first, we analyzed the link prediction performance of *Time score* by varying α from 0.1 to 0.9. We analyzed the precision, recall, and F-measure of the predictions for facebook data set by varying α between 0.1 and 0.9. The purpose of this analysis is to provide a guideline for selecting the model parameter α according to the unit of k. The range of k depends on the time unit. For the facebook data, we used days as the time unit. Time stamps of the links are created using the time stamps of the wall postings. Timestamp of a link represents the day of the most recent wall posting between two users. Since, we used user interactions (wall postings) within three consecutive months to predict the links of the following month, the approximate range of k for the facebook data is 0 to 90 days. The time stamp for the interaction between a pair of users represents the day of the most recent wall posting between them. Number of the wall postings between users is considered to be the link wights. Figure 4.2 shows the variation of average precision, recall, and F-measure for each α value for facebook data. The average precision, recall, and F-measure increase as α increases. A notable increase occurs at $\alpha = 0.1$. We conducted a Grubbs' test to determine the significance of the difference between the F-measure at $\alpha = 0.1$ and the F-measures at $\alpha = 0.2$ to 0.9. The results of the Grubbs' test indicate that $\alpha = 0.1$ is an outlier with a significance level of 5%. This indicates that the performance of the *Time score* at $\alpha = 0.1$ is significantly higher than for other α values and is thus a good parameter for facebook data. Thus, we used *Time score* at $\alpha = 0.1$ to compare with baseline combination.

In our second experiment, we compared the link prediction performances of *Time score* using *Baseline combination (BC)* and *Time score combination (TSC)*. The performance metrics for facebook data are compared in Figure 4.3. They show a notable improvement for *Time score combination*, as compared to the *Baseline combination*. On average, the use of *Time score* increased the precision, recall, and F-measure by 4%, 3%, and 7%, respectively.

According to the wall post data shown in Figure 4.4, and as stated in [103],

Table 4.2: Statistics of the facebook data

Prediction month	Training data		Test data	
	Nodes	Edges	Nodes	Edges
2008 Feb	13,733	50,248	13,732	47,986
2008 Mar	13,732	47,986	13,998	48,238
2008 Apr	13,998	48,238	14,762	50,732
2008 May	14,762	50,732	15,705	56,014
2008 Jun	15,705	56,014	16,381	58,546
2008 Jul	16,381	58,546	17,268	60,718
2008 Aug	17,268	60,718	18,339	63,392
2008 Sep	18,339	63,392	20,476	71,792
2008 Oct	20,476	71,792	22,732	80,848
2008 Nov	22,732	80,848	25,427	92,990
2008 Dec	25,427	92,990	28,370	106,106
2009 Jan	28,370	106,106	31,832	123,650

the number of wall posts increases rapidly from July 2008 to January 2009. This makes the network more active, and most of the existing links become stronger. The stronger links have a greater influence on future link evolution. Therefore, the use of *Time score* is more effective and yields better results. This observation further indicates that *Time score* is more sensitive to the temporal behavior of user interactions. However, in February 2008 and June 2008, there is a decrease in the number of wall posts. Thus, the network becomes less active, and the strengths of the links do not exhibit temporal variations in behavior in the network during this period. Therefore, the performance metrics exhibit slightly lower values for *Time score combination* than for *Baseline combination*. Except for the results of February 2008 and June 2008, the t-test at the 5% significance level indicates significant improvements. Therefore, we can conclude that *Time score* is more effective for rapidly evolving networks.

In the facebook friendship network, the friends of a user can view the wall posts of that user if the user shares the wall posts with his/her friends. Thus, users who have that particular user as a common neighbor, while having no other relationship, can become friends through each other's postings. Burst of wall postings indicates that more people are interacting with each other and become

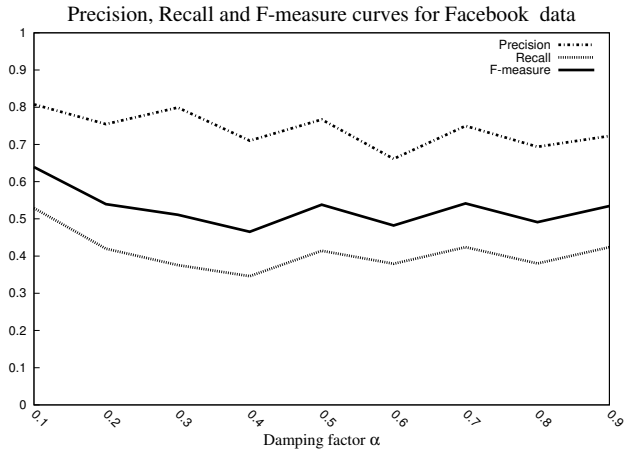


Figure 4.2: Variation of performance metrics with α for facebook data

friends. Therefore, recent interactions happen in closer proximity of time have a greater influence on link evolution. Besides the factors we investigated in our experiments, the link evolution could be depend on other temporal factors such as duration of data collection and geographical region of the network. In particular, the facebook network exhibits different patterns depending on the time, the major events that occur during the period of data collection, and the geographical region of the network. Such kind of factors are to be explored in our future works.

4.5 Experiment using coauthorship data

The second data set is a coauthorship data set extracted from 66,791 publications on condensed matter physics from 1997 to 2005 in the *cond-mat archive*[¶]. This data set contains data for 79,208 authors who are connected by 641,796 links. Table 5.4 shows the statistics of the coauthorship network data sets used in the experiments. We used data for three consecutive years to predict the links of following year. For example, in order to predict the set of links that emerged in 2010, features for the training set were calculated using the coauthorship data from 2006 to 2008, and links that emerged in the year 2009 were considered to be positive examples for training data.

At first, we analyzed the link prediction performance of *Time score* by vary-

[¶] http://arxiv.org/archive/cond-mat/

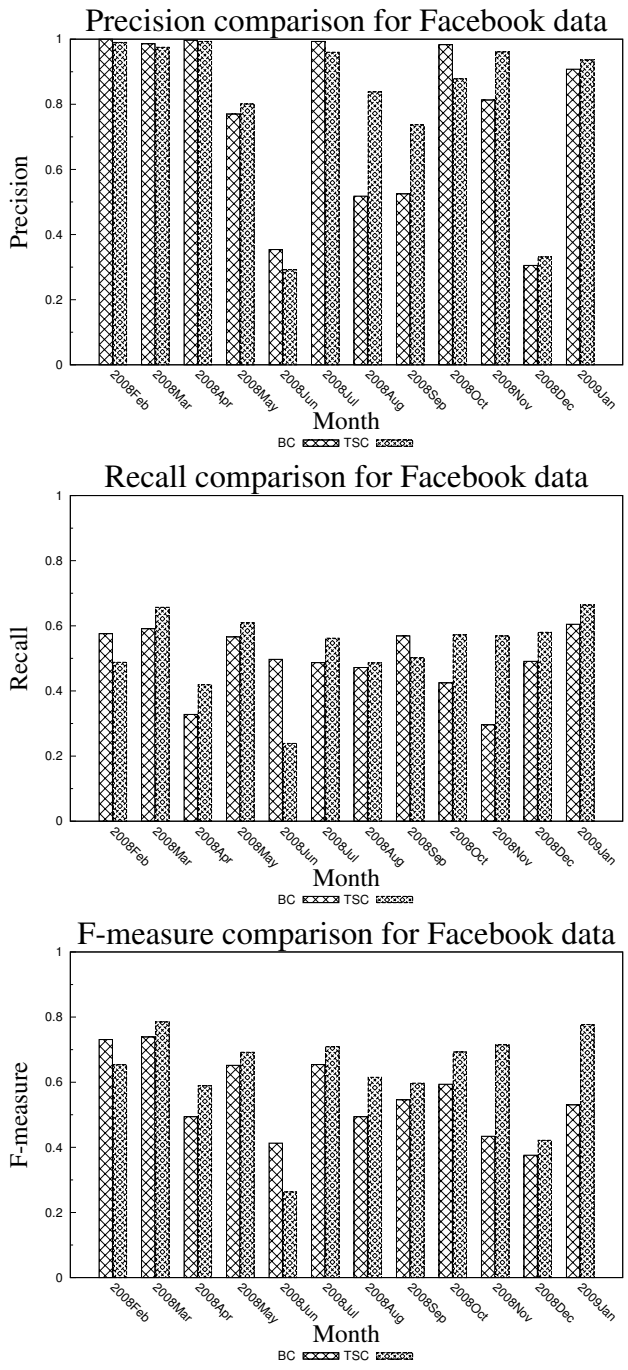


Figure 4.3: Comparison of performance metrics for facebook data

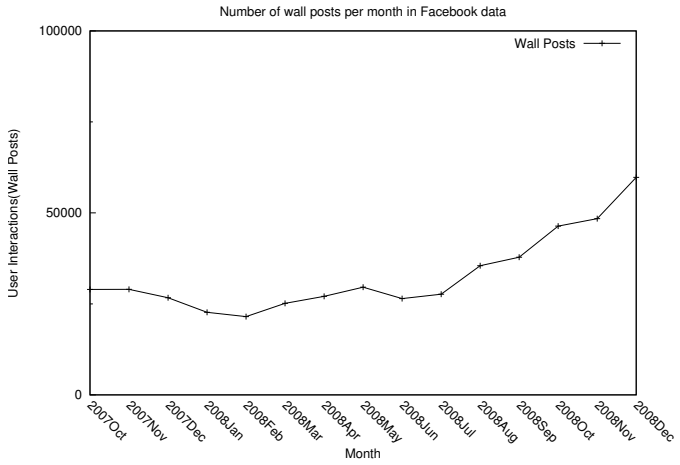


Figure 4.4: Variation of the number of wall posts in facebook data

Table 4.3: Statistics of coauthorship data

Prediction year	Training data		Test data	
	Nodes	Edges	Nodes	Edges
2001	23,411	135,798	27,349	167,180
2002	27,349	167,180	31,662	209,632
2003	31,662	209,632	34,860	237,346
2004	34,860	237,346	38,039	266,236
2005	38,039	266,236	41,213	288,796

ing α from 0.1 to 0.9. We analyzed the precision, recall, and F-measure of the predictions for coauthorship data set by varying α between 0.1 and 0.9. The purpose of this analysis is to provide a guideline for selecting the model parameter α according to the unit of k. The range of k depends on the time unit. For the coauthorship data, we used years as the time unit. Thus, the unit of k is years. The time stamp for the interaction between a pair of authors represents the most recent year of publication of the coauthored paper. Figure 4.5 shows the variation of average precision, recall, and F-measure for each α for coauthorship data. Better performance is obtained for lower α values, as indicated by the slight decrease in performance when α is greater than 0.5. We can see a clear peak of performance at $\alpha = 0.8$. Hence, we used the parameter values $\alpha = 0.8$ in this experiment.

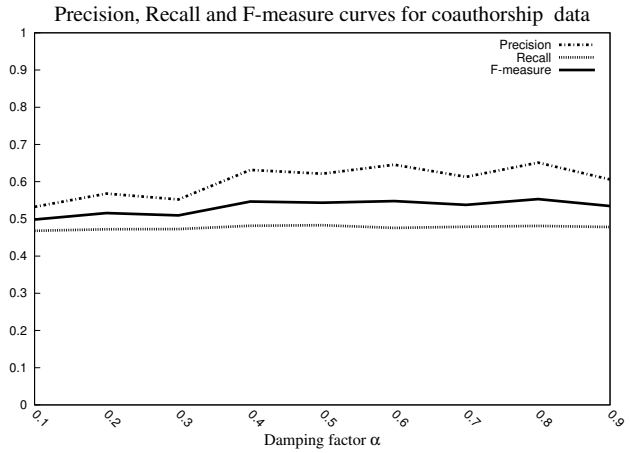


Figure 4.5: Variation of performance metrics with α for coauthorship data

In our second experiment, we compared the link prediction performances of *Time score* using *Baseline combination (BC)* and *Time score combination (TSC)*. The performance metrics of this experiment are compared in Fig. 4.6. The improvements in precision, recall, and F-measure indicate the impact of *Time score* for link prediction in the coauthorship network that evolves primarily over recent collaborations. In the graph comparing precision, with the exception of 2001, the results obtained using *Time score combination* are better than the results obtained using *Baseline combination*. All three performance metrics indicate significant improvements according to the t-test at the 5% significance level. The average improvements in precision, recall, and F-measure are 14%, 11%, and 13%, respectively.

4.6 Discussion

In this research, we introduced an effective novel time-aware feature ,*Time score*, defined on common neighbors. We tested it with two real-world data sets in order to clarify its effectiveness. The results show that *Time score* is significantly effective in rapidly changing networks. We used timestamps of the interactions/links to compute *Time score*. Unit of the time depends on the network and the nature of the interactions we considered. Therefore, the unit of k in *Time score* formula can

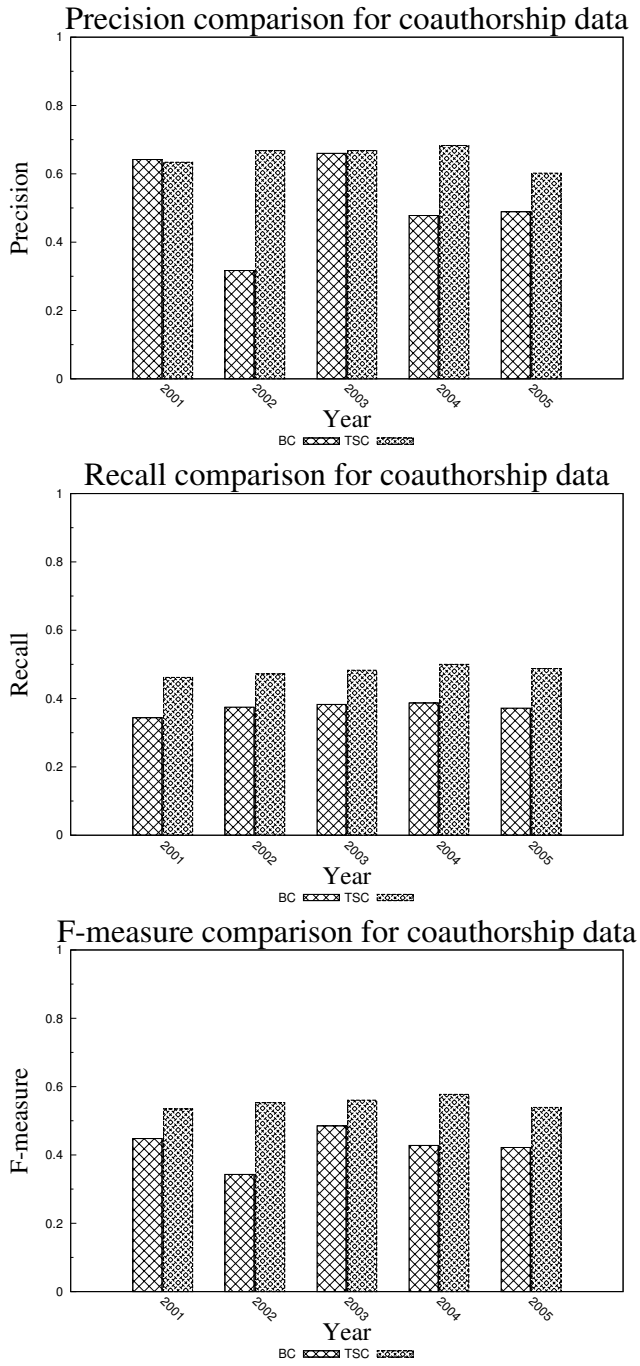


Figure 4.6: Comparison of performance metrics of coauthorship data

be years, days, or hours, depending on the data set. We need to set α according to the unit of k in order assign higher scores to interactions that have occurred more recently. When k increases, $(1-\alpha)^k$ decreases. For higher values of k and higher values of α, the term $(1-\alpha)^k$ is approximately 0. For example, when $k = 10$ and $\alpha = 0.5$, $(1-\alpha)^k$ is approximately 0.00098. In order to obtain a meaningful value for $(1-\alpha)^k$ when k has a wide range, it is better to use a lower α. In a network such as facebook, interactions that occurred ten days ago have more effect on future link evolution than interactions that occurred ten years ago in the coauthorship network. Therefore, lower α values are better when k has a wide range.

The range of k is small ($0 \leq k \leq 2$) for the coauthorship data because we used the data of three consecutive years to predict links in the following year. The term $(1-\alpha)^k$ can take a higher value, even for higher α. Since the range of k is small for coauthorship data, the term $(1-\alpha)^k$ takes closer values for lower α values, and *Time score* becomes less effective for the learning algorithm. Therefore, higher α values are more appropriate for computing *Time score* when k has a small range.

4.7 Conclusion

The past research has been introduced many neighbor-base features for link prediction. However, majority of them are static which means that they are unable cope with the rapidly changing nature of the neighbors. Hence, we introduced a significantly effective time-aware feature, *Time score*, for link prediction in rapidly changing social networks. The prominent feature of *Time score* is it incorporates relationship between the temporal behavior of link activeness and the time stamps of interactions/links for link evolution, which has not previously been discussed sufficiently. We examined link prediction performances of *Time score* using supervised machine learning methods. We tested it on two real-world data sets namely, facebook and coauthorship network data sets. The significant improvements of the experimental results verify the effectiveness of *Time score* for link prediction in highly dynamic social networks.

Chapter 5

T_Flow algorithm

Summary: *This chapter presents a novel algorithm for compute information flow via active links. Information flow heavily depends on link activeness. Links become active if interactions happen frequently and recently with respect to the current time. Time stamps of interactions or links provide vital information for determining the activeness of the links. Thus, we introduced a novel algorithm, referred to as T_Flow, that captures the important aspects of information flow via active links in social networks. Once we computed information flow using T_Flow, it is used as a feature with machine learning methods for link prediction. We tested T_Flow with two social network data sets, namely, a data set extracted from facebook friendship network and a coauthorship network data set extracted from ePrint archives. We compare the link prediction performances of T_Flow with the previous version PropFlow. The results of T_Flow algorithm revealed a notable improvement in link prediction for facebook data and significant improvement in link prediction for coauthorship data. Further, we compared T_Flow with Time score in terms of recall.*

5.1 Information flow for link prediction

Social networks are of interest to researchers in part because they are thought to mediate the flow of information in communities and organizations. Information flow or information propagation in social networks has been explored many re-

searchers [10]. Particularly, the information flow is highly correlated with the link evolution. Kossinets et. al. carried out a study for two-years period to understand the temporal dynamics of communication using on-line data, including e-mail communication among the faculty and staff of a large university [43]. They formulated a temporal notion of "distance" in the underlying social network by measuring the minimum time required for information to spread from one node to another concept that draws on the notion of vector-clocks from the study of distributed computing systems. They reported that such temporal measures provide structural insights that are not apparent from analyses of the pure social network topology. In particular, they have defined the network backbone to be the subgraph consisting of edges on which information has the potential to flow the quickest. they found the backbone is a sparse graph with a concentration of both highly embedded edges and long-range bridges, which sheds new light on the relationship between link strength and connectivity in social networks.

In the previous chapter, we introduced a novel feature called *Time score* defined based on link activeness, which showed an impressive link prediction performances. The fundamental assumption here is if the interactions happen frequently and recently the links become active and influence other nodes to become linked. However, *Time score* is limited to common neighbors. Then how do we extend this idea to any node pair for link prediction?. Therefore, we investigated the possible ways to extending the idea of *Time score* to a any node pair. We identified one possible way to integrate the idea of *Time score* to compute information flow between any node pair. In other words, if information flow between nodes happens regularly the links become active and it influence the evolution of new links. We learnt that timestamps provide the all necessary information in determining the activeness of links. Some of the recent link prediction research have introduced supervised/unsupervised random walk algorithms to compute information flow in social networks. Backstrom et. al. introduced a supervised random walk algorithm which use edge strengths as transition probabilities [5]. The edges strengths are learnt using network structure, node and edge attributes. Lichtenwalter et. al. introduced a random walk based algorithm called *PropFlow* to compute information flow [57]. *PropFlow* algorithm uses link weights are the transition probabilities. The common idea of both approaches is if a node pair has higher

transition probability, more information flow happens between the node pair and they are is more likely to get linked in the future. Although these methods effectively combine node and edge level attributes as well as network structure, link activeness still a missing part of these methods. We therefore, introduced *T_Flow* algorithm which computes the information flow between any pair of nodes in a social network by considering the link activeness [71]. Once we compute it, we used it as a feature for link prediction using supervised machine learning methods. *T_Flow* algorithm is an extension of the *PropFlow* algorithm. In the next section we discuss *PropFlow* algorithm in detail.

5.1.1 PropFlow algorithm

Information flow between nodes is a vital factor for link evolution in social networks and it depends very much on link attributes such as link weights and activeness. The *PropFlow* algorithm computes information flow based on random walk method which select its path based on link weights. This method is somewhat similar to rooted page rank, but restricted to local neighborhood of a node. Unlike rooted page rank, the random walker doesn't need to restart or convergence and use modified breadth first search restricted to depth l. The random walker starts from a particular node and reach the desired node in l steps or fewer. Revisiting any node including starting node is not allowed for the random walker. *PropFlow* algorithm computes the information flow called *PropFlow* for a pair of nodes i and j based on the random walks between them. The Equation 5.1 shows how to compute $PropFlow(i, j)$ if nodes i and j directly linked. In this case, random walker starts from node i and walk to node j.

$$PropFlow(i, j) = NodeInput_i * \frac{w_{ij}}{\sum_{k \in N(i)} w_{ik}} \tag{5.1}$$

Where, w_{ij} denotes the weight of the link between nodes i and j. k denotes a node and set $N(i)$ denotes node i's neighbors whose depth is greater than the depth of node of i from the starting node. Initial node input is regarded as 1. If nodes i and j are indirectly linked, *PropFlow* algorithm computes the information flow through all the shortest paths from node i to node j using Equation 5.1 recursively

and take the summation.

For example, the Equations 5.2 to 5.7 show how to compute the $PropFlow(A, D)$ between nodes A and D in the coauthorship network shown in Figure 5.1. Link weights are denoted by p. We assumed the random walker starts from node A. $PropFlow(A, D)$ is computed using link weights of links AB, BC, CD, BE, ED. There are four paths the random walker can reach node D from node A. They are A→B→C→D, A→B→E→D, A→B→E→C→D, and A→B→C→E→D. We have to note that *PropFlow* algorithm use modified breadth-first search method and it stops when revisiting any node. Thus, random walker doesn't revisit node C from node E. Therefore, the paths A→B→E→C→D and A→B→C→E→D have not considered for computations. First, we have to compute $PropFlow(A, B)$. Weight of the link between A and B is 3. The sum of the link weights of links between A and its neighbors is 4. Note that initial node input of A is considered as 1. $PropFlow(A, B)$ can be compute as;

$$PropFlow(A, B) = 1 * \frac{3}{(1+3)} = 1 * \frac{3}{4} = \frac{3}{4} \tag{5.2}$$

$PropFlow(B, C)$ can be compute as;

$$\begin{aligned} PropFlow(B, C) &= PropFlow(A, B) * \frac{1}{(1+1+2)} \\ &= \frac{3}{4} * \frac{1}{4} = \frac{3}{16} \end{aligned} \tag{5.3}$$

$PropFlow(B, E)$ can be compute as;

$$\begin{aligned} PropFlow(B, E) &= PropFlow(A, B) * \frac{1}{(1+1+2)} \\ &= \frac{3}{4} * \frac{1}{4} = \frac{3}{16} \end{aligned} \tag{5.4}$$

$PropFlow(C, D)$ can be compute as;

$$\begin{aligned} PropFlow(C, D) &= PropFlow(B, C) * \frac{5}{5} \\ &= \frac{3}{16} * 1 = \frac{3}{16} \end{aligned} \tag{5.5}$$

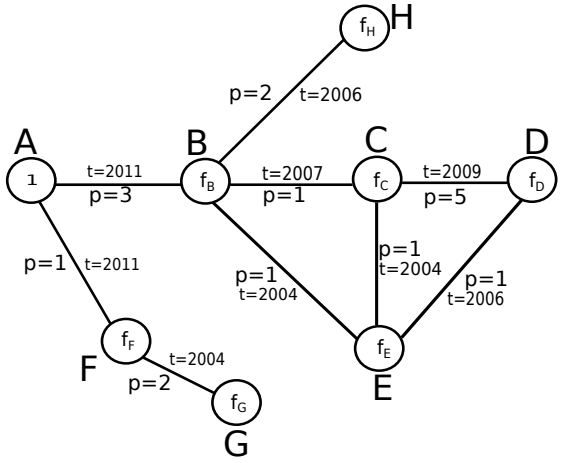


Figure 5.1: An example of a coauthorship network

$PropFlow(E, D)$ can be compute as;

$$\begin{aligned} PropFlow(E, D) &= PropFlow(B, E) * \frac{1}{1} \\ &= \frac{3}{16} * 1 = \frac{3}{16} \end{aligned} \tag{5.6}$$

Therefore, the $PropFlow(A, D)$ is;

$$PropFlow(A, D) = \frac{3}{16} + \frac{3}{16} = \frac{6}{16} = \frac{3}{8} \tag{5.7}$$

Although *PropFlow* algorithm computes the information flow in social networks using link weights, the information flow doesn't depend only on the link weights. The activeness of the links is a vital factor for information flow. The links become weak or deactivate if nodes haven't interacted recently with respect to the current time. Despite of their weights, the weakened or deactivated links can cause a decay in information flow. We therefore, introduced an extension of *PropFlow* algorithm referred to as *T_Flow* algorithm [71] in order to consider the effect of active links for information flow.

5.2 T_Flow algorithm

The time stamps of the links or interactions are useful in determining the activeness of the links. If a node pair interact recently the link between them become active. In other words, the time stamp of the last interaction is a vital information in deciding the activeness of a link. Hence, we used the most recent time stamps of the interactions between nodes for our computations. Time stamp can be the most recent hour, day or year of a interaction between a node pair. The time unit of the time stamps depends on the network. *T_Flow* algorithm use the same settings as in *PropFlow* algorithm for random walk. It considers link weight as well as link activeness to compute transition probabilities. The relationship between information flow and the activeness of links is defined by a decaying function. The empirical studies have revealed that the decay in social network links is a power function of time [7]. Therefore, we assumed the decay of information flow as a function of decaying factor α ($0 \leq \alpha < 1$) and difference of time stamps of adjacent links. The decaying function $d(i, j)$ for information flow from node i to its adjacent node j is defined as;

$$d(i, j) = (1 - \alpha)^{|t_x - t_y|} \tag{5.8}$$

The decaying factor α ($0 < \alpha < 1$) is the rate of decay per unit time of the information flow and t_x is the time stamp of the link which random walker comes into the node i and t_y is the time stamp of the link which random walker going to node j. The value of decaying function become 1 when $\alpha = 0$ which means no decay in information flow. At this point *T_Flow* algorithm is identical to its previous version *PropFlow* algorithm. The *T_Flow* algorithm computes information flow from node i to j via direct link as follows;

$$T\_Flow(i, j) = NodeInput_i * \frac{w_{ij}}{\sum\limits_{k \in N(i)} w_{ik}} * (1 - \alpha)^{|t_x - t_y|} \tag{5.9}$$

If nodes i and j are indirectly linked, *T_Flow* algorithm computes the information flow through all the shortest paths from node i to node j using Equation 5.9 re-

cursively and take the summation. The total flow between two nodes regarded as the *T_Flow* for the node pair. At the start of the random walk, t_x is regarded as the current time and the initial node input is considered as 1. We have listed the *T_Flow* algorithm in Algorithm 1.

For example, the Equations 5.10 to 5.15 show how to compute $T\_Flow(A, D)$ between nodes A and D in Figure 5.1. Time stamps of the links denoted by t in Figure 5.1. We assumed the random walker starts from node A and the current time is the year 2012. $T\_Flow(A, D)$ is computed using link weights of links AB, BC, CD, BE, ED and their time stamps. First, we have to compute $T\_Flow(A, B)$.

$$\begin{aligned} T\_Flow(A, B) &= 1 * \frac{3}{(1+3)} * (1-\alpha)^{|2012-2011|} \\ &= \frac{3}{4} * (1-\alpha)^1 = \frac{3}{4} * (1-\alpha) \end{aligned} \tag{5.10}$$

The link BC has the time stamp (2007) and the link BE has the time stamp (2004). Therefore, BC is the most active link. Thus, more information should flow through BC than BE which has the same weight as BC but less active than BC. $T\_Flow(B, C)$ can be compute as;

$$\begin{aligned} T\_Flow(B, C) = T\_Flow(A, B) * \frac{1}{(2+1+1)} * (1-\alpha)^{|2011-2007|} \\ = \frac{3}{4} * (1-\alpha) * \frac{1}{4} * (1-\alpha)^4 \\ = \frac{3}{16} * (1-\alpha)^5 \end{aligned} \tag{5.11}$$

$T\_Flow(B, E)$ can be compute as;

$$\begin{aligned} T\_Flow(B, E) = T\_Flow(A, B) * \frac{1}{(2+1+1)} * (1-\alpha)^{|2011-2004|} \\ = \frac{3}{4} * (1-\alpha) * \frac{1}{4} * (1-\alpha)^7 \\ = \frac{3}{16} * (1-\alpha)^8 \end{aligned} \tag{5.12}$$

$T\_Flow(C, D)$ can be compute as;

$$T\_Flow(C, D) = T\_Flow(B, C) * \frac{5}{5} * (1 - \alpha)^{|2007-2009|}$$
$$= \frac{3}{16} * (1 - \alpha)^7 \tag{5.13}$$

$T\_Flow(E, D)$ can be compute as;

$$T\_Flow(E, D) = T\_Flow(B, E) * \frac{1}{1} * (1 - \alpha)^{|2004-2006|}$$
$$= \frac{3}{16} * (1 - \alpha)^{10} \tag{5.14}$$

Therefore, the $T\_Flow(A, D)$ is;

$$T\_Flow(A, D) = \frac{3}{16} * (1 - \alpha)^7 + \frac{3}{16} * (1 - \alpha)^{10} \tag{5.15}$$

5.3 Experimental settings

At first, we analyzed effectiveness of *T_Flow* algorithm for link prediction by varying α from 0 to 0.9. Then, the link prediction performances of *PropFlow* algorithm and *T_Flow* algorithm were compared using three feature combinations.

1. Baseline combination *(BC)* which include neither *PropFlow* algorithm nor *T_Flow* algorithm

2. *PropFlow combination (PFC)* which includes *PropFlow* algorithm

3. *T_Flow combination (TFC)* which includes *T_Flow* algorithm

For the comparison, we conducted the experiments for *T_Flow combination* using two-loop cross validation where the inner loop determines α and the outer-loop evaluates prediction. Training data in the outer loop is used in the inner loop to find the optimal parameter value which is then used to evaluate the test data in the outer loop. The feature combinations used in the experiments are shown in Table 5.1. In our experiments, J48 Weka implementation [24] of C4.5 decision

Algorithm 1: T_Flow Algorithm

Input: network $G = (V, E)$, start node s, depth l, decaying factor α, current time t_c

Output: T_Flow T_f for all neighbors of s within depth l

begin
- insert s into $Visitedset$
- push s into $NewSearchqueue$
- push t_c into $Timequeue$
- insert $(s,1)$ into T_f
- $OldSearchqueue \longleftarrow empty$
- **for** $Distance \leftarrow 0$ *to* l **do**
  - $OldSearchqueue \longleftarrow NewSearchqueue$
  - *empty* $NewSearchqueue$
  - **while** $OldSearchqueue$ *is not empty* **do**
    - pop i from $OldSearchqueue$
    - pop t_x from $Timequeue$
    - find $NodeInput$ using i in T_f
    - $t_y \leftarrow 0$
    - $SumWeight \leftarrow 0$
    - $Flow \leftarrow 0$
    - **for** j *in neighborhood of* i **do**
      - **if** *depth of* $j >$ *depth of* i **then**
        - add weight of edge between i and j to $SumWeight$
      - **end**
    - **end**
    - **for** j *in neighborhood of* i **do**
      - **if** *depth of* $j >$ *depth of* i **then**
        - $w_{ij} \leftarrow$ weight of edge between i and j
        - $t_y \leftarrow$ time stamp of edge between i and j
        - $Flow \leftarrow NodeInput * \frac{w_{ij}}{SumWeight} * (1-\alpha)^{|t_x - t_y|}$
        - add $(j, Flow)$ into T_f
        - **if** j *is not in* $Visitedset$ **then**
          - insert j into $Visitedset$
          - push j into $NewSearchqueue$
          - push t_y into $Timequeue$
        - **end**
      - **end**
    - **end**
  - **end**
- **end**

end

Table 5.1: Features used in PropFlow combination and T_Flow combination

Feature	**Formula**	**BC**	**PFC**	**TFC**
Adamic/Adar	$\sum_{k \epsilon \Gamma(i) \bigcap \Gamma(j)} \frac{1}{log\|\Gamma(k)\|}$	✓	✓	✓
Common neighbors	$\|\Gamma(i) \bigcap \Gamma(j)\|$	✓	✓	✓
Jaccard's coefficient	$\frac{\|\Gamma(i) \bigcap \Gamma(j)\|}{\|\Gamma(i) \bigcup \Gamma(j)\|}$	✓	✓	✓
Preferential attachment	$\|\Gamma(i)\|\|\Gamma(j)\|$	✓	✓	✓
PropFlow			✓	-
T_Flow			-	✓

tree algorithm [86] was used with 10-fold cross validation. Machine learning approaches need *1) Training data* to train the learning method, and *2) Test data* to test the learned models. In practical terms, two-thirds of the data for training and one-third for test. In some cases the amount of data may not be sufficient for this holdout method of training and test. The other issue is examples of a certain class could missing in the training data. Such kind of situations may lead to learn inappropriate models by the learning methods. A simple statistical method called *cross-validation* is a promising technique to solve this problem. In cross-validation, we partition the data set in to a known number of partitions called *folds*. Then holdout one fold as test and remaining as the training sets. Repeating this procedure for all folds ensures that every instance in the data has been used exactly once for testing. Tenfold cross-validation is the standard way of evaluation in supervised learning. In tenfold cross-validation we split data in to ten portions and repeat the model evaluation ten times by holding one portion at a time as the test set. The experiments presented in this section used ten-fold cross-validation method for model evaluation. All network data sets are very sparse and hence SMOT oversampling algorithm [12] was used in order to deal with class imbalance problem. Precision, recall and F-measure are used as performance metrics in the experiments. In both *PropFlow* and *T_Flow* algorithms, the depth l is set to 3 which means we excluded the nodes that are more than three links away from a node. We tested the effectiveness of *T_Flow* algorithm for a data set extracted from facebook social network and coauthorship data sets extracted from *e-print*

Table 5.2: Statistics of facebook data

Training data	Nodes	Edges	Clustering coefficient	Mean degree
D1	7,094	13,294	0.0270	1.87
D2	12,862	29,656	0.0292	2.30
D3	9,310	18,138	0.0277	1.94
D4	14,405	30,142	0.0242	2.09
D5	19,614	51,030	0.0319	2.60
D6	17,277	36,414	0.0300	2.10

archive[*].

5.4 Experiment with facebook data

The facebook data set is a set of wall postings collected from the regional facebook network of New Orleans from September, 2006 to January, 2009 [103]. This data set consist of wall postings exchanged by 60,290 users who are connected by 1,545,686 links. We extracted six different snapshots of data from May, 2008 to December, 2008 which shows a rapid increase of wall postings. Wall postings are considered as the interactions between users. Each data set consist of wall postings of three weeks. Link weight represents the number of wall postings exchanged between a pair of users. The day of the most recent wall posting represents the time stamp of a link.

We train the decision tree algorithm for facebook data using wall postings in two consecutive weeks to predict links in the following week. The statistics of the facebook training data are shown in Table 5.2. The unit of time for facebook data is days. The experiment was conducted for six data sets and the average performance of *T_Flow* algorithm was computed.

Link prediction performance of *T_Flow combination* with the variation of α for facebook data is shown in the Figure 5.2. Average recall and average F-measure shows peaks at $\alpha = 0.1$ and $\alpha = 0.3$ and then decrease as α increase from 0.3 to 0.9 while the average precision doesn't show any drastic changes. We

[*] http://arxiv.org/

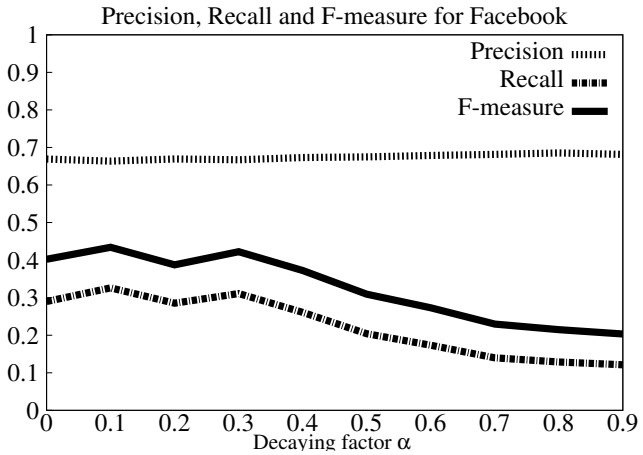


Figure 5.2: Performance of *T_Flow combination* for different α values (facebook data)

obtained the highest average F-measure for *T_Flow combination* at $\alpha = 0.1$. The decaying factor α measures the decay of influence of wall posting exchange per unit time on information flow. The links become more active if users exchange wall postings frequently and recently. Hence, the information flow decays with the time if users don't exchange wall postings frequently and recently. The facebook network grow rapidly over time and the interactions happen within a quick time. As a consequence, the decay in information flow per unit time(per day) proportionately small. In other words, if a wall posting does not exchange within a day the decay of information flow is proportionately low. Hence, the results are better for the smaller α values(smaller decay). As shown in Table 5.2, the clustering coefficients and mean degrees of the data is fairly small. It implies that the users interact with less number of friends and only few links are active during the particular time period.

Table 5.3 shows the comparison of *PropFlow combination* and *T_Flow combination* for facebook data. We tested *T_Flow combination* using two-loop cross validation method for determining α and 10-fold cross validation for computing the results. The results shows that average F-measure of *T_Flow combination* is better than the average F-measure of *PropFlow combination* which implies that the information flow via active links is a vital factor for link prediction.

Table 5.3: Comparison of *PropFlow combination* and *T_Flow combination* for facebook data

Feature combination	Avg. Precision	Avg. Recall	Avg. F-measure
BC	**0.8368**	0.1238	0.2132
PFC	0.6692	0.2898	0.4023
TFC	0.6658	**0.3327**	**0.4412**

5.5 Experiment with coauthorship data

The coauthorship data sets extracted from *e-print archive* within ten years period of publications on subject areas Astro physics (Astro-ph), Condensed matter physics (Condmat-ph), High energy physics(theory) (Hep-th) and High energy physics(phenomenology) (Hep-ph) from 1992 to 2002. We created six data sets for each subject area and computed the average performance of *T_Flow* algorithm. We have shown the statistics of each coauthorship network in the Table 5.4. Publications are considered as the interactions between authors and the year of the most recent publication represents the time stamp of a link. Link weights were computed using method introduced in [78] which is explained here. Let i and j are two authors and δ_i^k and δ_j^k are indicator functions. If author i is an author of paper k then $\delta_i^k = 1$ and zero otherwise. If paper k has n_k authors, the weight of collaboration w_{ij} between two authors i and j is computed as the summation of all coauthored papers;

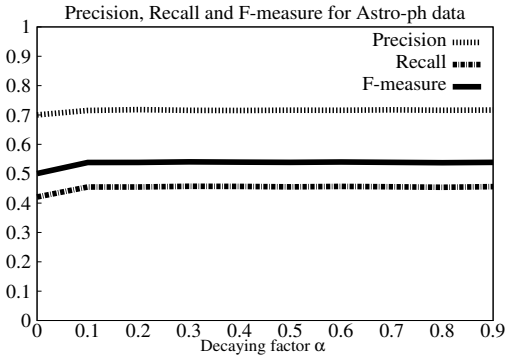
$$w_{ij} = \sum_k \frac{\delta_i^k \delta_j^k}{n_k - 1} \tag{5.16}$$

We train the decision tree algorithm using five consecutive years of coauthor data to predict links in the following year. For example, data from 1992 to 1996 is used as training data to predict links emerged in the year 1997. The unit of time for the coauthorship data is years.

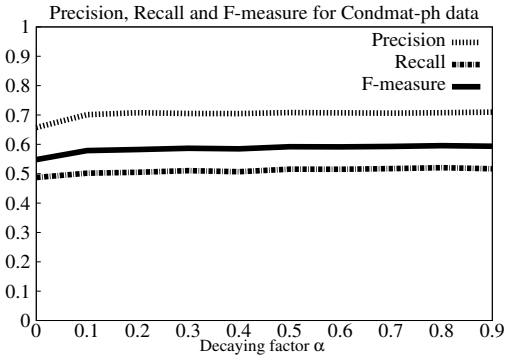
Link prediction performance of *T_Flow combination* with the variation of α for each coauthorship data is shown in Figure 5.3. We obtained the highest average F-measures at different α values for different subject areas. The activeness of links in coauthorship networks are not change rapidly as authors work together

Table 5.4: Statistics of coauthorship data

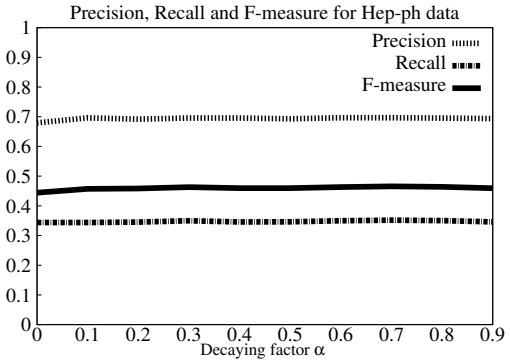
Data set (Subject area)	Training data	Nodes	Edges	Clustering coefficient	Mean degree
Astro-ph	D1(1992-1996)	8,098	53,086	0.6974	6.55
	D2(1993-1997)	12,647	113,924	0.7092	9.00
	D3(1994-1998)	17,346	177,390	0.7062	10.22
	D4(1995-1999)	22,180	261,724	0.7042	11.80
	D5(1996-2000)	27,067	358,794	0.7031	13.25
	D6(1997-2001)	31,526	455,670	0.6992	14.45
Condmat-ph	D1(1992-1996)	8,798	35,288	0.6269	4.01
	D2(1993-1997)	14,197	67,120	0.6702	4.73
	D3(1994-1998)	20,410	108,926	0.6965	5.33
	D4(1995-1999)	27,053	157,530	0.7139	5.82
	D5(1996-2000)	33,461	209,852	0.7229	6.27
	D6(1997-2001)	40786	278,152	0.7336	6.81
Hep-ph	D1(1992-1996)	9,029	56,108	0.5879	6.21
	D2(1993-1997)	10,670	71,328	0.6004	6.68
	D3(1994-1998)	12,230	88,644	0.6082	7.24
	D4(1995-1999)	13,189	98,494	0.6095	7.46
	D5(1996-2000)	14,325	136,754	0.6237	9.54
	D6(1997-2001)	15,259	139,362	0.6315	9.13
Hep-th	D1(1992-1996)	8,438	24,904	0.4904	2.95
	D2(1993-1997)	9,459	29,286	0.4976	3.09
	D3(1994-1998)	10,242	33,026	0.5094	3.22
	D4(1995-1999)	10,543	35,322	0.5164	3.35
	D5(1996-2000)	11,001	38,648	0.5146	3.51
	D6(1997-2001)	11,392	41,212	0.5162	3.61



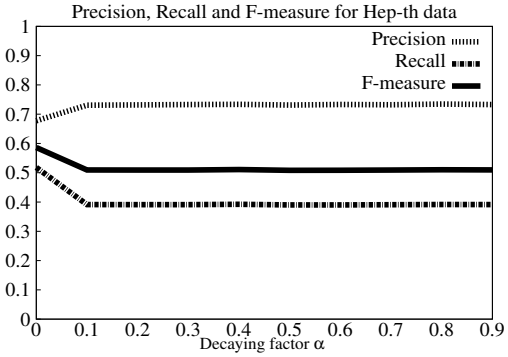
(a) Astro-ph, highest F-measure at $\alpha = 0.3$



(b) Condmat-ph, highest F-measure at $\alpha = 0.8$



(c) Hep-ph, highest F-measure at $\alpha = 0.7$



(d) Hep-th, highest F-measure at $\alpha = 0.6$

Figure 5.3: Variation of F-measure with decaying factor α(coauthorship data)

for long time to publish research papers. Therefore, the influence of coauthorship on link activeness is proportionately high. The other notable characteristic is that Astro-ph, Condmat-ph and Hep-ph coauthorship networks have high clustering coefficients and mean degrees as shown in Table 5.4. Higher clustering coefficients and mean degrees in the recent years tells that authors tends to interact(via publications) with more coauthors as networks grow with the time. More interactions makes networks more active and *T_Flow* combination perform better than *PropFlow* combination.

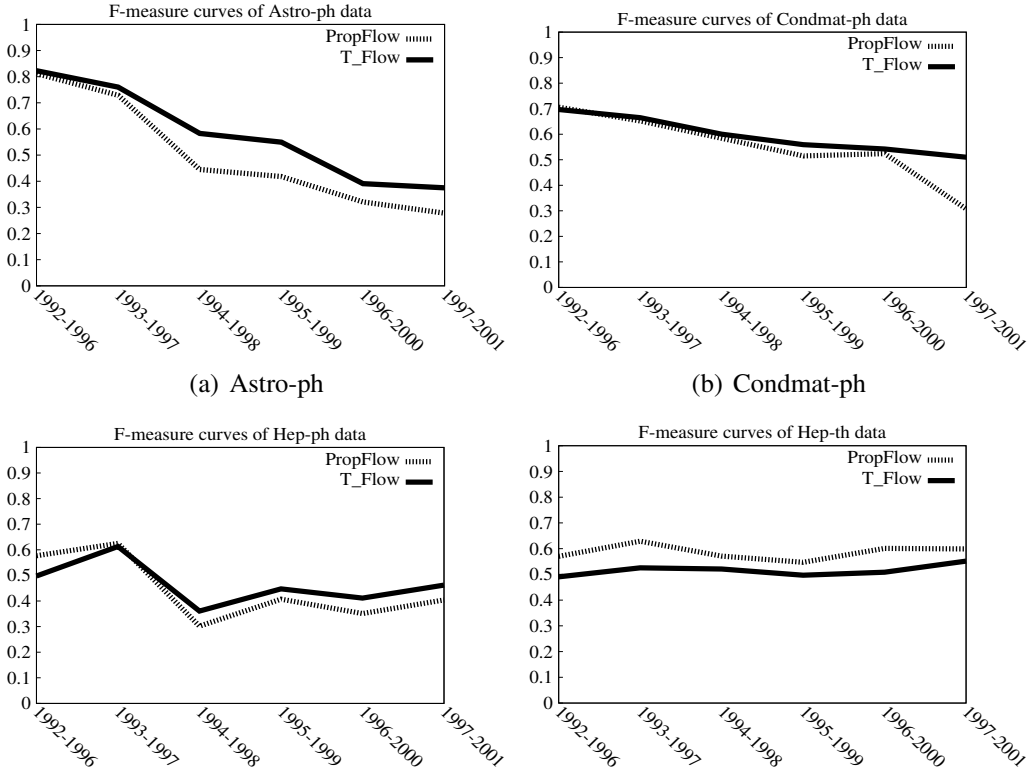
In contrast, *PropFlow* combination performs significantly better than *T_Flow* combination for Hep-th data as shown in Figure 5.3(d). As shown in Table 5.4, Hep-th coauthorship network has low clustering coefficients and low mean degrees. This observation tells that this network is less active compared to the other subject areas. In other words, the authors rarely make new coauthorships. This

Table 5.5: Comparison of *PropFlow* combination and *T_Flow* combination for coauthorship data

Data set (Subject area)	Feature combination	Avg. Precision	Avg. Recall	Avg. F-measure
Astro-ph	*BC*	0.7128	0.4367	0.5263
	PFC	0.7003	0.4208	0.5005
	TFC	**0.7394**	**0.5005**	**0.5802**
Condmat-ph	*BC*	0.6617	0.4485	0.5227
	PFC	0.6573	0.4872	0.5480
	TFC	**0.7095**	**0.5208**	**0.5960**
Hep-ph	*BC*	**0.7480**	0.3162	0.4400
	PFC	0.6795	0.3438	0.4443
	TFC	0.6963	**0.3525**	**0.4654**
Hep-th	*BC*	**0.7987**	0.3875	0.5208
	PFC	0.6775	**0.5180**	**0.5862**
	TFC	0.7381	0.3973	0.5157

phenomenon could specific to the network. In our experiments, we have assumed that the average time taken for a publication is one year. However, it takes more than one year in some research areas to make a publication. In such kind of situations, we have to choose the time unit depending on the interaction time.

The summary of results is shown in Table 5.5 with comparison of *PropFlow combination* and *T_Flow combination* for coauthorship data. We tested *T_Flow combination* using two-loop cross validation method for determining α and 10-fold cross validation for computing the results. The results shows that average F-measure of *T_Flow combination* is better than the average F-measure of *PropFlow combination*. In fact, *T_Flow* combination shows significant improvement in average F-measure for Astro-ph data. The results implies that the information flow via active links is a vital factor for link prediction. In our further analysis, we observed that the difference between F-measure values of *PropFlow* and *T_Flow* combinations increase for recent coauthorship networks as shown in Figure 5.4. In other words, *T_Flow* combination shows better performances on recent data sets which has higher clustering coefficient and mean degrees. Further, we obtained the highest F-measure for Condmat-ph data in the experimental results shown



(a) Astro-ph (b) Condmat-ph

(c) Hep-ph (d) Hep-th

Figure 5.4: Variation of F-measure with network growth(coauthorship data)

in Table 5.5. This means that the decay of information flow per unit time in Condmat-ph data is higher than the other subject areas. Such kind of data are appropriate to study the correlation between dynamic behavior of networks(network growth) and performance of *T_Flow* algorithm.

5.6 Experiments with a rapidly growing network

We carried out further experiments to investigate the performance of *T_Flow* algorithm when networks change rapidly. The coauthorship data shows drastic rise of links and nodes in the recent years. It implies that many interactions happen in the recent past. Thus, we used six different network data sets extracted from Condmat-ph publications from 1997 to 2007. Statistics of the data sets are shown in Table 5.6 and experimental settings are the same as in the Section 5.5.

Table 5.6: Statistics of Condmat-ph data

Training data	Nodes	Edges	Clustering coefficient	Mean degree
D1(1997-2001)	40,786	278,152	0.7336	6.81
D2(1998-2002)	46,124	328,432	0.7348	7.11
D3(1999-2003)	50,632	373,934	0.7347	7.38
D4(2000-2004)	55,425	424,116	0.7349	7.65
D5(2001-2005)	59,742	467,608	0.7357	7.82
D6(2002-2006)	62,802	493,634	0.7367	7.86

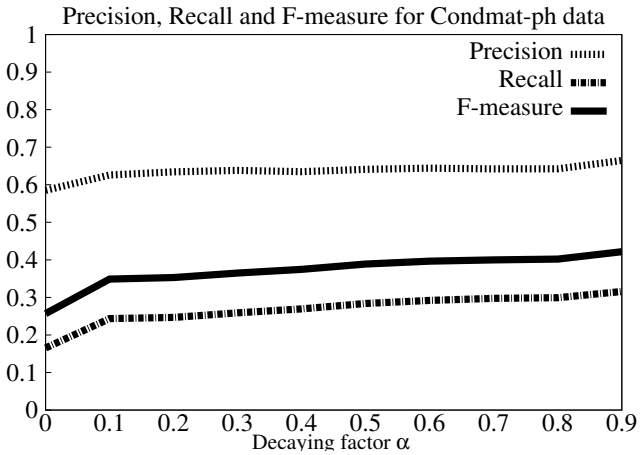


Figure 5.5: Performance of *T_Flow combination* for different α values (Condmat-ph)

Clustering coefficients and mean degrees are almost same for six data sets. Link prediction performance of *T_Flow combination* with the variation of α is shown in Figure 5.5. Comparison of *PropFlow* combination with *T_Flow* combination is shown in Table 5.7. We tested *T_Flow combination* using two-loop cross validation method for determining α and 10-fold cross validation for computing the results. The results shows a significant improvement for *T_Flow* combination. It implies that *T_Flow* algorithm is more sensitive for rapid changes in link activeness and hence, shows better performance for dynamic networks.

Table 5.7: Comparison of *PropFlow combination* and *T_Flow combination* for Condmat-ph data

Feature combination	Avg. Precision	Avg. Recall	Avg. F-measure
PFC	0.5852	0.1655	0.2567
TFC	**0.6637**	**0.3258**	**0.4302**

5.7 Comparison of Time score and T_Flow

Time score is limited to node pairs which have common neighbors. In other words, it is restricted the node neighborhood to 2-hops (two links). In contrast, *T_Flow* is applicable for any node pair. However, we limited the node neighborhood for 3-hops for *T_Flow* to reduces the computational cost. In reality, new links emerge between nodes which resides more than 2-hops or 3-hops apart. Therefore, it is necessary to know the amount (percentage) of links can be predict by *Time score* and *T_Flow* with respect to the whole network. In other words, the percentage recall of *Time score combination(TSC)* and *T_Flow combination(TFC)* is useful and necessary: (1) to know the coverage of links predicted by each feature compared to the whole network, (2) to compare *Time score* and *T_Flow* in terms of percentage recall. Therefore, we computed recall for *TSC* and *TFC* with respect to whole networks of facebook data and Condensed matter physics (Condmat-ph) data.

In the experiments, we used J48 weka implementation of C4.5 decision tree algorithm as our supervised learning method. First, we apply *TSC* and *TFC* for facebook data which is described in Table 5.2. We train the decision tree algorithm for facebook data using wall postings in two consecutive weeks to predict links in the following week. The unit of time for facebook data is days. The experiment was conducted for six data sets and the average performance of *TSC* and *TFC* was computed. Second, we apply *TSC* and *TFC* for coauthorship data extracted from Condmat-ph publications from the year 2000 to 2005. We train the decision tree algorithm using data from year 2000 to 2004 to predict links appeared in the year 2005. The unit of time for Condmat-ph data is days. Details of the experimental results are given in the Tables 5.8 and 5.9 and Table 5.10. Table 5.8 shows number of links predicted by *TSC* and *TFC* for each facebook

Table 5.8: Comparison of number of links predicted by *TSC* and *TFC* for facebook data with respect to whole network

Data	Number of correct predictions by TSC	Number of correct predictions by TFC	Number of links in whole network
D1	81	658	1637
D2	93	783	3028
D3	114	854	2365
D4	59	951	3679
D5	196	1382	5921
D6	64	1270	3952

data set compared to the number of links in the whole network. Table 5.9 shows number of links predicted by *TSC* and *TFC* for Condmat-ph data set compared to the number of links in the whole network. Table 5.10 presents the summary of the comparison of two features in terms of percentage recall of *TSC* and *TFC*. According to the Table 5.10, the average percentage recall of *TFC* is 30.6% for facebook data while average percentage recall of *TSC* is 3.2%. For Condmat-ph data, average percentage recall of *TFC* is 2.9% and average percentage recall of *TSC* is 0.9%. This results indicate that *TFC* perform 10-times better than *TSC* for facebook data and *TFC* perform 3-times better than *TSC* Condmat-ph data. The other observation is that both *TSC* and *TFC* perform better for facebook data compared to Condmat-ph data. This can be happen becuase facebook network mostly evolve through the interactions occur within the local communities while coauthorship networks evolve through the interactions which are not limited to the local communities. Therefore, this observation indicates that *Time score* and *T_Flow* are more effective for networks which evolve through local interactions. Moreover, *T_Flow* can cover any node pair and is more effective than *Time score* which is limited to 2-hops.

Table 5.9: Comparison of number of links predicted by *TSC* and *TFC* for Condmat-ph data with respect to whole network

Data	Number of correct predictions by TSC	Number of correct predictions by TFC	Number of links in whole network
Condmat-ph (2000-2005)	208	702	24406

Table 5.10: Comparison of average percentage recall of *TSC* and *TFC*

Feature combination	Average percentage recall(%)	
	facebook data	Condmat-ph data
TSC	3.229	0.851
TFC	30.582	2.875

5.8 Conclusion

There have been numerous path based/flow based algorithms introduced for link prediction. Most of them have rarely accounted the link activeness. Hence, we extended one of the previous algorithm *PropFlow* by incorporating link activeness and introduced a new algorithm called *T_Flow*. *T_Flow* algorithm computes information flow using activeness of links and link weights. The main characteristic of *T_Flow* algorithm is that it combines the impact of link activeness for information flow which has not been considered in the previous method *PropFlow*. We combined the activeness of links and link weights in *T_Flow* algorithm and investigated how it affect the information flow which is a vital factor for link evolution. The experimental results shows that *T_Flow* algorithm outperform the previous *PropFlow* algorithm which only considers the impact of link weights for information flow. Thus, *T_Flow* algorithm is better for link prediction in social networks where the link activeness varies rapidly over time.

Chapter 6

Conclusion

Summary *In this chapter we summarize the research presented in this thesis and our contributions. We introduced two easy to implement yet significantly effective time-aware methods which can be used for link prediction in highly dynamic social networks. The effectiveness is proved with the evidence of experimental results. Further, we emphasis that these methods are easily extendable to any type of network data.*

6.1 Contribution

We considered the classical problem of link prediction where we are given a snapshot of a social network at time t, and we seek to accurately predict the edges that will be added to the network during the interval from time t to a given future time $t + 1$. Most common way is to measure the closeness/similarity of nodes to each other in terms of various social aspects. Social networks continuously evolve in response to the underlying social dynamics, and those similarities change over time due to highly dynamic behavior of nodes and links. Clearly, older events are less likely to be relevant for determining the future linkages than recent ones. Therefore, it is necessary to develop features or methods which can treat the rapidly changing network data to understand the mechanisms of network evolution. We learnt that time-related data is very important to understand underlying mechanisms of temporality. Thus, we introduced the aspect of time for link prediction.

We contributed by introducing two novel time-related features which can be used with machine learning methods for link prediction in rapidly evolving social networks. We devised two novel time-related features based on link activeness. We used timestamps of interactions, which is strongly correlated with link activeness, to define our novel features. The advantage of using timestamps of interactions is they are easy obtain across many social networks.

6.2 Discussion

The research presented in this thesis introduced two highly effective time-aware features for link prediction in rapidly evolving social networks. Most of the recent link prediction research are being focused on temporal and local patterns of the networks. Number of research works have been introduced time-related features and methods to deal with temporal behavior of node and links. Those features or methods have been defined using social scientific aspects such as *decay* of relationships or social links over time. This phenomenon has been studied extensively in the empirical studies, and revealed that the *decay* of social links is a power function of time. The strength/activeness of social links associated with various factors depending on the network. In most cases, link strength/activeness is strongly correlated with time-related factors but in some others are not. However, generally, strength of social links strongly correlated with time-related factors. A simplest yet vital factor is *link age*. Link age can be interpreted in two ways: (1) elapsed time since the creation of link (2) elapsed time since the last interaction/transaction*, with respect to the current time. According to our perception, the second factor is strongly correlated with link activeness. If transactions or interactions happen frequently and recently the links become active and strong. Active links has an utmost importance for link evolution. Thus, we started from this point and introduced two robust features *Time score* and *T_Flow* to incorporate the following hypotheses of link activeness:

1. If a node pair interacted with each other recently with respect to the current time, then the link between them becomes active.

* We used terms interaction and transaction interchangeably with the same meaning

2. If a node interacted with its neighbors within a closer proximity of time, the neighbors are more likely to become linked.

According to the definition, *Time score* has two terms related to link activeness, $(1-\alpha)^{k_n}$ ($k_n = t_c - max(t_{1_n}, t_{2_n})$) corresponds to the first assumption and $\frac{1}{|t_{1_n}-t_{2_n}|+1}$ corresponds to the second assumption. In this case, *Time score* computes the decay in three different ways. First, when $k_n = 0$ ($t_c = max(t_{1_n}, t_{2_n})$) and $t_{1_n} \neq t_{2_n}$ the decaying function $(1-\alpha)^{k_n}$ becomes 1 for any α and the term $\frac{1}{|t_{1_n}-t_{2_n}|+1}$ uses to deal with decay of link strengths. Second, when $k_n \neq 0$ ($t_c \neq max(t_{1_n}, t_{2_n})$) and $t_{1_n} = t_{2_n}$ the decaying function $(1-\alpha)^{k_n}$ uses to deal with decay of link strengths. Third, when $k_n \neq 0$ ($t_c \neq max(t_{1_n}, t_{2_n})$) and $t_{1_n} \neq t_{2_n}$ both $\frac{1}{|t_{1_n}-t_{2_n}|+1}$ and $(1-\alpha)^{k_n}$ uses to deal with decay of link strengths. Changing its form according to some corner cases is one drawback of the *Time score*, i.e. it doesn't use unique function to deal with decay of link strength in some special cases such as first and second case. This drawback has been alleviated in *T_Flow* while including all the aspects of *Time score*. Important point is that *T_Flow* algorithm only use decaying function $(1-\alpha)^{|t_x-t_y|}$ to deal with decay of link strengths. At the start of random walk, t_x is considered as the current time or a given time such elapsed time measure with respect to the given time. In the middle of the random walk t_x and t_y are the timestamps of the adjacent edges. Therefore, *T_Flow* doesn't change its form irrespective of the cases such as $\alpha = 0$, $t_x = t_y$, $t_x = t_y = t_c$, etc. Thus, *T_Flow* algorithm computes the decay in information flow in the same manner for any node pair. The consistency has made *T_Flow* algorithm is an effective and generalized method to deal with the decay of link activeness. Both novel features are capable of dealing with rapidly changing nature of the networks. We used the timestamps of the links/interactions to devise the features. The timestamps of links/interactions are available and a common information across most social networks. Therefore, the novel time-aware features are easily adaptable for any kind of network.

We identified some limitations in the time-aware features we introduced in this thesis. One of them is, we used one type of interactions to determine the link activeness. The hypothesis behind the novel features is the activeness of links is highly correlated with future link evolution. Under this assumption we used timestamps of interactions or transactions to determine the activeness of links. In-

teractions can be happen in different ways. For example, facebook users interact with their friends by exchanging wall postings, photo/video tagging, chatting, likings, etc. Every way of interaction contribute for activeness of the links between nodes and hence, affect the future link evolution. It is worthwhile to study and investigate all these factors to determine the link activeness. It would lead to formulate accurate link prediction methods. However, it is a complex task to investigate the relationship between activeness of links all the factors contribute for link activeness. The influence of these factors vary over time in different and complex patterns. Besides, the data related to most of these factors are not easy to obtain due to various issues such as privacy and security matters. Thus, we choose easy obtain and commonly available interaction types for the data sets we used in our experiments. In order to study the phenomenon of link activeness, we picked wall postings as the interaction type for facebook data and coauthorships as the interaction type for scientific collaboration network data. We collected the timestamps of interactions of both data sets. Time take for an interaction is different for each data set. In facebook online social network, the average interaction/response time for a wall posting takes 12 to 48 hours. In contrast, in the social networks such as coauthor networks the average interaction time is around 1 year. Generally, the scientific papers published in conferences held annually. However, there are some exceptions exist depending on the research areas. Thus, we decided to use days as the time unit for facebook data and years as the time unit for coauthorship data as an average time measure.

The features *Time score* and *T_flow* include a parameter α, the decaying factor. Decaying factor α ($0 < \alpha < 1$) is the rate of decay in link activeness per unit time. In our work, we analyzed the performance by varying the parameter values within a given set of values from 0.1 to 0.9. The reason for this analysis it to provide a guideline to pick the parameter value particularly, according to the time unit. We observed following characteristics in the experimental analysis:

1. For facebook data, both *Time score* and *T_Flow* shows optimal link prediction performance for smaller α values ($\alpha = 0.1$). This observation implies that smaller α values are suitable for rapidly evolving networks with short interaction time such as days, hours, etc.

2. For coauthorship data, *Time score* shows the optimal link prediction performance for Condmat-ph data at $\alpha = 0.8$. *T_Flow* shows optimal link prediction performance for Astro-ph data at $\alpha = 0.3$, for Condmat-ph data at $\alpha = 0.9$, for Hep-ph data at $\alpha = 0.7$and for Hep-ph data at $\alpha = 0.6$. These observations implies that higher α values are suitable for rapidly growing networks with longer interaction time such as months, years, etc. However, optimal link prediction performance for Astro-ph data observed at $\alpha = 0.3$ which an exception compared to the other data sets. According to Table 5.4 Astro-ph data has the highest growth of links and mean degree which implies that this network is highly active in terms of interactions compared to the other networks. In other words the interactions happen fairly quickly than in the other networks. Thus, the average time taking for an interaction might be less than 1 year. Thus, smaller α values shows better results for Astro-ph data.

Although we picked the parameter value for α *Time score* using the above analytical procedure,parameter learning method called two-loop cross validation used for evaluating *T_flow*. Two-loop cross validation method learns the best parameter value in its inner loop and used it for feature evaluation in the our loop. However, we used the set of nine values $(0.1, 0.2, ..., 0.9)$ for this evaluation. We didn't test the values in between them such as $0.15, 0.24, 0.36, \ldots$. It would be more effective if we tested such kind of values. This can be done by more sophisticated parameter learning methods which we hope to incorporate in our future works.

In the present work, we developed our methods for homogeneous networks. However, most of the social networks are represented as multi-relational networks. Users linked with others via different types of links. Thus, predicting the links which are more likely to happen with their types has become an important aspect of link prediction. Each type of link in a network is related to a specific service or purpose. This implies that, a user use different channels or links to connect with another user depending on the time. This is another time related temporal behavior of links, which is one of our future directions. In our future works, We will be incorporate the novel features introduced in this thesis in multi-relational link prediction framework to provide improved and accurate link prediction system.

6.3 Conclusion

There have been numerous attempts to address the problem of link prediction using supervised learning methods which acquire the knowledge through various measures. However, the knowledge gained through the existing static knowledge extraction measures are not sufficient for accurate link prediction in highly dynamic social networks. It leads to inaccurate and less precise predictions. In order to alleviate this problem, we contributed by introducing two novel time-aware features which are based on common neighbors and information flow via active links. We used the latest timestamps, which are easily obtainable in most networks, of interactions/links to compute them. Both features tested in conjunction with supervised learning method on real world social networks namely, *facebook* friendship network data and coauthorship data extracted from *ePrint archives*. The results shows that new time-aware methods, *Time score* and *T_Flow*, outperform the existing static methods. Besides, that we analyzed the performance of both methods by varying their parameter values in order to provide a guideline for choosing parameters for different networks. Thus, they are sophisticated methods to use any kind of evolving network. In our concluding remarks, we would like to emphasis that two time-aware methods are extremely helpful in link prediction in dynamic social networks.

Bibliography

[1] Lada A. Adamic and Eytan Adar. Friends and neighbors on the web. *SOCIAL NETWORKS*, 25:211–230, 2001.

[2] Muhammad Aurangzeb Ahmad, Zoheb Borbora, Jaideep Srivastava, and Noshir S. Contractor. Link prediction across multiple social networks. In *Proceedings of The 10th IEEE International Conference on Data Mining Workshops*, pages 911–918, 2010.

[3] Luca Maria Aiello, Alain Barrat, Ciro Cattuto, Giancarlo Ruffo, and Rossano Schifanella. Link creation and profile alignment in the anobii social network. In *Proceedings of 2010 IEEE Second International Conference on Social Computing/ IEEE International Conference on Privacy, Security, Risk and Trust*, pages 249–256, 2010.

[4] Luca Maria Aiello, Alain Barrat, Rossano Schifanella, Ciro Cattuto, Benjamin Markines, and Filippo Menczer. Friendship prediction and homophily in social media. *ACM Transactions on the Web*, 6(2):9, 2012.

[5] Lars Backstrom and Jure Leskovec. Supervised random walks: predicting and recommending links in social networks. In *Proceedings of Forth International Conference on Web Search and Web Data Mining*, pages 635–644, 2011.

[6] Albert-László Barabási. From networks to human behavior. In *Proceedings of 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 435, 2009.

[7] Ronald S Burt. Decay functions. *Social Networks*, 22:1 – 28, 2000.

[8] Deng Cai, Zheng Shao, Xiaofei He, Xifeng Yan, and Jiawei Han. Community mining from multi-relational networks. In *Proceedings Knowledge Discovery in Databases: PKDD 2005, 9th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 445–452, 2005.

[9] Bin Cao, Nathan Nan Liu, and Qiang Yang. Transfer learning for collective link prediction in multiple heterogenous domains. In *Proceedings of 27th International Conference on Machine Learning*, pages 159–166, 2010.

[10] Meeyoung Cha, Alan Mislove, and P. Krishna Gummadi. A measurement-driven analysis of information propagation in the flickr social network. In *Proceedings of 18th International Conference on World Wide Web*, pages 721–730, 2009.

[11] Deepayan Chakrabarti and Rupesh R. Mehta. The paths more taken: matching dom trees to search logs for accurate webpage clustering. In *Proceedings of 19th International Conference on World Wide Web*, pages 211–220, 2010.

[12] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[13] Feilong Chen, Jerry Scripps, and Pang-Ning Tan. Link mining for a social bookmarking web site. In *Proceedings of IEEE / WIC / ACM International Conference on Web Intelligence*, pages 169–175, 2008.

[14] Justin Cheng, Daniel Mauricio Romero, Brendan Meeder, and Jon M. Kleinberg. Predicting reciprocity in social networks. In *3rd International Confernece on Social Computing*, pages 49–56, 2011.

[15] Eunjoon Cho, Seth A. Myers, and Jure Leskovec. Friendship and mobility: user movement in location-based social networks. In *Proceedings of 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1082–1090, 2011.

[16] Elizabeth F. Churchill and Christine A. Halverson. Guest editors' introduction: Social networks and social networking. *IEEE Internet Computing*, 9(5):14–19, 2005.

[17] Aaron Clauset, Cosma Rohilla Shalizi, and Mark E. J. Newman. Power-law distributions in empirical data. *SIAM Review*, 51(4):661–703, 2009.

[18] David Cohn and Huan Chang. Learning to probabilistically identify authoritative documents. In *Proceedings of 17th International Conference on Machine Learning*, pages 167–174, 2000.

[19] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3), 1995.

[20] Bing Tian Dai, Freddy Chong Tat Chua, and Ee-Peng Lim. Structural analysis in multi-relational social networks. In *Proceedings of 12th SIAM International Conference on Data Mining*, pages 451–462, 2012.

[21] Darcy A. Davis, Ryan Lichtenwalter, and Nitesh V. Chawla. Multi-relational link prediction in heterogeneous information networks. In *Proceedings of International Conference on Advances in Social Networks Analysis and Mining*, pages 281–288, 2011.

[22] Yuxiao Dong, Jie Tang, Sen Wu, Jilei Tian, Nitesh V. Chawla, Jinghai Rao, and Huanhuan Cao. Link prediction and recommendation across heterogeneous social networks. In *Proceedings of 12th IEEE International Conference on Data Mining*, pages 181–190, 2012.

[23] Janardhan Rao Doppa, Jun Yu, Prasad Tadepalli, and Lise Getoor. Learning algorithms for link prediction based on chance constraints. In *Proceedings of European Conference Machine Learning and Knowledge Discovery in Databases*, pages 344–360, 2010.

[24] Eibe Frank, Mark A. Hall, Geoffrey Holmes, Richard Kirkby, and Bernhard Pfahringer. Weka - a machine learning workbench for data mining. In *The Data Mining and Knowledge Discovery Handbook*, pages 1305–1314. 2005.

[25] Lise Getoor. Link mining: a new data mining challenge. *SIGKDD Explorations*, 5(1):84–89, 2003.

[26] Lise Getoor. Link mining and link discovery. In *Encyclopedia of Machine Learning*, pages 606–609. 2010.

[27] Lise Getoor and Christopher P. Diehl. Introduction to the special issue on link mining. *SIGKDD Explorations*, 7(2):1–2, 2005.

[28] Rumi Ghosh and Kristina Lerman. Structure of heterogeneous networks. In *12th IEEE International Conference on Computational Science and Engineering*, pages 98–105, 2009.

[29] Jiawei Han, Micheline Kamber, and Jian Pei. *Data mining: concepts and techniques*. 2006.

[30] Nobuyuki Hanaki, Alexander Peterhansl, Peter S. Dodds, and Duncan J. Watts. Cooperation in evolving social networks. *Management Science*, 53(7):1036–1050, 2007.

[31] Mohammad Al Hasan and Mohammed J. Zaki. A survey of link prediction in social networks. In *Social Network Data Analytics*, pages 243–275. 2011.

[32] T. Ryan Hoens, Qi Qian, Nitesh V. Chawla, and Zhi-Hua Zhou. Building decision trees for the multi-class imbalance problem. In *Proceedings of Advances in Knowledge Discovery and Data Mining - 16th Pacific-Asia Conference*, pages 122–134, 2012.

[33] Zan Huang, Xin Li, and Hsinchun Chen. Link prediction approach to collaborative filtering. In *Proceedings of ACM/IEEE Joint Conference on Digital Libraries*, pages 141–142, 2005.

[34] Zan Huang and Dennis K. J. Lin. The time-series link prediction problem with applications in communication surveillance. *INFORMS Journal on Computing*, 21(2):286–303, 2009.

[35] Zan Huang and Daniel Dajun Zeng. A link prediction approach to anomalous email detection. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, pages 1131–1136, 2006.

[36] Reid A. Johnson, Yang Yang, Everaldo Aguiar, Andrew K. Rider, and Nitesh V. Chawla. Alive: A multi-relational link prediction environment for the healthcare domain. In *Proceedings of Emerging Trends in Knowledge Discovery and Data Mining - PAKDD 2012 International Workshops*, pages 36–46, 2012.

[37] Krzysztof Juszczyszyn, Adam Gonczarek, Jakub M. Tomczak, Katarzyna Musial, and Marcin Budka. A probabilistic approach to structural change prediction in evolving social networks. In *Proceedings of International Conference on Advances in Social Networks Analysis and Mining*, pages 996–1001, 2012.

[38] Indika Kahanda and Jennifer Neville. Using transactional information to predict link strength in online social networks. In *Proceedings of 3rd International Conference on Weblogs and Social Media*, 2009.

[39] Hisashi Kashima and Naoki Abe. A parameterized probabilistic model of network evolution for supervised link prediction. In *Proceedings of 6th IEEE International Conference on Data Mining*, pages 340–349, 2006.

[40] Hisashi Kashima, Tsuyoshi Kato, Yoshihiro Yamanishi, Masashi Sugiyama, and Koji Tsuda. Link propagation: A fast semi-supervised learning algorithm for link prediction. In *Proceedings of SIAM International Conference on Data Mining*, pages 1099–1110, 2009.

[41] Myunghwan Kim and Jure Leskovec. Modeling social networks with node attributes using the multiplicative attribute graph model. In *Proceedings of 27th Conference on Uncertainty in Artificial Intelligence*, pages 400–409, 2011.

[42] Myunghwan Kim and Jure Leskovec. The network completion problem: Inferring missing nodes and edges in networks. In *Proceedings of 11th SIAM International Conference on Data Mining*, pages 47–58, 2011.

[43] Gueorgi Kossinets, Jon M. Kleinberg, and Duncan J. Watts. The structure of information pathways in a social communication network. In *Proceedings of 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 435–443, 2008.

[44] Sotiris B. Kotsiantis, Ioannis D. Zaharakis, and Panayiotis E. Pintelas. Machine learning: a review of classification and combining techniques. *Artificial Intelligence Review*, 26(3):159–190, 2006.

[45] Jérôme Kunegis, Andreas Lommatzsch, and Christian Bauckhage. The slashdot zoo: mining a social network with negative edges. In *Proceedings of 18th International Conference on World Wide Web*, pages 741–750, 2009.

[46] Vincent Leroy, Berkant Barla Cambazoglu, and Francesco Bonchi. Cold start link prediction. In *Proceedings of 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 393–402, 2010.

[47] Jure Leskovec, Lars Backstrom, Ravi Kumar, and Andrew Tomkins. Microscopic evolution of social networks. In *Proceedings of 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 462–470, 2008.

[48] Jure Leskovec, Daniel P. Huttenlocher, and Jon M. Kleinberg. Predicting positive and negative links in online social networks. In *Proceedings of 19th International Conference on World Wide Web*, pages 641–650, 2010.

[49] Jure Leskovec, Jon M. Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *Transactions on Knowledge Discovery from Data*, 1(1), 2007.

[50] Cheng-Te Li and Shou-De Lin. Social flocks: a crowd simulation framework for social network generation, community detection, and collective behavior modeling. In *Proceedings of 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 765–768, 2011.

[51] Cheng-Te Li, Shou-De Lin, and Man-Kwan Shan. Finding influential mediators in social networks. In *Proceedings of 20th International Conference on World Wide Web*, pages 75–76, 2011.

[52] David Liben-Nowell and Jon M. Kleinberg. The link-prediction problem for social networks. *Journal of American Society for Information Science and Technology*, 58(7):1019–1031, 2007.

[53] Louis Licamele and Lise Getoor. Social capital in friendship-event networks. In *Proceedings of 6th IEEE International Conference on Data Mining*, pages 959–964, 2006.

[54] Ryan Lichtenwalter and Nitesh V. Chawla. Lpmade: Link prediction made easy. *Journal of Machine Learning Research*, 12:2489–2492, 2011.

[55] Ryan Lichtenwalter and Nitesh V. Chawla. Link prediction: Fair and effective evaluation. In *Proceedings of International Conference on Advances in Social Networks Analysis and Mining*, pages 376–383, 2012.

[56] Ryan Lichtenwalter and Nitesh V. Chawla. Vertex collocation profiles: subgraph counting for link analysis and prediction. In *Proceedings of 21st World Wide Web Conference*, pages 1019–1028, 2012.

[57] Ryan Lichtenwalter, Jake T. Lussier, and Nitesh V. Chawla. New perspectives and methods in link prediction. In *Proceedings of 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 243–252, 2010.

[58] Shou-De Lin and Hans Chalupsky. Unsupervised link discovery in multi-relational data via rarity analysis. In *Proceedings of 3rd IEEE International Conference on Data Mining*, pages 171–178, 2003.

[59] Xingjie Liu, Qi He, Yuanyuan Tian, Wang-Chien Lee, John McPherson, and Jiawei Han. Event-based social networks: linking the online and offline social worlds. In *The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1032–1040, 2012.

[60] Jing-Kai Lou, Shou-De Lin, Kuan-Ta Chen, and Chin-Laung Lei. What can the temporal social behavior tell us? an estimation of vertex-betweenness using dynamic social information. In *Proceedings of International Conference on Advances in Social Networks Analysis and Mining*, pages 56–63, 2010.

[61] Zhengdong Lu, Berkant Savas, Wei Tang, and Inderjit S. Dhillon. Supervised link prediction using multiple sources. In *Proceedings of 10th IEEE International Conference on Data Mining*, pages 923–928, 2010.

[62] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*. Cambridge University Press, 2008.

[63] Paolo Massa. Social networks of wikipedia. In *Proceedings of 22nd ACM Conference on Hypertext and Hypermedia*, pages 221–230, 2011.

[64] Paolo Massa, Martino Salvetti, and Danilo Tomasoni. Bowling alone and trust decline in social network sites. In *proceedings of 8th IEEE International Conference on Dependable, Autonomic and Secure Computing*, pages 658–663, 2009.

[65] Julian J. McAuley and Jure Leskovec. Learning to discover social circles in ego networks. In *Proceedings of 26th Annual Conference on Neural Information Processing Systems*, pages 548–556, 2012.

[66] Victor Ströele A. Menezes, Geraldo Zimbrão, and Jano M. Souza. Modeling, mining and analysis of multi-relational scientific social network. *Journal of Universal Computer Science*, 18(8):1048–1068, 2012.

[67] Alan Mislove, Bimal Viswanath, P. Krishna Gummadi, and Peter Druschel. You are who you know: inferring user profiles in online social networks. In *Proceedings of 3rd International Conference on Web Search and Web Data Mining*, pages 251–260, 2010.

[68] Lankeshwara Munasinghe and Ryutaro Ichise. Time aware index for link prediction in social networks. In *proceedings of 13th International Con-*

ference of Data Ware housing Knowledge and Discovery, pages 342–353, 2011.

[69] Lankeshwara Munasinghe and Ryutaro Ichise. Time score: A new feature for link prediction in social networks. *IEICE Transactions*, E95-D(3):821–828, 2012.

[70] Lankeshwara Munasinghe and Ryutaro Ichise. Exploiting Information Flow and Active Links for Link Prediction in Social Networks. In *proceedings of 26th Annual Conference of the Japanese Society for Artificial Intelligence*, 2012.

[71] Lankeshwara Munasinghe and Ryutaro Ichise. Link prediction in social networks using information flow via active links. *IEICE Transactions*, E96-D(7):1495–1502, 2013.

[72] Lankeshwara Munasinghe and Ryutaro Ichise. Multi-class Link Prediction in Social Networks. In *proceedings of 27th Annual Conference of the Japanese Society for Artificial Intelligence*, 2013.

[73] Tsuyoshi Murata. Detecting communities in social networks. In *Handbook of Social Network Technologies*, pages 269–280. 2010.

[74] Tsuyoshi Murata and Sakiko Moriyasu. Link prediction of social networks based on weighted proximity measures. In *Proceedings of IEEE / WIC / ACM International Conference on Web Intelligence*, pages 85–88, 2007.

[75] Tsuyoshi Murata and Sakiko Moriyasu. Link prediction based on structural properties of online social networks. *New Generation Computing*, 26(3):245–257, 2008.

[76] Galileo Namata and Lise Getoor. Link prediction. In *Encyclopedia of Machine Learning*, pages 609–612. 2010.

[77] Mark E. J. Newman. Scientific collaboration networks. ii.shortest paths, weighted networks, and centrality. *Physical review E*, 64(1):016132, 2001.

[78] Mark. E. J. Newman. Scientific collaboration networks. i.network construction and fundamental results. *Physical Review E*, 64:016131, 2001.

[79] Mark E. J. Newman. The structure of scientific collaboration networks. *Proceedings of National Academy of Sciences of the United States of America*, 98:404–409, 2001.

[80] Mark E. J. Newman, Duncan J Watts, and Steven H Strogatz. Random graph models of social networks. *Proceedings of National Academy of Sciences of the United States of America*, 99(Suppl 1):2566–2572, 2002.

[81] Satoshi Oyama, Kohei Hayashi, and Hisashi Kashima. Cross-temporal link prediction. In *11th IEEE International Conference on Data Mining*, pages 1188–1193, 2011.

[82] Satoshi Oyama, Kohei Hayashi, and Hisashi Kashima. Link prediction across time via cross-temporal locality preserving projections. *IEICE Transactions*, 95-D(11):2664–2673, 2012.

[83] Alexis Papadimitriou, Panagiotis Symeonidis, and Yannis Manolopoulos. Friendlink: Link prediction in social networks via bounded local path traversal. In *Proceedings of International Conference on Computational Aspects of Social Networks*, pages 66–71, 2011.

[84] Milen Pavlov and Ryutaro Ichise. Finding experts by link prediction in co-authorship networks. In *Proceedings of 2nd International ISWC+ASWC Workshop on Finding Experts on the Web with Semantics*, pages 42–55, 2007.

[85] Alexandrin Popescul, Lyle H. Ungar, Steve Lawrence, and David M. Pennock. Statistical relational learning for document mining. In *Proceedings of 3rd IEEE International Conference on Data Mining*, pages 275–282, 2003.

[86] J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[87] Garry Robins. Social networks, exponential random graph (p*) models for. In *Encyclopedia of Complexity and Systems Science*, pages 8319–8333. 2009.

[88] Mrinmaya Sachan and Ryutaro Ichise. Using abstract information and community alignment information for link prediction. *International Journal of Engineering and Technology*, 2(4):334–339, 2010.

[89] Purnamrita Sarkar, Deepayan Chakrabarti, and Michael I. Jordan. Nonparametric link prediction in dynamic networks. In *Proceedings of 29th International Conference on Machine Learning*, 2012.

[90] Purnamrita Sarkar, Deepayan Chakrabarti, and Andrew W. Moore. Theoretical justification of popular link prediction heuristics. In *The 23rd Conference on Learning Theory*, pages 295–307, 2010.

[91] Rossano Schifanella, Alain Barrat, Ciro Cattuto, Benjamin Markines, and Filippo Menczer. Folks in folksonomies: social link prediction from shared metadata. In *Proceedings of 3rd International Conference on Web Search and Web Data Mining*, pages 271–280, 2010.

[92] Jerry Scripps, Pang-Ning Tan, Feilong Chen, and Abdol-Hossein Esfahanian. A matrix alignment approach for link prediction. In *19th International Conference on Pattern Recognition*, pages 1–4, 2008.

[93] Jerry Scripps, Pang-Ning Tan, and Abdol-Hossein Esfahanian. Exploration of link structure and community-based node roles in network analysis. In *Proceedings of 7th IEEE International Conference on Data Mining*, pages 649–654, 2007.

[94] Ted E. Senator. Link mining applications: progress and challenges. *SIGKDD Explorations*, 7(2):76–83, 2005.

[95] Umang Sharan and Jennifer Neville. Temporal-relational classifiers for prediction in evolving domains. In *Proceedings of 8th IEEE International Conference on Data Mining*, pages 540–549, 2008.

[96] Naoki Shibata, Yuya Kajikawa, and Ichiro Sakata. Link prediction in citation networks. *Journal of the American Society for Information Science and Technology*, 63(1):78–85, 2012.

[97] Jie Tang, Tiancheng Lou, and Jon M. Kleinberg. Inferring social ties across heterogenous networks. In *Proceedings of 5th International Conference on Web Search and Web Data Mining*, pages 743–752, 2012.

[98] Lei Tang and Huan Liu. Graph mining applications to social network analysis. In *Managing and Mining Graph Data*, pages 487–513. 2010.

[99] Lei Tang, Xufei Wang, and Huan Liu. Uncoverning groups via heterogeneous interaction analysis. In *The 9th IEEE International Conference on Data Mining*, pages 503–512, 2009.

[100] Benjamin Taskar, Ming Fai Wong, Pieter Abbeel, and Daphne Koller. Link prediction in relational data. In *Proceedings of Advances in Neural Information Processing Systems*, 2003.

[101] Yuan Tian, Qi He, Qiankun Zhao, Xingjie Liu, and Wang-Chien Lee. Boosting social network connectivity with link revival. In *Proceedings of 19th ACM Conference on Information and Knowledge Management*, pages 589–598, 2010.

[102] Tomasz Tylenda, Ralitsa Angelova, and Srikanta J. Bedathur. Towards time-aware link prediction in evolving social networks. In *Proceedings of 3rd Workshop on Social Network Mining and Analysis*, page 9, 2009.

[103] Bimal Viswanath, Alan Mislove, Meeyoung Cha, and P. Krishna Gummadi. On the evolution of user interaction in facebook. In *Proceedings of 2nd ACM Workshop on Online Social Networks*, pages 37–42, 2009.

[104] Chao Wang, Venu Satuluri, and Srinivasan Parthasarathy. Local probabilistic models for link prediction. In *Proceedings of 7th IEEE International Conference on Data Mining*, pages 322–331, 2007.

[105] Dashun Wang, Dino Pedreschi, Chaoming Song, Fosca Giannotti, and Albert-László Barabási. Human mobility, social ties, and link prediction. In *Proceedings of 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1100–1108, 2011.

[106] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)*. 2005.

[107] Till Wohlfarth and Ryutaro Ichise. Semantic and event-based approach for link prediction. In *Practical Aspects of Knowledge Management, 7th International Conference*, pages 50–61, 2008.

[108] Elena Zheleva, Lise Getoor, Jennifer Golbeck, and Ugur Kuter. Using friendship ties and family circles for link prediction. In *Advances in Social Network Mining and Analysis, Second International Workshop*, pages 97–113, 2008.