Link Prediction in Heterogeneous Social Networks

Thesis submitted in partial fulfilment
of the requirements for the degree of

Doctor of Philosophy
in
Computer Science

by

T. Jaya Lakshmi

10MCPC21

Under the supervision of

Dr. S. Durga Bhavani

School of Computer and Information Sciences

University of Hyderabad

Central University

Hyderabad - 500 046, INDIA

September 2018

Certificate

This is to certify that the thesis entitled **Link Prediction in Heterogeneous Social Networks** submitted by **T.Jaya Lakshmi** bearing registration number 10MCPC21 in partial fulfilment of the requirements for award of Doctor of Philosophy in the school of Computer and Information Sciences is a bonafide work carried out by her under my supervision and guidance.

This thesis is free from plagiarism and has not been submitted previously in part or in full to this or any other University or Institution for award of any degree or diploma. Further, the student has the following publications before submission of the thesis for adjudication and has produced evidence for the same in the form of acceptance letter or the reprint in the relevant area of her research.

1. T. Jaya Lakshmi and S. Durga Bhavani. Link Prediction Measures in Various Types of Information Networks : A Review. **ASONAM 2018**, Barcelona, Spain, August, 2018.(DBLP) (Presented.)
2. T. Jaya Lakshmi and S. Durga Bhavani. Link Prediction in Temporal Heterogeneous Networks. **PAISI 2017, (PAKDD Workshop)**, South Korea, pp 83-98, May 2017. (DBLP)
3. T. Jaya Lakshmi and S. Durga Bhavani, Temporal probabilistic measure for link prediction in collaborative networks". **J. Applied Intelligence**, Volume 47, Issue 1, pp 83-95, July 2017. (SCI)

Further, the student has passed the following courses towards fulfilment of coursework requirement for Ph.D:

S. No	Course Code	Name	Credits	Pass/Fail
1	CS 801	Data Structures and Algorithms	4	Pass
2	CS 802	Operating Systems and Programming	4	Pass
3	AI 853	Data Mining	4	Pass
4	AI 851	Trenda in Softcomputing	4	Pass

Supervisor
School of Computer and Information Sciences
University of Hyderabad
Hyderabad - 500046

Dean
School of Computer and Information Sciences
University of Hyderabad
Hyderabad - 500046

DECLARATION

I, T.Jaya Lakshmi, here by declare that this thesis entitled **Link Prediction in Heterogeneous Social Networks** submitted by me under the guidance and supervision of Dr. S. Durga Bhavani is a bonafide research work. I also declare that it has not been submitted previously in part or in full to this University or any other University or Institution for the award of any degree or diploma.

Date T.Jaya Lakshmi

Place

Acknowledgements

Writing this thesis has been fascinating and extremely rewarding. I would like to thank a number of people who have contributed to the final result in many different ways.

First and foremost, I am profoundly grateful to my research supervisor Dr.S.Durga Bhavani, School of Computer and Information Sciences, for her valuable support throughout the thesis. I will be grateful for her throughout my rest of life as she had set a great example of mentorship. I learned several qualities other than research from her. Her expertise, invaluable guidance, constant encouragement, affectionate attitude, understanding, patience and healthy criticism added considerably to my experience. Without her continual inspiration, it would have not been possible to complete this study.

I take this opportunity to convey my respectful regards to the present and previous deans of School of Computer Information Sciences for providing the necessary resources and a pleasant working atmosphere.

I express my deep sense of gratitude to my review committee members Dr. Hrushikesha Mohanthy and Dr.Alok Singh for their constructive suggestions in reviews.

I sincerely admire the contribution of all my lab mates of Computational Intelligence Lab at University of Hyderabad.

My special regards to my teachers at all stages of my study from childhood because of whose teaching at different stages of education has made it possible for me to see this day.

Words prove a meagre media to write down my feelings for my family for their eternal support and understanding of my goals and aspirations. My husband Mr.Ram Prasad Chivukula and my son Rohit Chivukula's infallible love and support has always been my strength. Their unconditional love, patience and sacrifice will remain my inspiration throughout my life. Without their help, I would not have been able to complete much of what I have done and become who I am. It would be ungrateful on my part if I thank Ramprasad and Rohit in these few words.

My research would have been impossible without the support of my parents Mr. T.R.K. Murthy and Ms. Balachamundi. They gave me invaluable help whenever I need even before asking them. They have celebrated my small happy moment as great achievement.

I miss my late father-in-law Jagadguru Sri Sri Sri Siva Kalyananda Bhatathi Maha Swami at this moment. I will always be grateful to him for recognizing potential in me that I could do research and encouraging me for study after my marriage, when I myself did not realize.

Finally, I would like to thank the management, principal, Head of the department of IT and all

the colleagues from Vasireddy Venkatadri Institute of Technology for their cooperation and adjusting my work whenever I availed leaves for my research work.

Above all, I owe it all to the invisible super power for granting me the wisdom, health and strength to undertake this research task and enabling me to its completion.

Date T.Jaya Lakshmi

ABSTRACT

A social network is represented as a graph where nodes represent entities and edges/links represent the set of interactions/relations between these entities. Link Prediction problem predicts the likelihood of a future interaction between two nodes when an interaction is not present at the current instant of time. Link prediction problem has potential application to problems such as network evolution, recommendation systems and drug-target discovery.

The nodes and edges in a network can be of same type which are called homogeneous networks and those having multiple types are called multi-relational and heterogeneous networks. Many measures based on graph topology like Common Neighbours, Jaccard Coefficient, Adamic Adar, Preferential Attachment, Katz, PropFlow etc. have been proposed in the literature for homogeneous networks. Extensions of many of these measures to heterogeneous networks are not available. In this work, we extend these measures to heterogeneous networks.

In this thesis, we focus our attention on Co-occurrence probability (COP) measure, that has been proposed in the graphical model framework. We find that the time information associated with the links plays a major role in future link formation. There have been a few measures like Time-score, Link-score and T_Flow, which utilize temporal information for link prediction. In this work, Time-score has been innovatively incorporated into the graphical model framework, yielding a novel measure called Temporal Co-occurrence Probability (TCOP) for link prediction.

Further, we extend the measure $TCOP$ to heterogeneous networks which is named as Heterogeneous Temporal Co-occurrence probability (*Hetero-TCOP*) measure. All the extended heterogeneous measures along with *Hetero-TCOP* are evaluated on the bench mark datasets of DBLP, HiePh-collab, HiePh-cite and Condmat bibliographic networks for predicting two types of links: heterogeneous (author-conference) and homogeneous (author-author) in the heterogeneous environment. In all the four networks, *Hetero-TCOP* achieves superior performance over the standard topological measures. In the case of DBLP dataset, *Hetero-TCOP* shows an improvement of 15% accuracy over neighbourhood-based measures, 6% over temporal measures and 5% over Co-occurrence probability measure. Similar improvement in performance is observed for other datasets also.

Time and memory are major challenges for the link prediction task in large heterogeneous social networks. This challenge is addressed in this work, by proposing a divide and conquer method for link prediction called Community based Link Prediction (CBLP). By implementing the proposed LP measures within different communities and combining the results, show a significant speed-up of the algorithm as

well as improvement in the results of prediction performance.

The effectiveness of the newly proposed LP measures have been tested in a novel domain, that of recommender systems. The application is limited to the scope of predicting a possible link between an item and a user, and does not predict the actual rating. This approach is evaluated on the bench mark MovieLens dataset using AUROC, AUPR and Rank-score measures. TCOP outperformed all the other link prediction measures as well as some of the classical recommendation algorithms.

Contents

List of Figures

List of Tables

1. Introduction

Any system containing entities with interactions existing between the entities can be modelled as a network. A social network is represented as a graph where nodes represent entities and edges represent a set of the interactions between these entities. An example of a social network is shown in Fig. 1.1 in which the nodes are representing people and there are two types of links : friendship (facebook) and professional (LinkedIn).

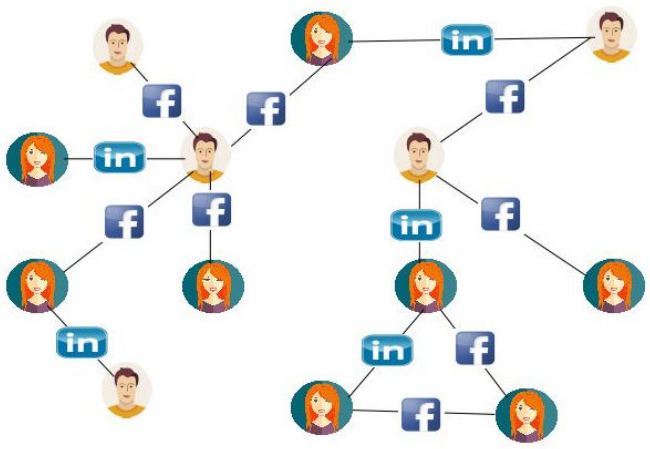


Figure 1.1.: Social Network

In this thesis, we use network and graph interchangeably and also the vertices and nodes. The links/edges refer to the associations, interactions or relationships between the nodes. Social networks differ based on the types of nodes and edges as explained below.

- **Homogeneous network** contains nodes and edges of same type. An example of homogeneous network is shown in Fig.1.2.
- **Heterogeneous Network** has multiple types of nodes and multiple types of edges. An example of a collaboration network is given in Fig.1.5. **Multi-relational**(MR) network and **Bipartite network**

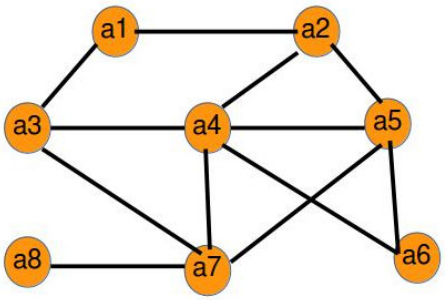


Figure 1.2.: Homogeneous Network where a_i are authors linked by a coauthorship relation

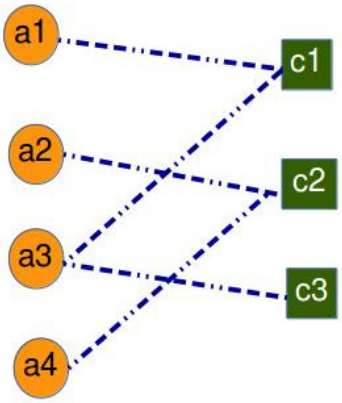


Figure 1.3.: Bipartite Network in which author nodes a_i are linked to conference nodes c_j by publish relation

are special types of heterogeneous networks. A MR network contains single type of nodes and multiple types of edges ss shown in Fig.1.4 where as a bipartite network contains exactly two types of nodes and edges exist between two different types of nodes as shown in Fig.1.3.

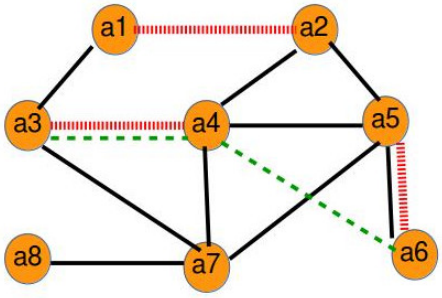


Figure 1.4.: Multi-relational Network in which authors are linked by three types of relations shown in different colors

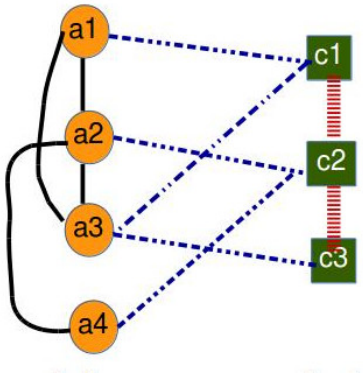


Figure 1.5.: Heterogeneous Network

- If the edges in the network has a weight associated with them, then it is called as a **weighted network**.
- **Temporal network** is a network where time of formation of edges is available. An example temporal weighted network is given in Fig.1.6.

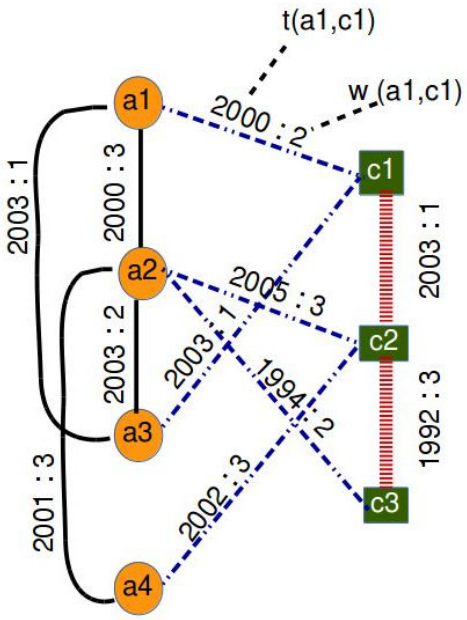


Figure 1.6.: Temporal Weighted Network

Social networks are dynamic in nature. New interactions may be established between nodes, leading to addition of new edges to network, causing growth in networks. The growth of the social graphs has been exponential in the past decade. Fig.1.7 shows the increase in the publications of DBLP [1] from its inception to the year 2018 and Fig.1.8 shows the growth of facebook [2] users during the years 2008 to 2017.

Network evolution is a research domain which studies models of growth in the networks [3, 4, 5, 6, 7]. A large number of parameters affect this growth. A relatively easier task would be focus on possible association between pairs of specific nodes, instead of predicting the whole graph evolution and this problem is called as **link prediction problem**. The problem of predicting future links or identifying missing links in a network is a very important problem for understanding the evolution of the social network.

1.1. Link Prediction Problem

Definition 1.1. Liben Nowell et al. define link-prediction problem [8] as : Given a social network $G(V,E,w,t)$ with vertex set V, set of edges E along with time function t and weight function w on E, to predict new edges that are likely to form in the network at a future time.

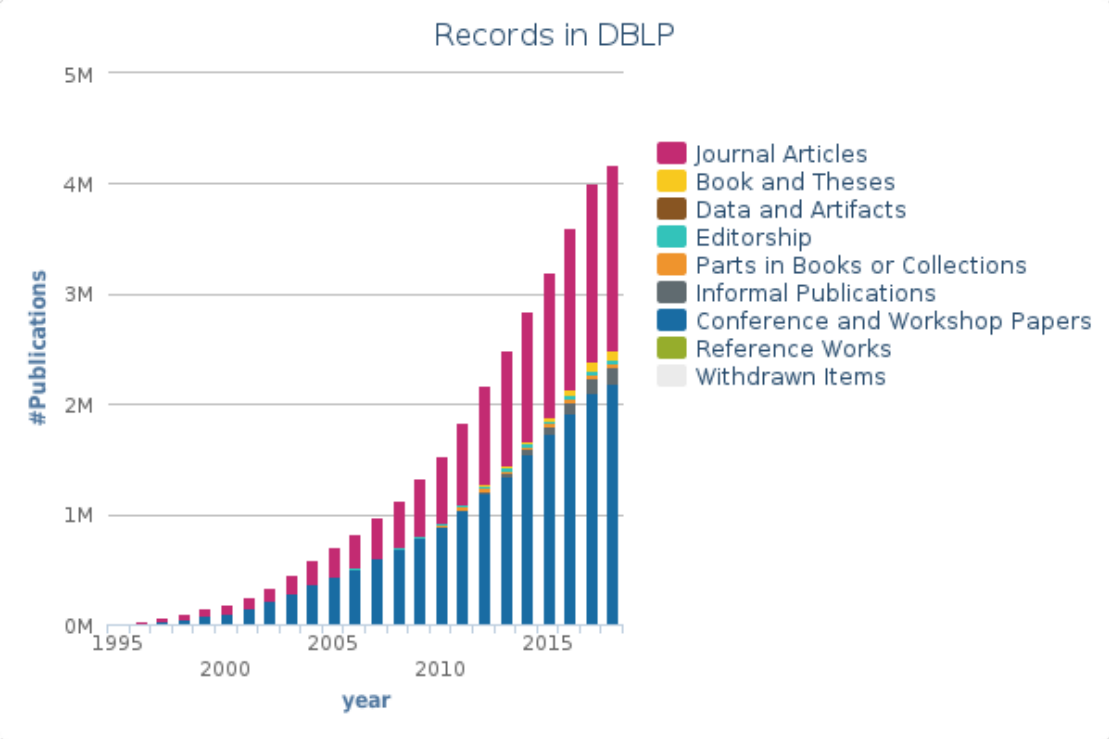


Figure 1.7.: DBLP publications upto the year 2017 [1]

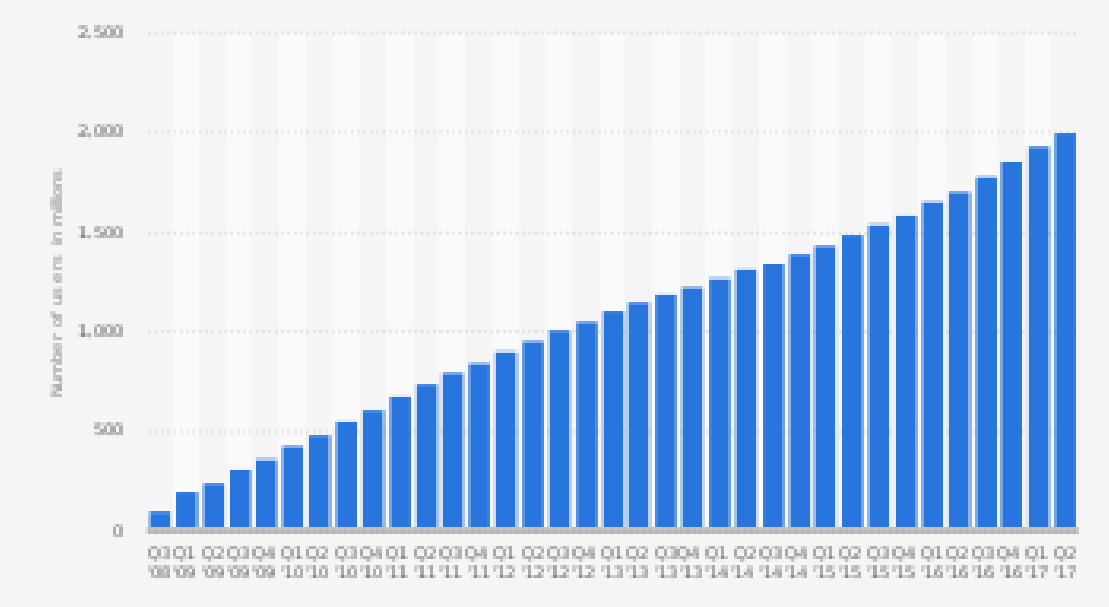


Figure 1.8.: Number of monthly active Facebook users worldwide as of 2nd quarter of 2017 (in millions)[2]

.

Link prediction is a highly unbalanced problem since the number of possible edges is of $O(|V||V-1|)$, where as the actual number of existing edges is very few. Network sparsity and heterogeneity are other challenges for link prediction. Moreover, different types of links are correlated, and are better predicted collectively instead of independently.

1.2. Applications

Link prediction problem has potential significance in many fields. In the case of biological networks very few interactions are known to us. For instance, 80% of the atomic collaborations in cells of yiest and 99.7% of human are as yet obscure [9, 10]. Link prediction techniques can be used to find the most probable interactions in such cases, and can reduce experimentation cost significantly. In friendship networks like Facebook and LinkedIn, link prediction can be used to suggest friends. In e-commerce sites such as Amazon, products such as movies, music, books, news, web pages can be recommended to the users by predicting links between user nodes and item nodes in a user-item bipartite graph [11]. Link prediction in coauthorship networks like DBLP can suggest potential collaborations. Relationships in malicious networks can be more specifically focussed by security mechanisms if information about probable links in such networks are predicted [12]. In the field of epidemiology the link prediction measures can be used to predict the spread of disease and plan interventions to diminish it [13, 14]. In transportation domain, these measures can be used to plan the additional routes based on the travel needs of people [15]. Transportation networks are large and need scalable solutions.

1.3. Motivation

Majority of link prediction measures in the literature utilize the properties of nodes, edge attributes and the topological features of the network to predict the future links.

Very few solutions have been offered in the literature for heterogeneous link prediction. Many solutions available in the literature treat all the types of links as equal or suppress the heterogeneous information available in the network. We propose that there is a lot of scope for improving the accuracy of existing methods by making use of heterogeneous information available in the network. With this motivation, we extend the existing measures to heterogeneous and bipartite environments.

One approach for finding the probability of future link formation is use of Probabilistic Graphical Models (PGM). In PGM's the nodes are treated as random variables and a link between a pair of nodes corresponds to a high co-occurrence probability of the two nodes. Here, the link prediction problem is

translated into finding the pairs of nodes having high co-occurrence probability (COP), given observations about the other nodes. COP outperforms all the topological measures for link prediction in the literature. Hence it will be interesting to extend this measure to heterogeneous networks.

The temporal behaviour of nodes and links play a major role in future link prediction. Usage of the static features does not provide complete information for predicting future links. Hence in this thesis, we attempt to utilize the temporal information of links for link prediction. With this motivation, we develop new measures by incorporating the available temporal information of the existing links.

Scalability is a major issue for analysing social networks. In another contribution, we address the problem of scalability by utilizing the structure of social networks. Social networks exhibit a natural community structure. Large complex networks are sparse as a whole and dense within community. We utilise this community membership of nodes and propose a scalable approach to link prediction.

1.4. Contributions

The contributions made in this thesis are :

1. Extended standard standard link prediction measures available for homogeneous environment to **bipartite** and **heterogeneous** environments.
2. Explored the efficiency of the Probabilistic Graphical Models for predicting future links in social networks by extending COP to **heterogeneous** environment.
3. Proposed a **new measure named Temporal Co-occurrence Probability** measure (TCOP) for link prediction, which uses temporal information available in network and evaluated on real world coauthorship networks. Also extended some of the temporal measures available in the literature to heterogeneous environment. Extensive evaluation of the proposed measures has been carried out.
4. Addressed the challenging issue of scalability in social networks by proposing **a community based Approach** to Link Prediction and evaluated its efficiency on a synthetic as well as real-world networks.
5. Link prediction approach is applied to Recommender Systems domain for recommending items to users.

1.5. Organization of the Thesis

The organization of the thesis is as follows:

Chapter 2 gives the definition of link prediction problem and gives the details of various link prediction measures for homogeneous and heterogeneous networks proposed in the literature. Further, the benchmark datasets and evaluation measures used for experimentation are presented.

Chapter 3 motivates the need for considering heterogeneous information of the network for link prediction task. Extensions of the standard link prediction techniques to bipartite and heterogeneous environments are proposed.

Chapter 4 gives the preliminaries of Probabilistic Graphical Models as well as the computation of the probabilistic measure called Co-occurrence probabilistic measure (COP). The extension of COP to heterogeneous environment is the main contribution in this chapter.

Chapter 5 proposes a new temporal probabilistic measure called Temporal Co-occurrence Probabilistic measure (TCOP) and evaluates the measure for four coauthorship networks. Further, some of the existing temporal measures for link prediction are extended to heterogeneous environment.

Chapter 6 proposes a scalable approach based on the network structure. The proposed method called Community based Link Prediction (*CBLP*), discovers communities of the network and computes prediction scores within each community in parallel. The community based approach is evaluated on benchmark synthetic as well as DBLP networks.

Chapter 7 addresses the recommender systems problem using link prediction approach. The proposed link prediction measures are evaluated on the bench mark MovieLens dataset.

The thesis ends with the conclusions drawn from the experimentation and discusses extensions to link prediction problem.

2. Background and Related Work

The problem of link prediction is to predict the possible formation of link between candidate pairs of nodes in the network. Numerous measures have been proposed in the literature that compute the likelihood of link formation between a pair of nodes [16, 8, 17, 18]. These measures can be used in unsupervised as well as supervised frameworks for link prediction. In this chapter, the standard measures for link prediction available in the literature are explored. The details about the datasets used for experiments, the experimental setup that are used in this thesis are given. The link prediction evaluation metrics are given in further sections.

2.1. Link Prediction Problem

Liben Nowel et al. [8] are the first researchers to define link prediction problem. The problem of **link-prediction** is defined as follows : Given a social graph $G(V,E)$ with vertex set V and edge set E at time t, the link prediction task is to yield a set of edges not present in $G[t_0,t_i]$ but rather are anticipated to appear in $G[t_j]$ for $t_0 < t_i < t_j$. The problem is depicted in Fig.2.1.

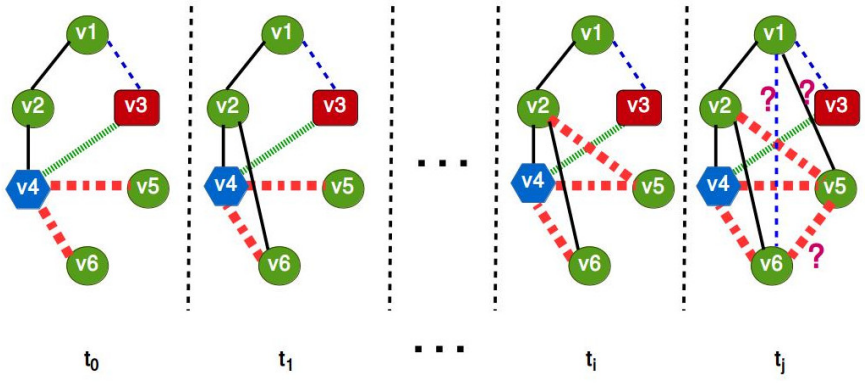


Figure 2.1.: Link Prediction Problem

Commonly, the process of link prediction has the following steps.

1. Assign a score that computes the likelihood of link formation to every pair of nodes without an edge between them [8, 17, 18].
2. Sort the pair of nodes by their scores and output the top k nodes as the list of predicted links.
3. Assess the prediction performance.

The link prediction measures available in the literature for homogeneous, heterogeneous and temporal networks are explained in the following sections.

2.2. Link Prediction Measures for Homogeneous Networks

The nodes in a social network are entities having some attributes such as the profile information in facebook network, professional information in LinkedIn network and research interests, keywords and demographic information in coauthorship networks. The attributes of a node(user) are represented as a vector, and the distance such as Euclidean or cosine may be used to compute the similarity between the nodes [19, 20, 21]. There are two challenges in such measures. One is that the attributes are domain dependent and the other is privacy. The domain dependent measures cannot be generalized to all types of networks and the node attributes are not made public many times due to privacy issues.

Therefore, another class of measures based on structural similarity of nodes in the graph became popular. These domain independent measures are classified into three categories: Topological, Probabilistic and Linear Algebraic methods [16]. These measures are discussed in the following sections.

2.2.1. Topological Measures

Topological measures are categorized into three types: Neighbourhood based, path based and random-walk based.

2.2.1.1. Neighbourhood based measures

- **Common Neighbours(CN):** It is common intuition that the probability of a link formation increases, if two nodes have many neighbours in common. This is a simplest measure which counts the neighbourhood overlap between the two nodes x and y. The definition of $CN(x,y)$ is given in Eq.2.1.

$$CN(x,y) = |\Gamma(x) \cap \Gamma(y)| \tag{2.1}$$

where $\Gamma(x)$ is the set of neighbours of x. It is obvious that $CN(x,y) = A^2[x][y]$, where A is the adjacency matrix of graph G.

- **Jaccard Coefficient(JC):** Jaccard Coefficient is the fraction of common neighbours out of the total number of neighbours of nodes x and y. This measure defines the likelihood of a neighbour of x to be a neighbour of y and vice versa. $JC(x,y)$ is defined as follows.

$$JC(x,y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|} \tag{2.2}$$

- **Salton Index (SI):** Salton et al. [22, 23] describe another normalization to CN defined as follows :

$$SI(x,y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{\sqrt{|\Gamma(x)|.|\Gamma(y)|}} \tag{2.3}$$

  Salton et al.[22] use this index in the field of information retrieval to find the document similarity using vector space model. Salton index is also called as cosine index and is equivalent to the cosine angle between rows of adjacency matrix having nodes x and y.

- **Sørensen Index(SSI):** This index is utilized primarily for ecological community data to measure similarities among two samples in species. The matches in species composition between the two samples are given more weight than mismatches [24]. The index is defined as follows:

$$SSI(x,y) = \frac{2|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x)| + |\Gamma(y)|} \tag{2.4}$$

  Justification for the use of SSI is empirical rather than theoretical. Sørensen Index is also called ad $Dice - Index$. SSN can be written using vector operations as below:

$$SSI(x,y) = \frac{2(A[x].A[y])}{A[x]^2 + A[y]^2} \tag{2.5}$$

  where A[x] and A[y] are the rows corresponding to the nodes x and y in the adjacency matrix A of G.

- **Hub Promoted Index (HPI)** [25]: This measure is proposed for evaluating the topological cover of pairs of sub traces in metabolic systems. The nodes with large topological overlap are expected to be biologically interesting modules. HPI is defined as

$$HPI(x,y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{min\{|\Gamma(x)|, |\Gamma(y)|\}} \tag{2.6}$$

The edges incident to high degree nodes are assigned high scores in HPI.

- **Hub Depressed Index (HDI):** This index is also defined in terms of HPI and is defined as

$$HDI(x,y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{max\{|\Gamma(x)|, |\Gamma(y)|\}} \tag{2.7}$$

The edges incident to high degree nodes are depressed in HDI.

- **Leicht-Holme-Newman Index1 (LHN)** [26]: LHN index assigns high score to the node pairs that have many common neighbours compared to the possible number of neighbours of each. LHN is defined as follows:

$$LHN1(x,y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x)| * |\Gamma(y)|} \tag{2.8}$$

- **Preferential Attachment (PA)** [27]: It is believed that in social graphs, nodes with highest degree tend to connect to other nodes of high degree in future. PA is computed by multiplying the degrees of node x and y and is defined as follows:

$$PA(x,y) = |\Gamma(x)| * |\Gamma(y)| \tag{2.9}$$

This measure does not need any information other than the degree of the nodes. Therefore, PA has the lowest computational complexity.

- **Adamic-Adar (AA)** [28]: This measure gives importance to the common neighbours with low degree. The measure AA is developed based on shared items on web pages. If two people have many things in common, they are more likely to be friends. Additionally, rare (special) shared items, contribute more to the connection formation. AA is defined as follows:

$$AA(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{log(|\Gamma(z)|)} \tag{2.10}$$

- **Resource Allocation (RA)**[29]: RA is motivated by the resource allocation flow in graphs. Consider a pair of nodes, x and y, without an edge between them. Let the edges connecting the common neighbours of x and y be transmitters to send some resource from x to y, and transmitters equally distribute the resource to all their neighbours. Then the similarity between x and y is the amount of resource y receives from x. RA is defined as:

$$RA(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{|\Gamma(z)|} \tag{2.11}$$

2.2.1.2. Path based measures

Path based measures are global similarity measures.

- **Katz (*KZ*)** [30]: This measure sums the number of paths between x and y of lengths between 2 and a given upper path length limit L. As the length of the path increases, the information flow between the nodes weakens. That is why, Katz measure uses a damping factor β whose value is between 0 and 1 to damp the longer paths. The definition of KZ is given below.

$$KZ(x,y) = \sum_{l=2}^{L} \beta^l |path_l(x,y)|^l \tag{2.12}$$

  where $path_l(x,y)$ is a path consisting of homogeneous edges of length l between x and y. Katz measure is computationally expensive. KZ score between all the pairs of nodes can be computed by finding $(I - \beta A)^{-1} - I$, where A is the adjacency matrix and I is an identity matrix. This formalism has cubic complexity.

- **SimRank (*SR*)** [8]: SimRank is based on the intuition that two nodes are similar if they are connected to similar nodes and is defined as:

$$SIM(x,y) = \beta \frac{\sum_{a \in \Gamma(x)} \sum_{b \in \Gamma(y)} SIM(a,b)}{|\Gamma(x).\Gamma(y)|} \tag{2.13}$$

  where β is damping factor with $0 < \beta < 1$ and $SIM(x,x) = 1$.

2.2.1.3. Random-walk based measures

- **Hitting Time (*HT*)** [8]: HT is a random-walk based measure, starts at a node x and recursively moves to a random neighbour of x. The hitting time $HT(x,y)$ is the expected number of steps for a random-walk beginning at x and ending at y.

- **Average Commute Time (*CT*)**: Average Commute Time is a symmetric metric, in which the random-walks from x to y and vice versa are added.

$$CT(x,y) = -(HT(x,y) + HT(y,x)) \tag{2.14}$$

- **Page Rank (*PR*)** [8]: PR represents the significance of x in the network based on the significance of the other nodes that are adjacent to it. PR score between two nodes x and y is the probability of

y returning to x with a probability α in a random-walk in each step, moving to a random neighbour with probability $1-\alpha$. The recursive definition of Page Rank is given as below.

$$PR(x) = \frac{1-\alpha}{M} + \alpha \sum_{z \in \Gamma(x)} \frac{PR(z)}{|\Gamma(z)|} \tag{2.15}$$

where M is the total number of edges in G.

- **PropFlow (*PF*)** [18]: *PF* is the probability of a random-walk starting at node x ends at y within l steps. A recursive definition of *PF* is given by the equation below.

$$PF(x,y) = \sum_{l=2}^{L} \sum_{p \in paths_l(x,y)} \sum_{\forall (z_1,z_2) \in p} PF(z_1,z_2) \tag{2.16}$$

If there is an edge between x and y, then $PF(x,y)$ is given by

$$PF(x,y) = PF(a,x) * \frac{w(x,y)}{\sum_{z \in \Gamma(x)} w(x,z)} \tag{2.17}$$

where a is previous node of x on random-walk, $PF(a,x)$=1 if x is starting node and $paths_l(x,y)$ is set of homogeneous paths of length l between x and y.

2.2.2. Probabilistic Measures

The probabilistic approach for link prediction has been addressed by Wang et al. [31] by proposing a probabilistic measure called Co-occurrence probability measure (COP) for homogeneous networks. The hidden advisor-advisee relation between the authors in a coauthorship network has been mined by Wang et al. in [32]. Kashima et al. [33] propose a method to reduce the problem of network evolution to link prediction. Clauset et al.[34] propose a probabilistic model based on hierarchical structure of the network. The learning algorithm utilizes the data accessible on existing links and infers the most likely hierarchical structure through statistical inference.

2.2.3. Linear Algebraic Measures

The linear algebraic methods work on the adjacency/Laplacian matrix of G. A graph kernel based method using dimensionality reduction techniques has been proposed by Kunegis et al. in [35]. A spectral transformation is performed on the adjacency matrices of training(say A_{train}) and test set(say A_{test}) which minimizes the error between the predictions in training set and test set given in the following optimization

problem:

$$min_F ||F(A_{train}) - A_{test}||_F \tag{2.18}$$

subject to $F \in S$. Here, $||.||_F$ denotes Frobenius Norm.

Specifically, the function F takes a matrix as input and outputs another matrix which is used for link prediction. The entries in the output matrix contain the similarity score between the corresponding pair of nodes. Many graph kernel methods such as Exponential kernel, Von-Neumann kernel and Laplacian kernel can be used as the function F. Li et al. [36] define a random-walk based kernel function to capture the similarity between two types of nodes and predict heterogeneous links in a bipartite network.

These matrix based measures can be naturally extended to multi-relational networks by representing multiple types of links in the form of tensors. Dunlavy et al. [37] represent the multi-relational network as a third order tensor and propose a mechanism for collapsing the tensor to matrix and use Katz method to predict links on the matrix. But the tensor based methods are global and time consuming.

These models are computationally prohibitive for large networks.

2.2.4. Temporal Measures

There are a few measures in literature, which consider temporal information for link prediction [38],[37], [39],[40],[41]. In [32], Wang et al. construct a time-constrained probabilistic factor graph model (TPFG), for a collaboration network and discover a new relation between two nodes. Selection of new relation is often application dependent. A mechanism for dynamic link inference in heterogeneous networks using temporal information has been presented in [42]. This method does not consider the weight of links for prediction.

A temporal measure called Time-score is defined in [43] for homogeneous networks. Time-score is an extension of Common neighbour measure. The authors utilize Time-score as an unsupervised measure to perform link prediction. The results obtained by the authors show that Time-score measure predicts the future links more accurately, compared to common neighbourhood based measures. Traditional neighbourhood based methods cannot differentiate two pairs of nodes that share same number of common neighbours, but having different likelihoods of link formation.

The authors of [44] propose a path based method called Link-score, which uses the temporal information available on links. They develop a Time Path Index, which models the path strength based on time stamp of links in paths. The experimental results of [44] show that Link-score has higher accuracy compared to Common Neighbourhood based measures, Katz and Time-score. However, the limitation of path based measures is that they suffer from high execution time.

A temporal random-walk based method called T_Flow is proposed in [45], which is extension of Propflow [18]. T_Flow computes information flow between nodes by considering link activeness which varies over time. Link-score and T_Flow consider either path or random-walk between the two nodes to predict the probability of link formation between them.

2.3. Link Prediction Literature for Heterogeneous Networks

Davis et al. [46], Lichtenwalter et al. [47] and Han et al. [41] propose solutions for link prediction problem for multi-relational networks. Benchettara [48] et al. predict future links in a bipartite graph by constructing homogeneous projections of the bipartite graph over one of its node sets and applying traditional link prediction methods on the projected graph. Li et al. [36] define a random-walk based kernel function to define the similarity between two types of nodes and predict heterogeneous links in a bipartite graph. Third order tensors represent the heterogeneous networks efficiently. Dunlavy et al. [37] represent the heterogeneous network as a third order tensor and propose a mechanism for collapsing the tensor to matrix and use Katz method to predict links on the matrix. But the tensor based methods are global and time consuming. Cold-start link prediction problem is proposed by Leroy et al.[49]. They predict homogeneous links using the heterogeneous information available in the network.

A meta-path based similarity between two nodes of same type in a heterogeneous graph is defined in [50]. Meta-path is a sequence of successive heterogeneous edges between two nodes of same type. They propose a measure called *PathSim* between two nodes x and y as the fraction of number of meta-paths between x and y among total paths between x and y. Meta-path selection is a major problem in meta-path based approach. Commonly meta-paths are selected using one of these ways: User may explicitly specify a meta-path combination, best path can be chosen by experiments or training instances can suggest a meta-path. An application of meta-path based approach on bibliographic networks and drug target predictions in chemical networks can be found in [51] and [52] respectively. Various meta-path based similarity measures like PathSim, random-walk and HeteSim have been proposed for link prediction in heterogeneous networks. In [51], a mutual information model for link prediction in heterogeneous networks has been presented.

The summary of the link prediction measures existing in the literature are given in Table.2.1.

Table 2.1.: Link prediction measures in the literature

Type of Networks	Unsupervised	Supervised
Homogenous	CN [8] JC [8] AA [28] PA [27] TS [43] PR [53] RPR [8] KZ [30] PF [18]	HPLP [18]
Multi-relational	MRLP [46] MRIP [41] VCP [47]	MR-HPLP [46]
Heterogeneous	Path-predict [51] Path-sim [50] DynaLink [42]	-

2.4. Supervised Framework for Link Prediction

It is found that no one single measure works equally well for all the datasets. Therefore, a supervised framework is proposed by Hasan et al.[17], where the strength of all these measures can be effectively utilized. The process of supervised learning for link prediction is shown in Fig.2.2.

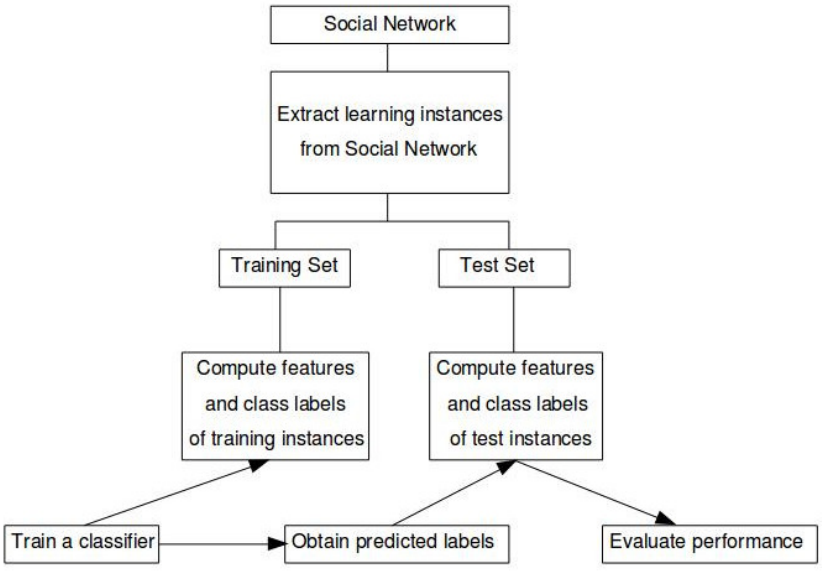
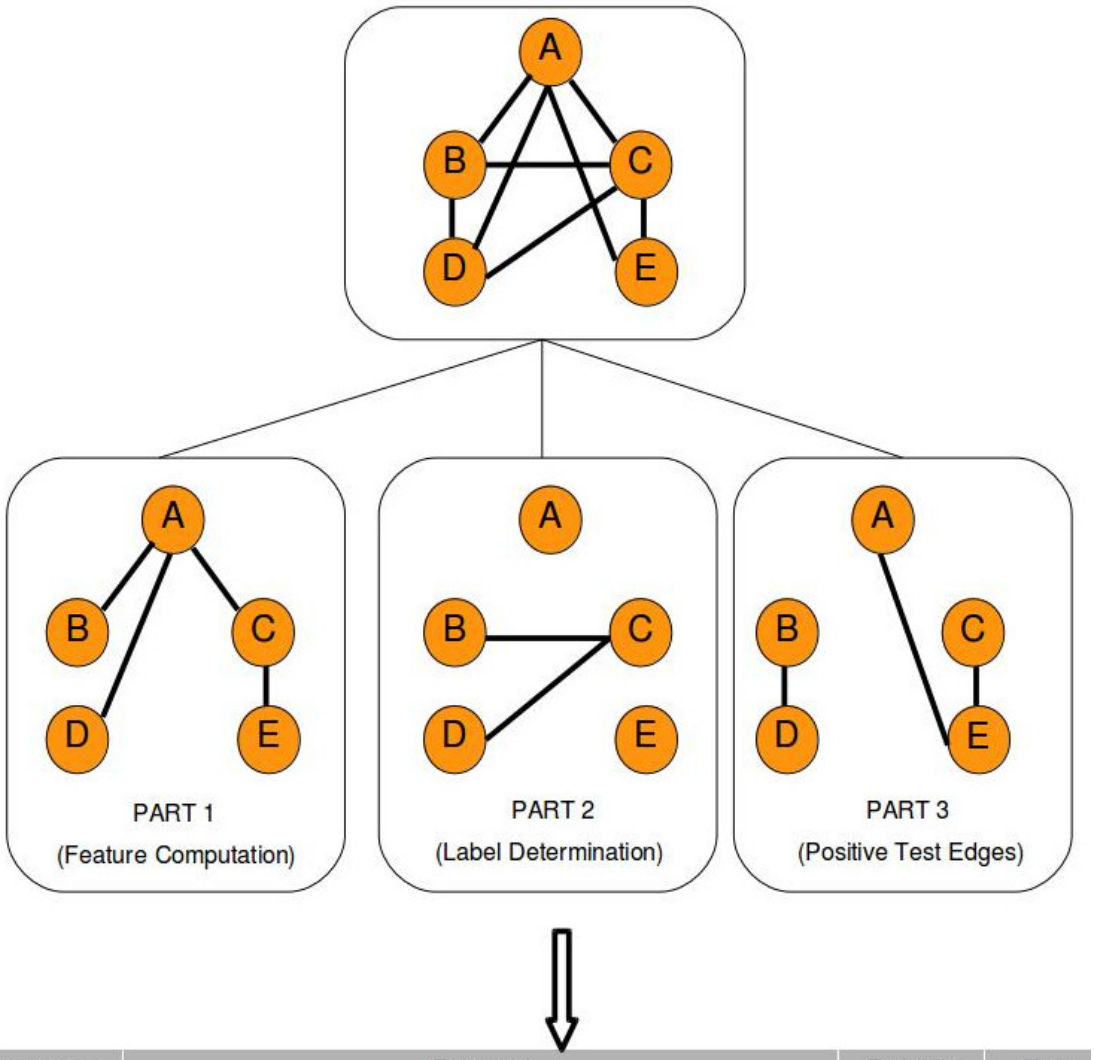


Figure 2.2.: Supervised setting for evaluating link prediction performance

The steps to solve link prediction problem using supervised classification framework are elaborated below:

- **Representation of learning instances**: The learning instances are taken as pairs of nodes of the graph.
- **Feature vector representation**: The feature vector is constructed for the instances. The vector contains a number of features that describes the instance. Hasan et al. [17] use neighbourhood based, path based and semantic measures as features. But the semantic features are domain dependent. Domain dependent features may not be available for all the networks due to the issues of privacy. Litchenwalter et al. [18] use all domain independent features for construction of feature vector and named the approach as High Performance Link Prediction (HPLP). They recognize link prediction problem as an unbalanced classification problem and design a method for constructing training and test sets for unbalanced datasets, which is adopted in this thesis.
- **Construction of Training and Test sets of the given network**[31]: To construct the training and test sets with class labels, the network is divided into three parts, Part1, Part2 and Part3. For all the node pairs (not connected by an edge), the features are calculated using the network of Part1. The class label of the instance will be **1** if there is link between the nodes in Part2, **0** otherwise. The test set is also constructed in the similar way by taking labels from Part3 [54]. If the time of formation of links is available, the Part3 may be taken as the network pertaining to the current year (year of prediction) is taken as the test set, Part2 consists of the network upto previous year and Part1 contains the rest of the entire network. When the time of formation of links is not available, Part1 consists of 80% of random links, Part2 and Part3 each may be taken as remaining 10% random links. The procedure is depicted in Fig.2.3.
- **Learning (Classification) algorithm**: Once the training and test sets and feature vectors are available, any classification algorithm can be used to obtain class labels of instances which are not in training set. For link prediction, the features play important role in determining class labels. Therefore, the focus will be on investigating novel features rather than on classification algorithm used.
- **Handling imbalance**: Link prediction is a highly unbalanced problem with the number of possible candidate pairs being tested for link prediction is $O(|V|^2)$ where as the number of existing edges may be of $O(|V|)$. Lichtenwalter et al. [18] use undersampling along with bagging to handle imbalance, which is followed by many researchers.



Instances	Features	Training Labels	Test Labels
(A,E)	(CN(A,E), JC(A,E), AA(A,E), PA(A,E), KZ(A,E), PR(A,E), PF(A,E) ...)	0	1
(B,C)	(CN(B,C), JC(B,C), AA(B,C), PA(B,C), KZ(B,C), PR(B,C), PF(B,C) ...)	1	0
(B,D)	(CN(B,D), JC(B,D), AA(B,D), PA(B,D), KZ(B,D), PR(B,D), PF(B,D) ...)	1	1
(B,E)	(CN(B,E), JC(B,E), AA(B,E), PA(B,E), KZ(B,E), PR(B,E), PF(B,E) ...)	0	0
(C,D)	(CN(C,D), JC(C,D), AA(C,D), PA(C,D), KZ(C,D), PR(C,D),PF(C,D) ...)	1	0
(D,E)	(CN(D,E), JC(D,E), AA(D,E), PA(D,E), KZ(D,E), PR(D,E), PF(D,E) ...)	0	0

Figure 2.3.: Process of splitting a network into train and test set.

- **Performance evaluation**: After obtaining the class labels of instances using training set and label set, the performance of the labels obtained is measured on the test set constructed above. As link prediction is modelled as a binary classification problem, the popular evaluation metrics are Receiver Operating Curve (ROC), Precision- Recall Curve (PRC) and the area under these curves. The details of these metrics are discussed in section.2.5.

The above framework which we term as *HPLP* framework given by Lichtenwalter et al. in [18] is adopted by us for the entire experimentation in the thesis.

2.5. Performance Evaluation Measures

Link prediction is modelled as a binary classification task. The possible outcomes of binary classification fall into four categories as shown in Confusion Matrix(Table 2.2).

Table 2.2.: Confusion Matrix

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

Based on the TP, FP, TN and FN, the following are the standard measures of evaluation. [55]

$$Recall = TPR = \frac{TP}{TP+FN} \tag{2.19}$$

Recall also known as Specificity/True Positive Rate(TPR) is the fraction of positive instances predicted out of total available positive instances.

$$Precision = \frac{TP}{TP+FP} \tag{2.20}$$

Precision gives the fraction of positive instances predicted correctly out of total instances predicted as positive.

$$FPR = \frac{FP}{FP+TN} \tag{2.21}$$

False Positive Rate(TPR) is the fraction of negative instances predicted out of total available negative instances.

2.5.1. AUROC

It is conventional to use Area Under Receiver Operating Characteristic Curve(AUROC) [56] to evaluate the performance of a classifier. ROC curve is drawn by taking FPR on x-axis and TPR on y-axis. An example illustration of the above procedure is depicted on a sample social graph given in Fig.2.4, Table.2.3 and the corresponding ROC-curve is shown in Fig.2.5.

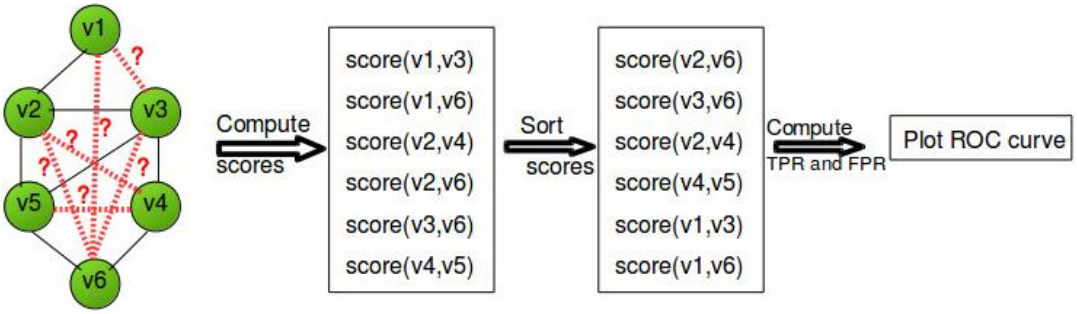


Figure 2.4.: Plotting ROC curve for unsupervised measures for Link Prediction

Table 2.3.: Illustration of ROC curve assuming the scores between the pairs of nodes.

Node pair	Score in Sorted order	Actual Link Exists (Yes/No)	TP	TN	FP	FN	FPR	TPR	Point in ROC Curve
(v2, v6)	0.8	Yes	1	3	0	2	0	1/3	(0, 0.3)
(v3, v6)	0.7	Yes	2	3	0	1	0	2/3	(0, 0.6)
(v2, v4))	0.6	No	2	2	1	1	1/3	2/3	(0.3, 0.6
(v4, v5)	0.5	Yes	3	2	1	0	1/3	1	(0.3, 1)
(v1, v3)	0.4	No	3	1	2	0	2/3	1	(0.6, 1)
(v1, v6)	0.2	No	3	0	3	0	3/3	1	(1, 1)

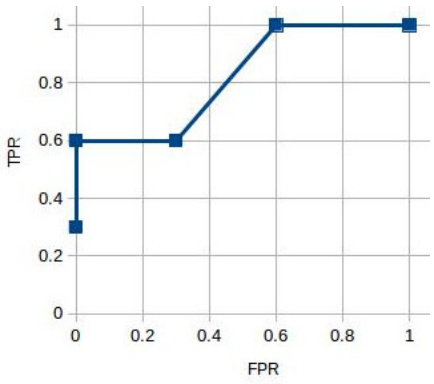
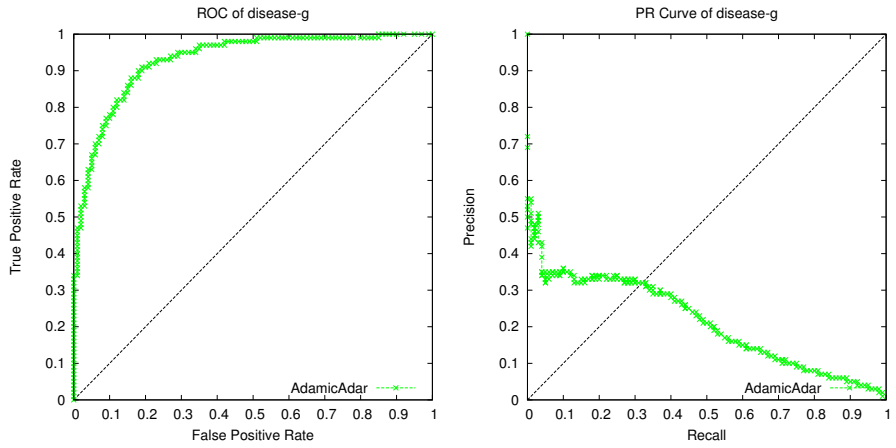


Figure 2.5.: ROC curve corresponding to Table 2.3 .

2.5.2. Significance of AUPR for unbalanced problems

AUROC gives the expected proportion of positives ranked before a uniformly drawn random negative. Link prediction is highly unbalanced. In the case of extremely unbalanced data, ROC curve may provide an overly optimistic view of a classifier's performance [57]. In that scenario, the Precision Recall curves(PR curves)[58] can provide more informative representation of assessing performance [59]. In case of unbalanced datasets, as the majority class samples outnumbers the minority class samples, the drastic change in false positive rates could not be captured by ROC curves due to the large denominator of eq.(2.21).



(a) ROC curve (b) PR curve

Figure 2.6.: ROC and PR curves of disease-g dataset

On the other hand, as the denominator of precision is the combination of both TP and FP, it could capture the changes in false positives. For example, Fig. 2.6 depicts the AdamicAdar score for disease-g dataset given in the tool lpmade [60]. It can be observed that there is no much scope for improvement in the ROC space. But there is still a room for improvement in PR space. The objective of ROC curves is to be in the upper left hand of the ROC space, a dominant Pr curve resides in the upper right hand of the PR space. However, an algorithm that optimizes the AUC in the ROC space does not guarantee to optimize the AUC in PR space. Hence, PR space is an effective evaluation technique and it has all the characteristics and analogous benefits of ROC.

In this work, ROC curve and PR curves are used to evaluate the performance of proposed link prediction algorithms.

2.6. Datasets

The datasets used in this work are as specified below.

2.6.1. Synthetic dataset

A synthetic dataset is designed by Tang et al, in [61]. The network is multi-relational dataset with 350 nodes and four types of relations among nodes. For each relation type, nodes connect with each other following a random generated within-group interaction probability. The interaction probability differs with groups at distinct relations. After that, some noise is added to the network by randomly connecting any two anodes with low probability. The network statistics are shown in table 2.4.

Table 2.4.: Synthetic data set statistics

Edge Type	**#Nodes**	**#Edges**
Relation1	350	12430
Relation2	350	16850
Relation3	350	15756
Relation4	350	15206

2.6.2. Real world benchmark datasets

1. **DBLP** [31] : DBLP is the benchmark dataset given in [31], which consists of research publications of 28 conferences in the fields of Data Mining, Databases and Machine Learning held during the years 1997 to 2006.
2. **Condmat** [18] : Condmat is a collaboration network consisting of 23,709 authors with 1,10,544 papers in the area of condensed matter physics from 1995 to 2000.
3. **HiePh-collab** [62] : HiePh-collab consists of a set of publications in theoretical High Energy Physics during the years 1992-2003.
4. **HiePh-cite** [62] : HiePh-cite is a citation network based on the High Energy Physics publications submitted to *arXiv*. Each node represents an author and edge exists between two authors i and j if author i cites the work of author j. We considered the citations during the years 1992-2003.

Table 2.5.: Real-world benchmark datasets statistics

Dataset	*Collabaration period*	$\|AuthorNodes\|$	$\|CoauthorEdges\|$
Condmat	6 Yrs	23,709	1,10,544
DBLP	10 Yrs	23,136	56,829
HiePh-collab	12 Yrs	8381	40736
HiePh-cite	12 Yrs	8249	3,35,028

2.7. Conclusion

In this chapter, various link prediction measures existing in the literature are presented. As observed, many measures are available in literature for homogeneous networks and a few are available for heterogeneous networks. Many times, the heterogeneity of the network is ignored while predicting links. But the heterogeneous information provides meaningful insights in many cases. In the next chapter, the preliminaries of heterogeneous networks are presented and some of the homogeneous measures are extended to heterogeneous networks.

3. Preliminaries of Heterogeneous Measures for Link Prediction

A social network modelled as a graph is called heterogeneous if it has multiple types of nodes and multiple types of edges. Many networks in the real world are heterogeneous in nature. For instance, a bibliographic network may have multiple types of nodes such as author, paper, conference, venue and keywords. In this chapter, notation adopted in the thesis as well as the definitions needed are presented. We extend many of the homogeneous link prediction measures to the heterogeneous environment, by extending in a natural way, the definitions of neighbourhood, path etc. appropriately.

3.1. Introduction

In a bibliographic network, two authors may be related with *coauthorship* relation; an author may *write* a paper; a paper may be *published* by a conference; an author *attends* a conference and a paper *contains* keywords. This scenario is depicted in Fig.3.1. Similarly, a movie network may have several types of nodes such as Movie, Actor, Director, Writer and Studio and different types of edges as shown in Fig.3.2.

A multi-relational(MR) network is a special type of heterogeneous network containing single type of nodes and multiple types of edges. For example in a network of users modelling the relation between the users, there can be several types of relations such as friend, relative, resident or colleague. Each type of relation is taken as a different type of edge in the example MR network shown in Fig.3.3

A bipartite network has two types of nodes and edges in such networks connect a pair of nodes of different types. For example, in e-commerce sites such as Amazon and Flipkart, users buy items. In such networks, there are two types of nodes *user* and *item* and the relation between the two types of entities is *buy*. An example bipartite network is shown in Fig.3.4.

We define a *homogeneous edge* as an edge between two nodes of same type and *heterogeneous edge* as an edge existing between nodes of different types. In Fig.3.1, *co-authorship* edges are homogeneous and the *publish* edges between paper node and conference node are heterogeneous edges. Note

Figure 3.1.: DBLP Heterogeneous Network

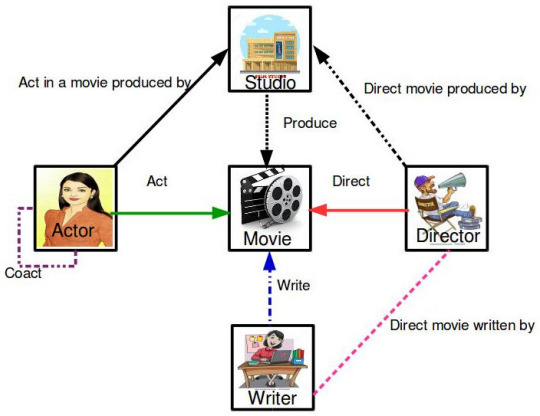


Figure 3.2.: Movie Network

that, a bipartite network contains only heterogeneous edges.

3.2. Motivation

Many link prediction measures in the literature are defined for homogeneous networks as discussed in 2.2; recently more work is seen with reference to heterogeneous networks. The structure of heterogeneous network is more complex compared to homogeneous network because of existence of multiple types of nodes and edges. The relation between two nodes of same type may be influenced by the existence of multiple types of edges between them. For example, in the multi-relational network of Fig.3.3, a friend relation may be formed between two persons because they stay in the same colony or being colleagues. Similarly in a movie network, actor collaboration may happen because of a director who directed them in two separate movies and may cast them together in an upcoming movie. This type of collaboration cannot be inferred if the homogeneous network containing only actors is used for prediction. Therefore, solving link prediction problem using homogeneous projections may not yield good results.

In this chapter, some of the available link prediction measures proposed for homogeneous networks are extended for heterogeneous networks. The proposed measures are evaluated on two benchmark bibliographic datasets of DBLP and HiePh-collab. It is interesting to find that the proposed measures show an improvement in accuracy when heterogeneous information is included.

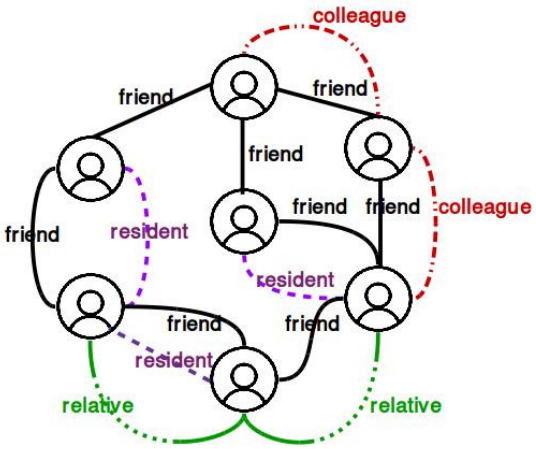


Figure 3.3.: A Multi-relational Network with single type of nodes: *user* and four types of relations : *friend*, *relative*, *resident*, *colleague*

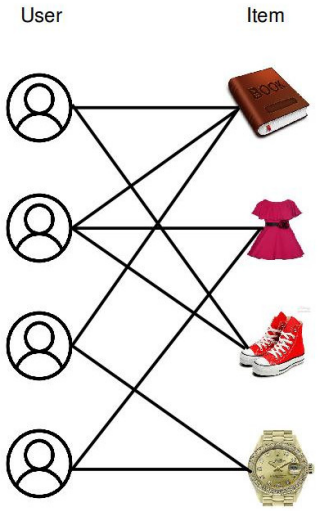


Figure 3.4.: An example of user-item Bipartite Network.

3.3. Definitions and Notation

Definition 3.1. A ***Heterogeneous Social Network*** $G = (V, E, w)$, is a graph, $V = \bigcup_{i=1}^{n} V_i$ represents n types of nodes, $E = \bigcup_{s=1}^{m} E_j$ denotes m types of edges (x, y) where $x \in V_i, y \in V_j$, $w : E \to R$, $w(x, y)$ denotes the weight of the interaction between x and y.

Definition 3.2. A **Bipartite network** contains exactly two types of nodes and single type of edges existing between different types of nodes. Bipartite network is defined as $G = (V_1 \cup V_2, \quad E)$, where V_1 and V_2 are sets of two types of nodes, E represents the set of edges between nodes of type V_1 and V_2.

Definition 3.3. A **Multi-relational network** contains same type of nodes and different types of edges and is defined as $G = (V, \quad E_1 \cup E_2 \cup \ldots E_m)$, where V is the set of nodes, $E_s, s \subseteq \{1 \ldots m\}$ represents the set of edges of type s.

Definition 3.4. A **Homogeneous network** contains nodes and edges of same type. A homogeneous social network is represented as a graph $G = (V, E)$, where V is the set of nodes, E is set of edges.

Definition 3.5. The problem of **link-prediction** is defined in 2.1. The aim of **link prediction** between a pair of nodes $x, y \in \bigcup_{i=1}^{n} V_i$ is to find the possibility of a link of type s, $s \in \{1, 2 \ldots m\}$ appearing between x

and y at a future instant of time. Graph G is a multi-relational if n=1 and m>1; homogeneous if $m = n = 1$ and bipartite if $n = 2$ and $m = 1$.

3.3.1. Notation

We follow the following notation throughout the thesis.

- (x, y) is a pair of nodes without an edge between them.
- s is used to represent type of links, k to represent hop-distance between two nodes and t to denote time.
- n is the number of types of nodes and m denotes the number of types of edges.
- **Meta-path** is a path consisting of homogeneous/heterogeneous edges. For example, in Fig.3.5(b), $a_1 - a_2$ is a homogeneous edge and $a_1 - c_1 - a_3 - c_3$ is a heterogeneous path.

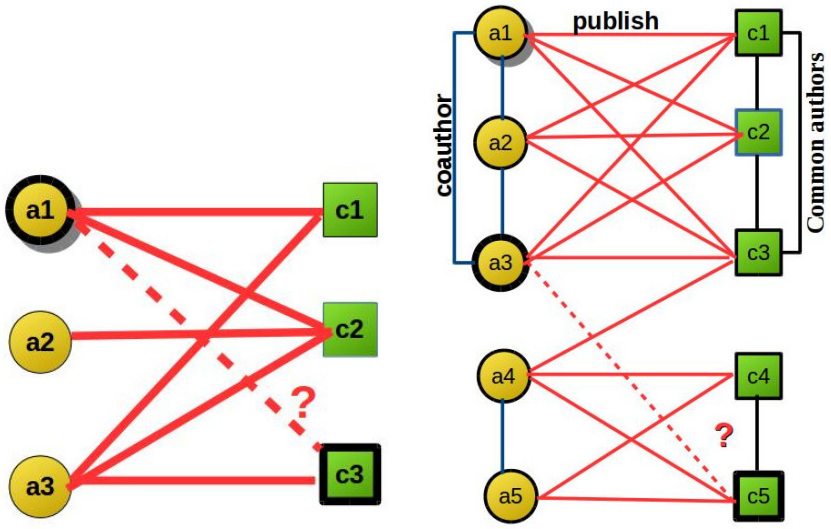


Figure 3.5.: Examples of Bipartite and Heterogeneous Networks

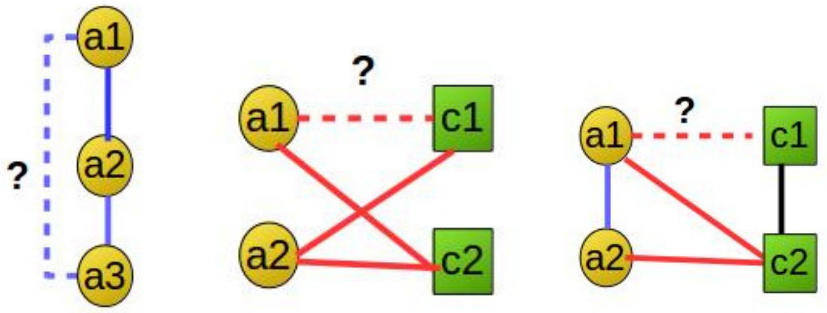
- $\Gamma_k(x)$ is the set of *k-hop* neighbours of node x. $\Gamma_1(x)$ refers to the set of all nodes connected by an edge of any type to x generally written as $\Gamma(x)$. $\Gamma(x) \cap \Gamma(y)$ denotes the set of common neighbours between node x and y, and $\Gamma_k(x) \cap \Gamma_k(y)$ contains all the common neighbours within k-hop distance between nodes x and y.

- $P_k(x,y)$ denotes the set of meta-paths connecting x and y by at most k edges.

Heterogeneous networks contain homogeneous as well as meta-paths. Preferential attachment and path based measures can be applied to heterogeneous networks, but common neighbourhood based measures cannot be applied directly.

3.4. Extensions of Link Prediction Measures to Heterogeneous Networks

Common neighbours in homogeneous networks as well as heterogeneous networks occur on paths of length 2 between the nodes. In other words, the common neighbours occur at 1-hop or 2-hop distance. For example, in Fig. 3.6(a), the node a_2 occurring on path of length 2 between the nodes a_1 and a_3, is a common neighbour. Note that in bipartite network in Fig. 3.6(b), if the edge $a_2 - c_2$ does not exist, then there is no path between the nodes a_1 and c_1. Hence for bipartite networks, paths of minimum length 3 have to be considered in Fig 3.6(b) for common neighbour computation. In heterogeneous networks, paths of length 2 as well as 3 exist through homogeneous/heterogeneous edges. Hence, to compute common neighbours in heterogeneous environment, we consider meta-paths of length ≤ 3 or neighbourhood of distance 1 or 2. This understanding leads to definitions that can be extended naturally to Bipartite/heterogeneous environment.



(a) (b) (c)

Figure 3.6.: Minimum length paths containing homogeneous edges (blue and black) and heterogeneous edges (red) between nodes in different types of networks

The baseline link prediction measures explained in detail in 2.2 are extended to heterogeneous environment as follows. For completeness sake, we repeat the definitions for homogeneous networks also below.

- **Common Neighbours (CN)** :

  The common neighbour measure in homogeneous and heterogeneous environments is given as follows:

$$\begin{aligned} CN(x,y) &= |\Gamma_1(x) \cap \Gamma_1(y)| \quad \text{in Homogeneous networks} \\ &= |\Gamma_2(x) \cap \Gamma_2(y)| \quad \text{in Bipartite/Heterogeneous networks} \end{aligned} \tag{3.1}$$

  Jaccard Coefficient, AdamicAdar and Preferential Attachment measures are also neighbourhood-based defined in a similar way as follows.

- **Jaccard Coefficient (JC)** : Jaccard Coefficient is the normalized *CN* measure by considering extended neighbourhoods, Γ_2.

$$\begin{aligned} JC(x,y) &= \frac{|\Gamma_1(x) \cap \Gamma_1(y)|}{|\Gamma_1(x) \cup \Gamma_1(y)|} \quad \text{in Homogeneous networks} \\ &= \frac{|\Gamma_2(x) \cap \Gamma_2(y)|}{|\Gamma_2(x) \cup \Gamma_2(y)|} \quad \text{in Bipartite/Heterogeneous networks} \end{aligned} \tag{3.2}$$

- **Adamic Adar (AA)** : This measure gives importance to the common neighbours with low degree. The following definition for bipartite networks has been hinted at [46].

$$\begin{aligned} AA(x,y) &= \sum_{z \in \Gamma_1(x) \cap \Gamma_1(y)} \frac{1}{log(|\Gamma_1(z)|)} \quad \text{in Homogeneous networks} \\ &= \sum_{z \in \Gamma_2(x) \cap \Gamma_2(y)} \frac{1}{log(|\Gamma_2(z)|)} \quad \text{in Bipartite/Heterogeneous networks} \end{aligned} \tag{3.3}$$

- **Preferential Attachment (PA)** : This measure does not change in heterogeneous environment because the measure is concerned only about the degree of the node whatever the type of the node may be. Therefore, PA remains the same for all homogeneous, bipartite and heterogeneous environments. But the neighbourhood in various environments may vary. For example, in heterogeneous network, x may be linked to homogeneous as well as heterogeneous edges, which contributes in computing $\Gamma_1(x)$.

$$PA(x,y) = |\Gamma_1(x)| * |\Gamma_1(y)| \quad \text{in all types of networks} \tag{3.4}$$

- **Katz (KZ)** : This measure is based on the total number of paths between x and y bounded by a limit penalized by path length. In heterogeneous and bipartite environments, meta-paths are considered

to compute the Katz score.

$$\begin{aligned} KZ(x,y) &= \sum_{l} \beta^l |paths^l_{(x,y)}| \quad \text{in Homogeneous networks} \\ &= \sum_{l} \beta^l |P_l(x,y)^l| \quad \text{in Bipartite/Heterogeneous networks} \end{aligned} \tag{3.5}$$

- **Page Rank (PR)** : This measure can be extended to heterogeneous network by including the heterogeneous edges in the random-walk.

$$\begin{aligned} PR(x) &= \frac{1-\alpha}{|E|} + \alpha \sum_{z \in \Gamma(x)} \frac{PR(z)}{|\Gamma(z)|} \quad \text{in Homogeneous networks} \\ &= \frac{1-\alpha}{|E|} + \alpha \sum_{z \in \Gamma_2(x)} \frac{PR(z)}{|\Gamma_2(z)|} \quad \text{in Bipartite/Heterogeneous networks} \end{aligned} \tag{3.6}$$

  where $|E|$ is the total number of links in G.

- **Rooted Page Rank (RPR)** : In order to make the measure PR symmetric, Rooted Page Rank is computed for an edge (x,y) as follows.

$$RPR(x,y) = PR(x,y) + PR(y,x) \tag{3.7}$$

- **PropFlow (PF)** : This is a random-walk beginning at node x and ending at y within l steps. This random-walk from node x to y terminates either when it reaches y or revisits any node. $PF(x,y)$ is the probability of information flow from x to y based on random transmission along all paths defined recursively as follows.

$$PF(x,y) = \sum_{l=2}^{L} \sum_{p \in paths_l(x,y)} \sum_{\forall (z_1,z_2) \in p} PF(z_1,z_2) \tag{3.8}$$

  where

$$PF(z_1,z_2) = PF(a,z_1) * \frac{w(z_1,z_2)}{\sum_{z \in \Gamma(z_1)} w(z_1,z)} \tag{3.9}$$

  with a as previous node of z_1 in the random-walk, $PF(a,z_1)$=1 if a is the starting node and $paths_l(x,y)$ is the set of homogeneous paths of length l between x and y.

  The heterogeneous version of PF is extended naturally by taking meta-paths.

$$PF(x,y) = \sum_{l=2}^{L} \sum_{p \in P_l(x,y)} \sum_{\forall (z_1,z_2) \in p} PF(z_1,z_2) \tag{3.10}$$

where $PF(z_1,z_2)$ is as defined in Eq.3.9 by also including heterogeneous edges.

The proposed measures are summarized in Table 3.1.

Table 3.1.: Link Prediction measures for a node pair (x,y) in various types of networks

LP	Homogeneous networks	Bipartite/Heterogeneous networks
CN	$\lvert\Gamma_1(x) \cap \Gamma_1(y)\rvert$	$\lvert\Gamma_2(x) \cap \Gamma_2(y)\rvert$
JC	$\frac{\lvert\Gamma_1(x) \cap \Gamma_1(y)\rvert}{\lvert\Gamma_1(x) \cup \Gamma_1(y)\rvert}$	$\frac{\lvert\Gamma_2(x) \cap \Gamma_2(y)\rvert}{\lvert\Gamma_2(x) \cup \Gamma_2(y)\rvert}$
AA	$\sum_{z \in \Gamma_1(x) \cap \Gamma_1(y)} \frac{1}{log(\lvert\Gamma_1(z)\rvert)}$	$\sum_{z \in \Gamma_2(x) \cap \Gamma_2(y)} \frac{1}{log(\lvert\Gamma_2(z)\rvert)}$
PA	$\lvert\Gamma_1(x)\rvert * \lvert\Gamma_1(y)\rvert$	$\lvert\Gamma_1(x)\rvert * \lvert\Gamma_1(y)\rvert$
KZ	$\sum_l \beta^l \lvert paths^l_{x,y}\rvert$	$\sum_l \beta^l \lvert P_l(x,y)^l\rvert$
PR	$\frac{1-\alpha}{m} + \alpha \sum_{z \in \Gamma(x)} \frac{PR(z)}{\lvert\Gamma(z)\rvert}$	$\frac{1-\alpha}{m} + \alpha \sum_{z \in \Gamma_2(x)} \frac{PR(z)}{\lvert\Gamma_2(z)\rvert}$
PF	$\sum_{l=2}^{L} \sum_{p \in paths_l(x,y)} \sum_{\forall (z_1,z_2) \in p} PF(z_1,z_2)$ $PF(z_1,z_2) = PF(a,z_1) * \frac{w(z_1,z_2)}{\sum_{z \in \Gamma(z_1)} w(z_1,z)}$ $PF(a,x)$=1 if x is starting node	$\sum_{l=2}^{L} \sum_{p \in P_l(x,y)} \sum_{\forall (z_1,z_2) \in p} PF(z_1,z_2)$ $PF(z_1,z_2) = PF(a,z_1) * \frac{w(z_1,z_2)}{\sum_{z \in \Gamma(z_1)} w(z_1,z)}$ $PF(a,x)$=1 if x is starting node

3.5. Experimental Evaluation

3.5.1. Dataset and Experimental Setup

The proposed measures are evaluated on two benchmark coauthorship networks of HiePh-collab and DBLP. The details of these datasets given in 2.6 are also provided in Table.3.2 for quick reference.

Table 3.2.: Datasets

Dataset	#Nodes		#Edges		
	Author	**Conference**	**Auth-Auth**	**Auth-Conf**	**Conf-Conf**
HiePh-collab	8381	255	40736	20826	2709
DBLP	23136	28	56829	35665	281

Link prediction is carried out using the proposed heterogeneous measures as individual measures in unsupervised framework as well as a set of features to form feature vector in supervised framework as explained in 2.2. Every edge is represented as an 8-length feature vector: (*Hetero-CN*, *Hetero-JC*, *Hetero-AA*, *Hetero-PA*, *Hetero-KZ*, *Hetero-PR*, *Hetero-RPR*, *Hetero-PF*). For bipartite networks, we use the bipartite versions of proposed measures.

The link prediction scores of all these measures are computed using the software tools LPmade [60] (with default parameters) with appropriate modifications made for heterogeneous environment. For evaluation using supervised framework, we adopt the basic set-up given by Wang et al. [31] for constructing train and test sets, which is given below.

Test set is composed of all the edges existing in the last year and the dataset is trained on the network except the current year. For DBLP dataset, the graph of 1997-2005 is taken as training set. The measures are computed on this train set. The performance is evaluated on the test set for pairs of nodes for which edges are formed in the year 2006 as shown in Fig.3.7.

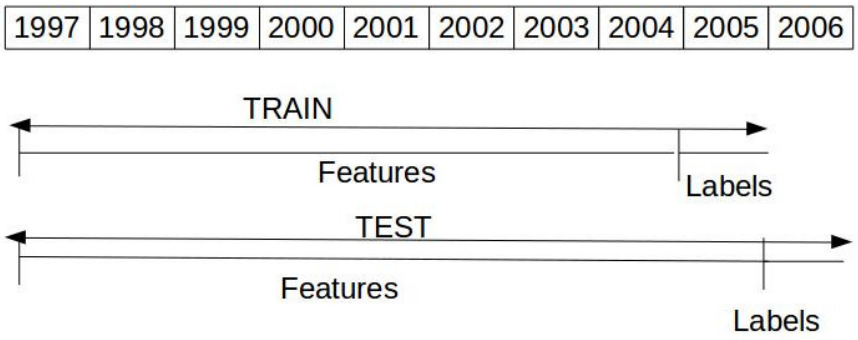


Figure 3.7.: Train-Test set split for DBLP Dataset

The same is the case with HiePh-collab dataset also. The training set is under-sampled and Bagging[63](10 bags) with Random Forest classification algorithm [64] is used. We used the tool WEKA [65] for Bagging and Random Forest algorithms.

The performance of the proposed heterogeneous measures is evaluated against with their homogeneous versions.

We predict two types of links in the coauthorship networks of DBLP and HiePh-collab

- Prediction of homogeneous links (*author-author*) to predict future collaborations of authors in homogeneous and heterogeneous networks.
- Prediction of heterogeneous links (*author-conference*) to recommend conferences to authors in bipartite as well as heterogeneous environments. For considering the bipartite network, we suppress the homogeneous links of auth-auth and conf-conf.

3.5.2. Prediction of Homogeneous links

The experiments are carried out both in homogeneous network consisting of only author nodes with coauthor links as well as in heterogeneous environment consisting of author as well as conference nodes with three types of interactions between them. When an author presents a paper in a conference, new interaction with other authors in the same conference may result in new collaborations. This translates to predicting a homogeneous link between $a_1 - a_2$ using the existing heterogeneous links $a_1 - c_1$ and $a_2 - c_1$.

The prediction results of coauthor homogeneous links in homogeneous as well as heterogeneous networks of HiePh-collab and DBLP are given in 3.8 and 3.9,Fig.3.10, 3.11 for AUPR and AUROC scores. Note that the results against *homogeneous* correspond to the measures implemented using only homogeneous links and the values in the row of *heterogeneous* use both homogeneous and heterogeneous links. The results of supervised classification performance are clearly much better than those of the individual measures.

For HiePh-collab dataset, when heterogeneous information is utilized, *KZ* shows an improvement in AUPR score from 0.0090 to 0.0193 as seen in Fig.3.8 and AUROC is improved from 61% to 64% as shown in Fig.3.9.

For DBLP dataset, an average improvement of 0.01 is observed in AUPR score when the heterogeneous linkages are used for prediction for *CN* and *AA*, in comparison to using only homogeneous links. *PA* performs best with an AUPR score of 0.1162 among all the measures and the improvement is slight when heterogeneous links are utilized. In the case of AUROC, an average improvement of 3% is observed for *CN* and *AA*. *PA* has has shown better performance with an AUROC score of 74% against others with a minute improvement in heterogeneous environment.

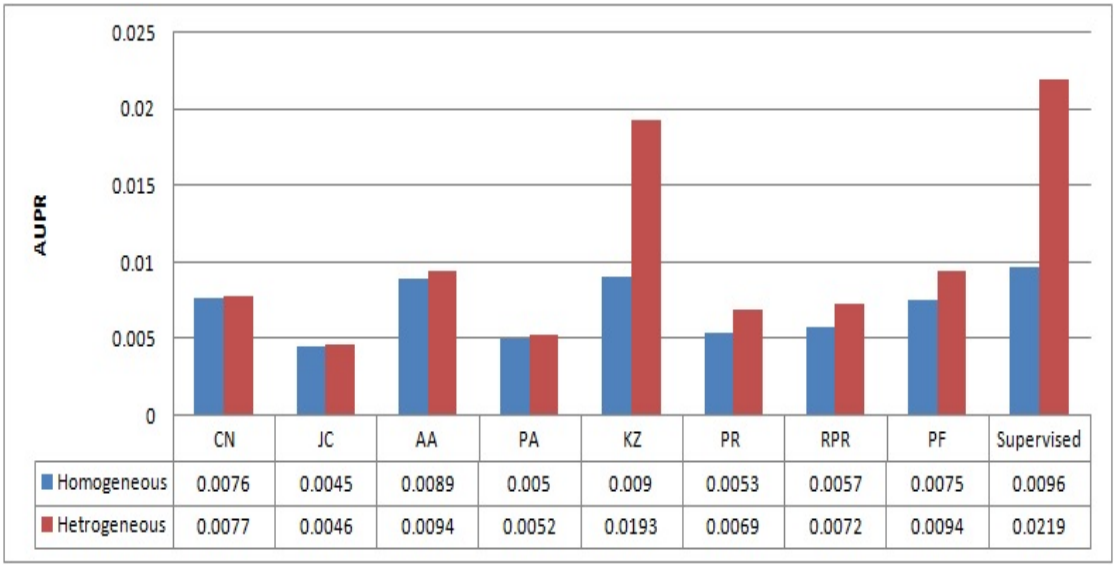
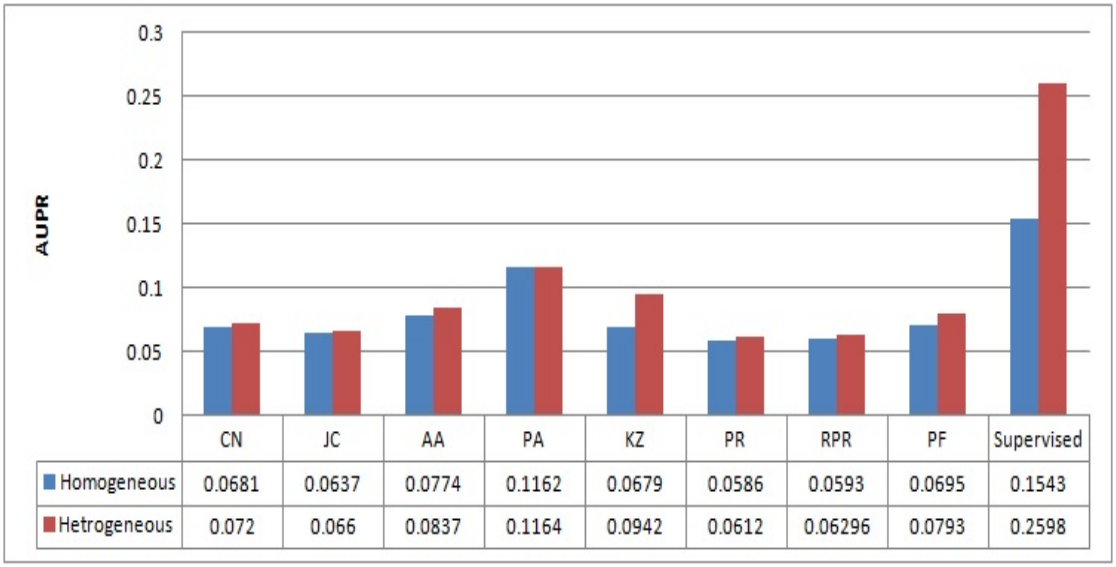


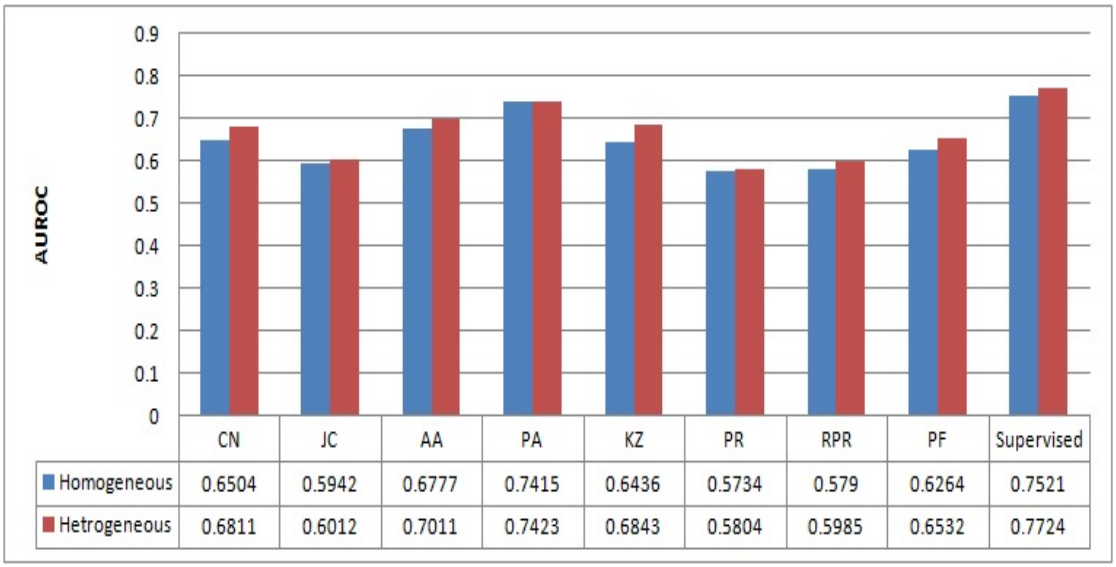
Figure 3.8.: AUPR of **homogeneous** link prediction for **HiePh-collab** network. *AA* and *KZ* have shown almost equal performance in homogeneous environment, but the performance of *KZ* has significantly improved with the use of heterogeneous information.

Figure 3.9.: AUROC of **homogeneous** link prediction for **HiePh-collab** network. *PA* and *KZ* have shown similar performance with a 2% improvement in heterogeneous environment.



	CN	JC	AA	PA	KZ	PR	RPR	PF	Supervised
Homogeneous	0.0681	0.0637	0.0774	0.1162	0.0679	0.0586	0.0593	0.0695	0.1543
Hetrogeneous	0.072	0.066	0.0837	0.1164	0.0942	0.0612	0.06296	0.0793	0.2598

Figure 3.10.: AUPR of **homogeneous** link prediction for **DBLP** network. *PA* shows better prediction performance over all topological measures and the performance improvement is very minute with the use of heterogeneous links for prediction.

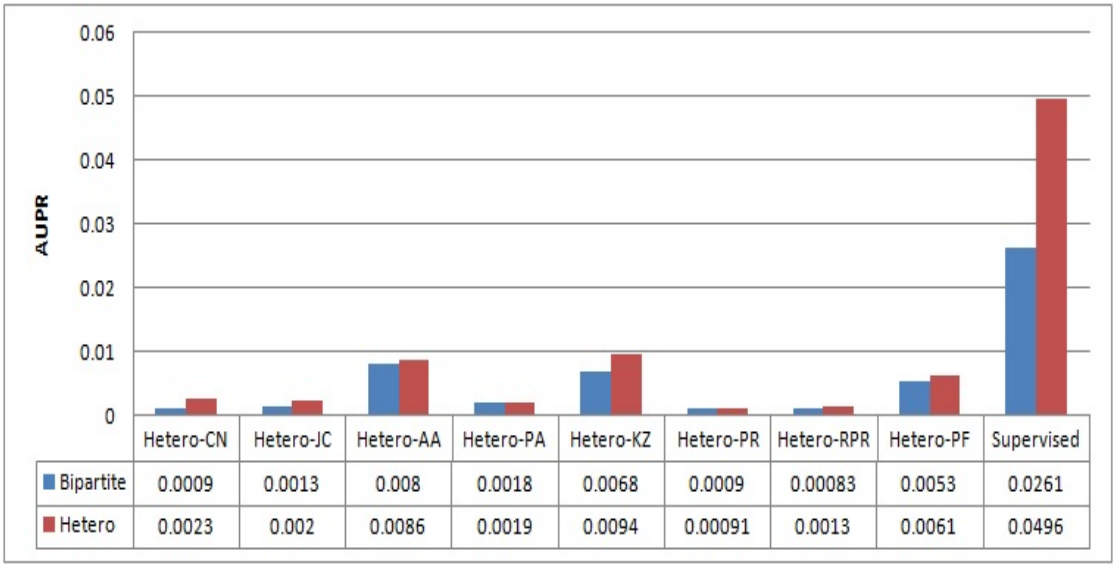


	CN	JC	AA	PA	KZ	PR	RPR	PF	Supervised
Homogeneous	0.6504	0.5942	0.6777	0.7415	0.6436	0.5734	0.579	0.6264	0.7521
Hetrogeneous	0.6811	0.6012	0.7011	0.7423	0.6843	0.5804	0.5985	0.6532	0.7724

Figure 3.11.: AUROC of **homogeneous** link prediction for **DBLP** network. The percentage of improvement is not that visible in AUROC compared to AUPR. *PA* has better prediction performance against others.

3.5.3. Prediction of Heterogeneous links

We generate the author-conference pairs and predict future links by applying the proposed measures. The results are given in Fig.3.12, 3.13, 3.14 and 3.15.



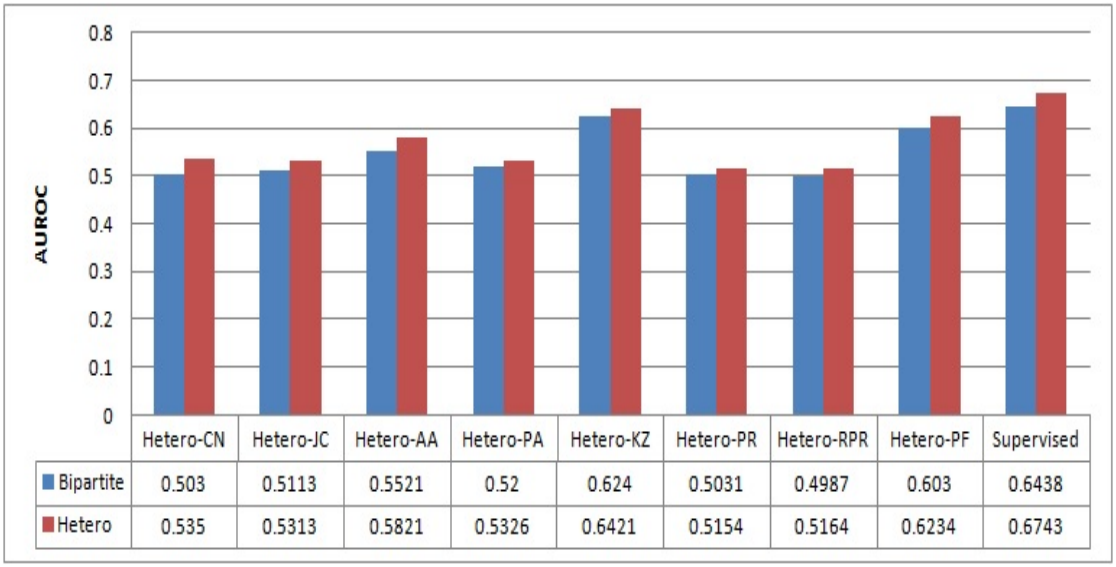
	Hetero-CN	Hetero-JC	Hetero-AA	Hetero-PA	Hetero-KZ	Hetero-PR	Hetero-RPR	Hetero-PF	Supervised
Bipartite	0.0009	0.0013	0.008	0.0018	0.0068	0.0009	0.00083	0.0053	0.0261
Hetero	0.0023	0.002	0.0086	0.0019	0.0094	0.00091	0.0013	0.0061	0.0496

Figure 3.12.: AUPR of **heterogeneous** link prediction for **HiePh-collab** network. *Hetero-AA* has performed with better accuracy over others in bipartite and *Hetero-KZ* has shown good performance in heterogeneous environment.

In predicting *author-conference* heterogeneous links in the bipartite network of HiePh-collab network, *Hetero-KZ* performs best among all the measures as seen in its AUPR score which has improved from 0.0080 to 0.0094 when co-author and co-conference homogeneous links are also used in computation. In terms of AUROC also, performance of *Hetero-KZ* is more in bipartite network and has shown an improvement of 2% in heterogeneous environment.

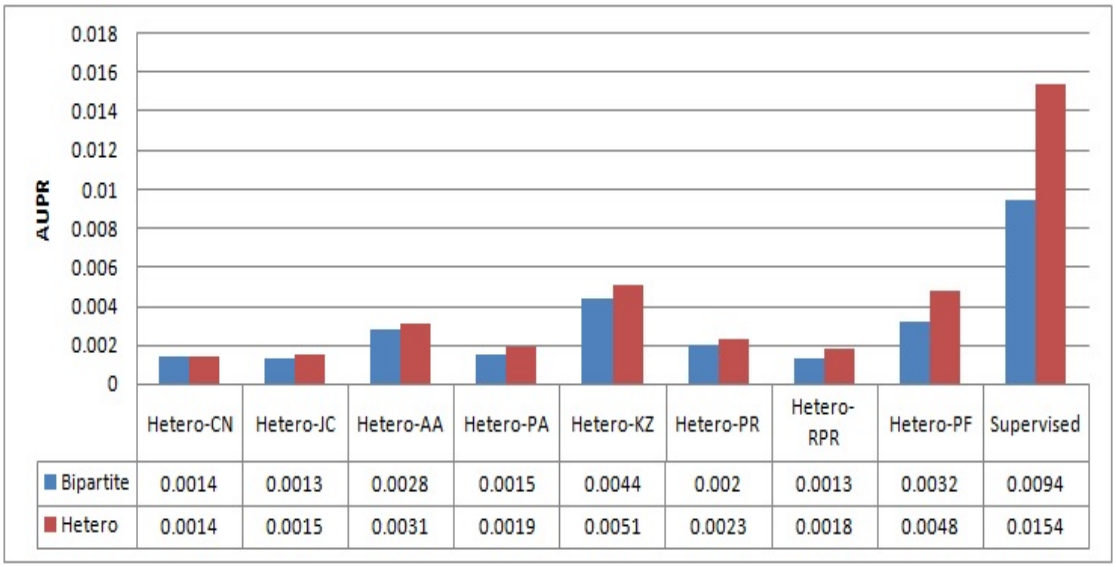
In the case of DBLP, *Hetero-AA* and *Hetero-KZ* performed almost equal with an AUROC score of 60%. An improvement of 4% is observed, when heterogeneous information is utilized. AUPR score is also improved from 0.0044 to 0.0051 for *Hetero-KZ*.

In supervised framework, the prediction score is improved from 69% to 71% for DBLP and from 64% to 67% for HiePh-collab. The improvement in AUPR score is almost double for HiePh-collab (from 0.0261 to 0.0496) as compared to DBLP (from 0.0094 to 0.0154).



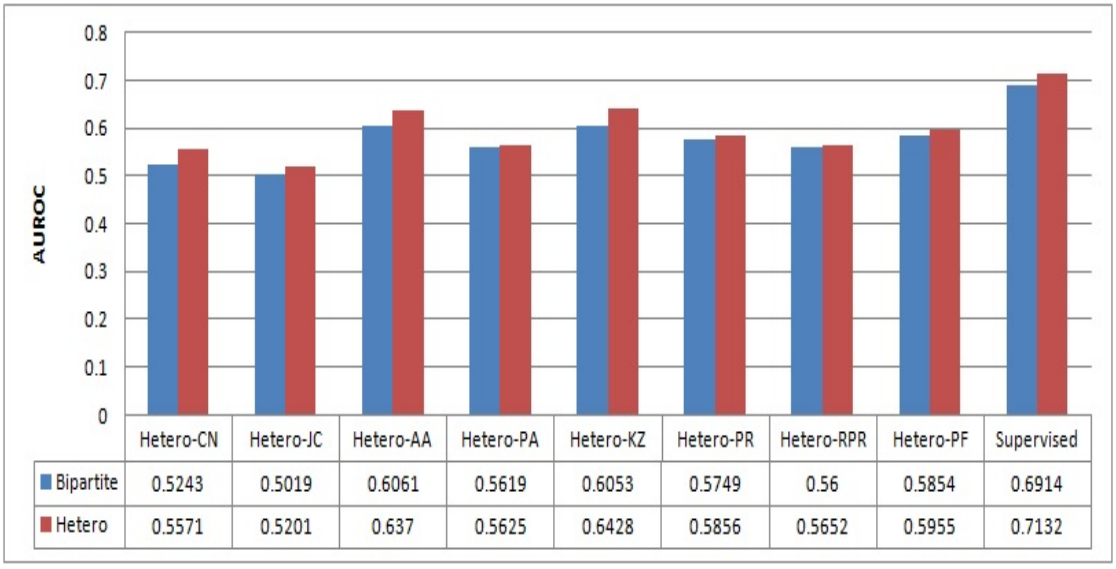
	Hetero-CN	Hetero-JC	Hetero-AA	Hetero-PA	Hetero-KZ	Hetero-PR	Hetero-RPR	Hetero-PF	Supervised
Bipartite	0.503	0.5113	0.5521	0.52	0.624	0.5031	0.4987	0.603	0.6438
Hetero	0.535	0.5313	0.5821	0.5326	0.6421	0.5154	0.5164	0.6234	0.6743

Figure 3.13.: AUROC of **heterogeneous** link prediction for **HiePh-collab** network. *Hetero-KZ* has performed better over other measures in bipartite as well as heterogeneous environment.



	Hetero-CN	Hetero-JC	Hetero-AA	Hetero-PA	Hetero-KZ	Hetero-PR	Hetero-RPR	Hetero-PF	Supervised
Bipartite	0.0014	0.0013	0.0028	0.0015	0.0044	0.002	0.0013	0.0032	0.0094
Hetero	0.0014	0.0015	0.0031	0.0019	0.0051	0.0023	0.0018	0.0048	0.0154

Figure 3.14.: AUPR of **heterogeneous** link prediction for **DBLP** network. *Hetero-KZ* predicted the links better over other measures in bipartite environment, but *Hetero-PF* has equal performance with *Hetero-KZ* in heterogeneous environment.



	Hetero-CN	Hetero-JC	Hetero-AA	Hetero-PA	Hetero-KZ	Hetero-PR	Hetero-RPR	Hetero-PF	Supervised
Bipartite	0.5243	0.5019	0.6061	0.5619	0.6053	0.5749	0.56	0.5854	0.6914
Hetero	0.5571	0.5201	0.637	0.5625	0.6428	0.5856	0.5652	0.5955	0.7132

Figure 3.15.: AUROC of **heterogeneous** link prediction for **DBLP** network. *Hetero-AA* and *Hetero-KZ* have shown equal performance in bipartite as well as heterogeneous environments.

3.6. Conclusion

Heterogeneous social networks are ubiquitous in nature and contain a lot of hidden information. In this chapter, some of the link prediction measures have been extended to heterogeneous environment. The experiments on two coauthorship networks HiePh-collab and DBLP prove that the heterogeneous networks taken as a whole instead of considering homogeneous projections improve prediction accuracy.

In the next chapter, the social networks are modelled as Probabilistic Graphical Models (PGM) and the problem of link prediction problem using PGM's is investigated.

4. Probabilistic Graphical Model Framework

This chapter describes link prediction using Probabilistic Graphical Models (PGM). The initial sections describe the PGM framework and the notation. Then the significance of PGM's for link prediction task is highlighted. In subsequent sections, a probabilistic measure based on PGM called Co-occurrence probabilistic measure (COP) [31] for homogeneous networks is presented. Main contribution in this chapter is extension of COP measure to bipartite and heterogeneous social networks. The implementation and evaluation of the proposed measure *Hetero*-COP is carried out on two benchmark coauthorship networks of DBLP and HiePh-collab.

4.1. Probabilistic Graphical Model

In probabilistic graphical model (PGM), every node is treated as a random variable and an edge between two nodes represents the dependency between the corresponding random variables.

Let $G = (V, E)$ be a graph representing a homogeneous social network. Define X_u, a random variable corresponding to every node $u \in V$ and X is set of all random variables corresponding to nodes in G. If G is directed, this graphical model is called Bayesian Network and if it is undirected, it is called Markov Random Field (MRF). Figure 4.1 gives an example of Bayesian Network and Figure 4.2 depicts a Markov Random Field along with their associated potential tables.

Since our current focus is on undirected social networks, we consider Markov Random Fields that can be induced very naturally on social networks. We list some of the standard properties of MRFs taken from [66] below.

If there is no edge between nodes u and$v$, then the random variables X_u and X_v are conditionally independent given all other nodes in the graph. This property is called as pair-wise Markov Property. This can be expressed as follows :

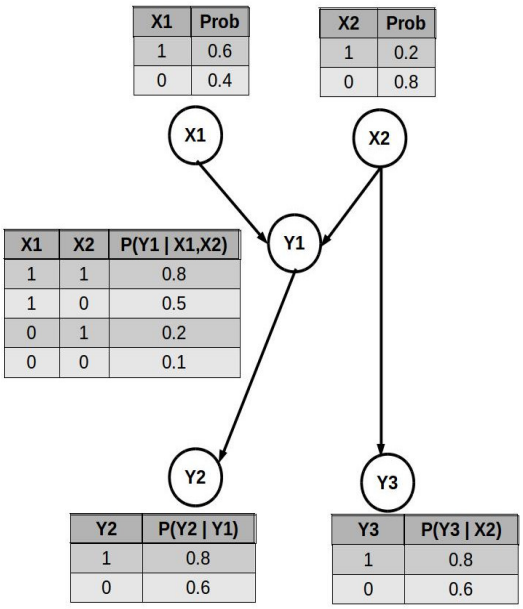


Figure 4.1.: Bayesian Network with 5 nodes and associated potential tables

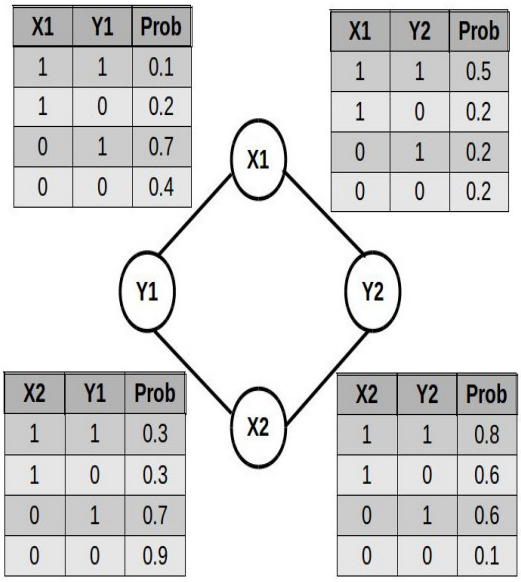


Figure 4.2.: Markov Random Field with 4 nodes and associated potential tables

$$X_u \perp\!\!\!\perp X_v, \quad if \ \ (u,v) \notin E \tag{4.1}$$

If two random variables are connected by an edge, then they have the correlation between them irrespective of all other variables [66]. This means, a random variable is conditionally independent of all the other variables given its neighbours. This property is called as local Markov property and can be expressed as follows:

$$X_u \perp\!\!\!\perp X_{V-\Gamma(v)} \mid \Gamma(v) \tag{4.2}$$

where $\Gamma(v)$ is set of neighbours of v.

Global Markovian property extends the notion of single random variable to sets of random variables. This property states that two sets of random variables $A \subseteq V$ and $B \subseteq V$ are conditionally independent, given their separator set $S = A \cap B$.

$$X_A \perp\!\!\!\perp X_B \mid X_S \tag{4.3}$$

Whenever a graph exhibits Markovian properties, it can be factorized over cliques of G. i.e, Let X_c be random variable corresponding to clique C. The clique potential table (potential function) is computed for each clique C, denoted as $\phi_c(X_C)$. Then the joint probability of x, an assignment of values to random variables X in the MRF is given by

$$P(x) = \frac{1}{Z} \prod_{c \in C(G)} \phi_c(X_C)|_x$$

where $\phi_c(X_C)|_x$ is evaluation of $\phi_c(X_C)$ at x and $C(G)$ is set of all maximal cliques of G and Z is normalizing factor defined by

$$Z = \sum_x \prod_{c \in C(G)} \phi_c(X_C)|_x$$

where x is set all possible assignments of X. Z is popularly known as partition function, which is the sum of the product of potential functions of cliques in C(G) over all possible assignments.

Markov Random Fields can be naturally induced in social networks. The next section gives the advantages of inducing PGM over social networks.

4.2. Significance of Probabilistic Graphical Model to Link Prediction

Social network contains entities as nodes and the interaction between the entities as edges. The graphical model represents the structure of social network in a natural way by considering the nodes of social network as random variables and edges as dependencies between them. By representing a social network as a PGM, the problem of link prediction, which is calculation of the probability of link formation between two nodes x, y is translated to computing the joint probability of the random variables X, Y.

The social networks are large. Therefore, finding the joint probabilities of link formation is intractable. But the links in the social networks are also sparse, with nodes generally directly connected to a only a few other nodes. This property allows the PGM distribution to be represented tractably. The model of this framework is simple to understand. Inference between two random variables is same as finding the joint probability between those two random variables in PGM. Many algorithms are available for computation of joint probability between variables, given evidence on others. These inference algorithms work directly on the graph structure and are generally faster than computing the joint distribution explicitly. With all these advantages, link prediction in PGM is more effective.

In most of the cases, the pair-wise interactions of entities are available in the event logs. For example, a research publication by three authors say u, v and w is available in the corresponding publication database. In order to model the unknown distribution of co-occurrence probabilities, events available in the event logs can be used as constraints to build a model for the unknown distribution. Probabilistic Graphical Models efficiently utilize this higher order topological information and thus are efficient in link prediction task [31].

The probabilistic model helps in estimating the joint probability distribution over the variables of the model [67]. That means, a probabilistic model represents the probability distribution over all possible deterministic states of the model. This facilitates the inference of marginal probabilities of the individual variables of the model.

4.3. Related Literature

Wang et al. [31] are among first researchers who modelled the problem of link prediction using MRFs. Kashima et al. [33] propose a probabilistic model of network evolution and apply it for predicting future links. The authors show that by intelligently selecting the parameters in an evolution model, the problem of network evolution reduces to the problem of link prediction. Clauset et al. [34] propose a probabilistic model based on hierarchical structure of the network. In hierarchical structure, the vertices are divided into groups that further subdivide into groups of groups and so on. The model infers hierarchical structure from network data and can be used for prediction of missing links. The learning task uses the observed network data and infers the most likely hierarchical structure through statistical inference.

The works of Kashima [33] and Clauset [34] are global models and are not scalable for large networks. The method of Wang et. al. [31] uses local probabilistic information of graphs for link prediction and we adopt their method in our work. The following section explains the algorithm for link prediction using MRF proposed by Wang et al. [31]

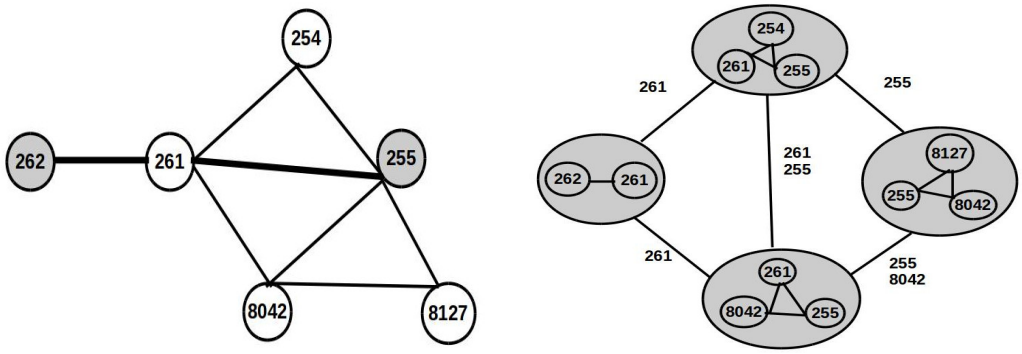
4.4. Link Prediction Using MRF

Wang et al. [31] propose a measure called Co-occurrence Probability (COP) to be computed between a pair of nodes with out an edge between them. The procedure for computing COP is explained below.

4.4.1. Co-occurrence Probability Measure

To make the probabilistic graphical model tractable, the authors propose a local model that is based on a neighbourhood set called Central neighbourhood set (CNS). This approach is based on the hypothesis that far away nodes may not influence link formation between two nodes. The CNS is constructed by choosing nodes that frequently occur on short paths between a start node and a target node.

After constructing the CNS, a Probabilistic Graph Model is constructed with the nodes in the CNS. Wang et al. consider Markov Random Field (MRF) as the PGM. The main hurdle in construction of MRF is computing cliques, as clique computation is NP-Hard. For example, in coauthorship networks, extracting cliques of the form (a,b,c) where (a,b),(b,c),(c,a) correspond-



(a) (b)

Figure 4.3.: (a) A snapshot from DBLP co-authorship network. Nodes represent authors and edges represent co-authorship relation between two authors. A clique corresponds to a set of authors publishing a paper together. (b) The corresponding clique graph.

ing to three different publications is not tractable since it amounts to subset selection problem. On the other hand, in most of the real world datasets, a specified type of cliques are readily available. For example, in co-authorship networks, a clique corresponds to a set of authors publishing a paper. A snap shot of DBLP co-authorship network is shown in Fig.4.3. Similarly, in transactional datasets, a clique may correspond to a set of items bought together by a customer. In the same way, a clique in the movie dataset is the set of actors and technicians who worked for a movie.

The authors utilize the advantage of Non Derivable Frequent Itemsets (NDI) [68] [69], which produce itemsets whose support count cannot be derived from the other itemsets in the set. Maximal itemsets can be extracted by ignoring any itemset, which is a subset of the other. The authors use these maximal itemsets extracted from NDIs for learning MRF.

Once the MRF is constructed, the Co-occurrence Probability of a pair of nodes x and y is inferred using junction tree inference algorithm [70]. Though the junction tree performs exact inference, and may not be tractable for large networks, the authors of [31] make it tractable by restricting the MRF to NDI containing only the nodes of CNS.

The computation of COP is summarized in the flow diagram given in Figure 4.4.

The original paper that proposes COP [31] does not carry the details, which are given here. The details of the underlying computations of MRF and inference using Junction tree algorithm are given elaborately in Appendix. A.

Wang et. al. [31] proposed the measure COP for homogeneous networks. We extend this

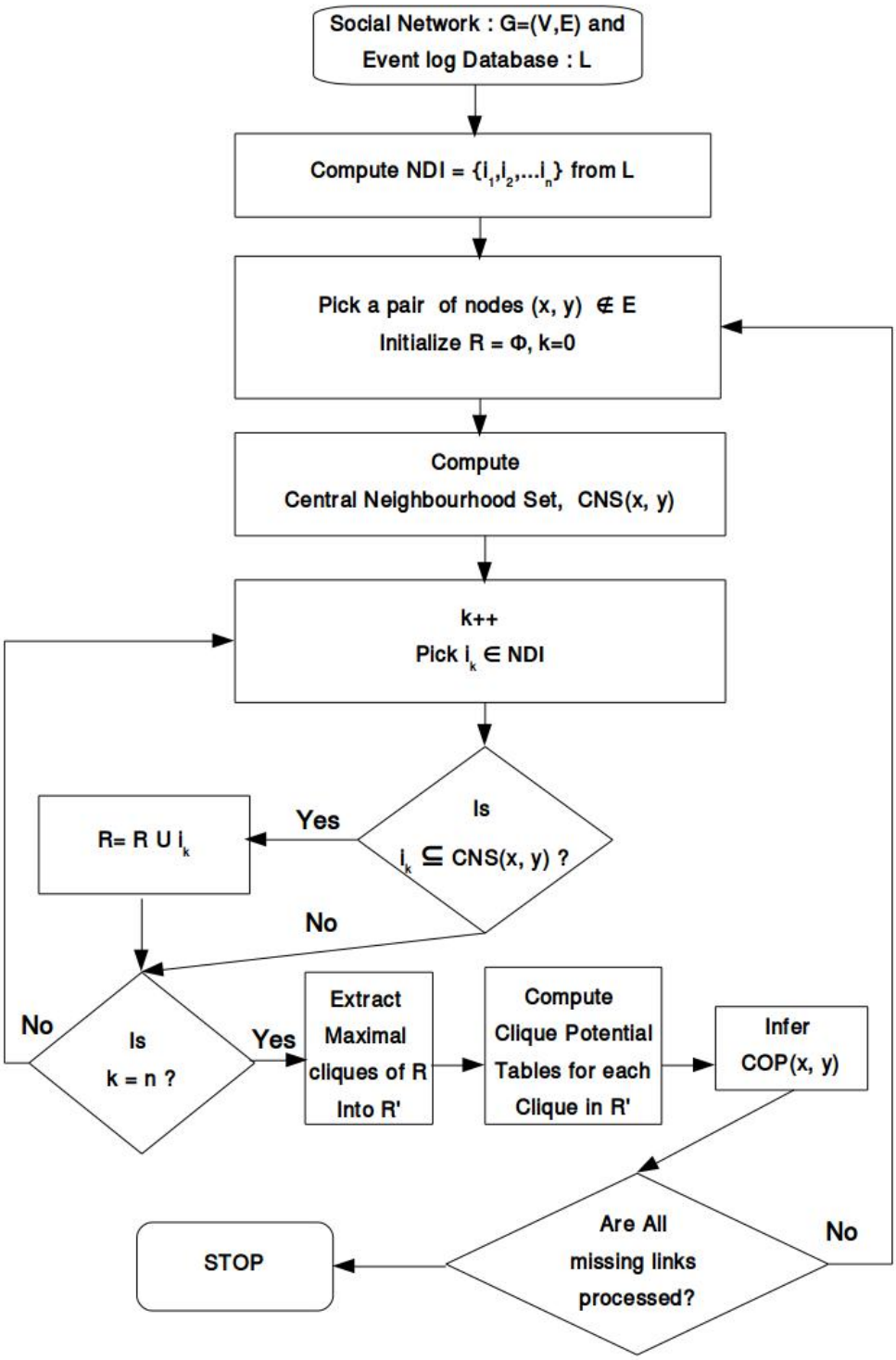


Figure 4.4.: Computation of COP

measure to bipartite and heterogeneous networks, which is the main contribution of this chapter.

4.4.2. Time complexity analysis

The time complexity analysis of the algorithm given in Fig. 4.4 is presented below:

- Computing NDI: An itemset is derivable if its support can be determined from the support of its subsets. Derivable itemsets represent redundant information and can be pruned from the set of frequent itemsets. For low values of support count, the number of derivable frequent itemsets will be very large, irrespective of the algorithm used. Calder et al. [68] propose a condensed representation of frequent itemsets, called non-derivable itemsets. Non-derivable itemsets eliminate redundant frequent itemsets significantly, using deduction rules. For low minimum support, the number of non-derivable itemsets is less compared to derivable itemsets [69]. These itemsets form cliques in the graph of collaboration network. We adopt the efficient implementation of computing non-derivable itemsets, using depth first search as presented in [68].
- Computing CNS(x,y) involves breadth first search of node y starting from node x. So, the complexity is $O(|V|+|E|)$
- Extract the cliques in NDI: This step needs scanning each clique in NDI and see whether all nodes in clique are from CNS or not. Therefore, the complexity will be $O(|NDI|)$
- Compute MRF: Construction of MRF involves scanning each clique in NDI and computing the total probability of each assignment of the nodes in the clique. Therefore, the complexity is $O(|NDI|)$ where NDI is the set of non derivable itemsets.
- Junction tree inference : The time complexity scales by the width of the junction tree.

4.5. Proposed Measure : $Hetero$-COP

4.5.1. Preliminaries

In homogeneous environment, the prediction score between a pair of nodes which are not directly connected in a network can be computed by first identifying the cliques connecting the nodes in the central neighbourhood set; constructing a PGM induced by the nodes in the CNS involves computing maximal cliques and inferring the probability between the nodes using junction tree

algorithm. The identification of cliques is central to the algorithm. This task is not trivial even for homogeneous networks.

Recall that from 3.1 and 3.2 that a heterogeneous network contains multiple types of nodes and multiple types of edges, a bipartite network contains only two types of nodes and heterogeneous edges. The definition of heterogeneous network and bipartite network are given below for ready reference.

Definition 4.1. A ***Heterogeneous Social Network*** is defined as $G = (V, E)$, where $V = V_1 \cup V_2 \cup \ldots \cup V_n$ represents n types of nodes, $E = E_1 \cup E_2 \cup \ldots E_m$ denotes m types of edges.

Definition 4.2. Bipartite network is defined as $G = (V_1 \cup V_2, \quad E)$, where V_1 and V_2 are sets of two types of nodes, E represents the set of edges between nodes of type V_1 and V_2.

An edge between same type of nodes is called as **homogeneous edge** and an edge between different types of nodes is called as **heterogeneous edge**.

4.5.2. Computation of *Hetero*-COP

The proposed algorithm for construction of *Hetero*-COP of a missing link (x, y) in the lines of *COP* is given in Algorithm 1. The modification of the computation procedure for heterogeneous networks is given in subsequent sections.

Algorithm 1 *Hetero*-COP measure for Link Prediction in heterogeneous networks

Input: $G = (V, E)$, where
$V = \{V_1 \cup V_2 \cup \ldots V_n\}$ is the set of n types of nodes,
$E = \{E_1 \cup E_2 \cup \ldots E_m\}$ is set of m types of links.
Output: *Hetero*-COP(x, y) where (x, y) is a missing link.

Step 1: Extract H-cliques from G, from the event logs using Algorithm 2. Call it *HCliq*.
Step 2: Compute **HCNS**, the central neighbourhood set of (x, y) using the Algorithm 3. As G is heterogeneous graph, HCNS is computed using **meta-paths** of G.
Step 3: Extract H-cliques formed with the nodes in HCNS and compute clique potentials. This forms the local MRF with the nodes in HCNS.
Step 4: Return *Hetero*-COP(x, y) which is the joint probability of link (x, y) using junction tree algorithm.

A clique in heterogeneous networks consists of multiple types of nodes fully connected by homogeneous or heterogeneous edges. We term this type of clique as **H-clique**. H-cliques are complex in nature. For example in Fig. 4.5, there are three types of nodes, three types of homogeneous links and three types of heterogeneous links.

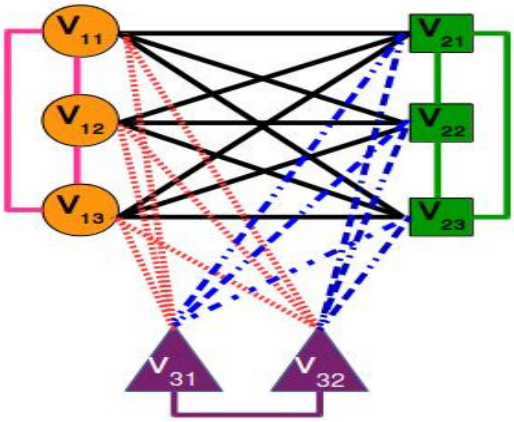


Figure 4.5.: H-clique with three types of nodes, three types of homogeneous links and three types of heterogeneous links.

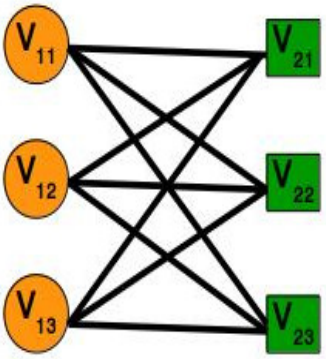


Figure 4.6.: B-clique extracted from Fig. 4.5 by choosing two types of nodes and suppressing homogeneous links.

In bipartite networks, the notion of H-clique can be considered as complete bipartite subgraph consisting only of heterogeneous edges. We call this clique as **B-clique**. B-cliques are extracted from H-cliques having only two types of nodes by suppressing homogeneous links. A sample B-clique extracted from H-clique in Fig. 4.5 is given in Fig.4.6.

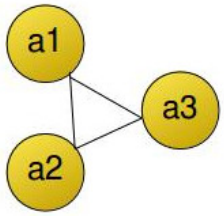
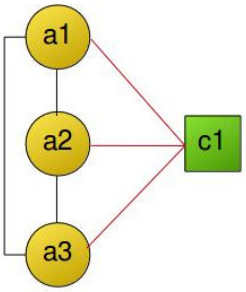
Extracting all types of H-cliques in a given heterogeneous network is not tractable. A tractable approach used in this work for extraction of H-cliques is given below.

4.5.2.1. Extraction of H-cliques

To make the procedure of extraction of H-cliques tractable, we restrict the computation to constructing H-cliques using homogeneous cliques. As there may be multiple types of links existing between two nodes, we limit the H-clique definition to the existence of any one type of link between the nodes in the clique.

In most of the cases, the homogeneous cliques as well as heterogeneous edges are available in the event logs. For example, in coauthorship networks, the group of authors who publish a paper together forms a homogeneous clique of author nodes and an author who publishes a paper in a conference forms a heterogeneous edge between the author node and conference node. Table.4.1 shows the types of cliques available in the publication event logs.

Table 4.1.: Types of cliques available in publication event logs

Event	Clique available in event log	Clique
a_1,a_2,a_3 publish a paper together	Homogeneous clique <a_1, a_2, a_3>	
a_1 publish a paper in conference c_1	Heterogeneous Edge <a_1, c_1>	
a_1,a_2,a_3 publish a paper together in conference c_1	Heterogeneous clique <a_1, a_2, a_3, c_1>	

We first extract homogeneous cliques of each type using NDI algorithm [69]. For each pair of homogeneous cliques C_1 and C_2 of different types, if every pair of nodes $u \in C_1$ and $v \in C_2$ is connected by a heterogeneous edge, then it is clearly a H-clique.

Algorithm 2 describes the extraction of H-cliques from a given heterogeneous network. After obtaining H-cliques using Algorithm 2, the event database is updated by adding the set of H-cliques to NDI. This set is given as input to the COP algorithm described in Fig.4.4 in order to construct MRF.

The step of constructing CNS considers **meta-paths** instead of homogeneous paths. Therefore, we term this as Heterogeneous Central Neighbourhood (HCNS). Then we extract H-cliques containing only the nodes in HCNS. Once the H-cliques in HCNS are extracted, the MRF can be constructed same way as in the homogeneous case.

Algorithm 2 Extraction of **H-Cliques** from a Heterogeneous network

Input:
$G = (V, E)$ where $V = V_1 \cup V_2 \cup \ldots \cup V_n$ with n types of nodes and $E = E_1 \cup E_2 \cup \ldots E_m$ with m types of edges.
$Mcliq$: Maximal homogeneous cliques derived from NDI of each type of nodes.
check_pair-wise_connections(C_1, C_2) : Sub procedure which returns true if all pair wise links between two cliques C_1 and C_2 exist.
Output: $Hcliq$, set of maximal H-cliques of G.

$Hcliq = \phi$
for $cliq_i \in Mcliq$ **do**
  for $cliq_j \in Mcliq$ **do**
    if $\exists$ a heterogeneous edge between nodes of $cliq_i$ and $cliq_j$ **then**
      can_form_cliq = $check\_pair\text{-}wise\_connections(cliq_i, cliq_j)$
      if can_form_cliq **then**
        $new\_hcliq = cliq_i \cup cliq_j$
        $Hcliq$ = $Hcliq \cup new\_hcliq$
      end if
    end if
  end if
end for

4.5.2.2. Computation of Heterogeneous Central Neighbourhood Set

HCNS(x, y) is computed using a breadth first search based algorithm on heterogeneous network as follows: All meta-paths between x and y are obtained using breadth first search (BFS) algorithm. Then, the frequency score of each meta-path is computed by summing the occurrence count of nodes on the path. Occurrence count of a node is the number of times the node appears in all meta-paths. The meta-paths are now ordered in the increasing order of length with equal paths have ordered in decreasing order of frequency score. The size of the central neighbourhood set is further restricted by considering only top k nodes. The procedure of computing $HCNS(x, y)$ is described in the Algorithm 3.

Algorithm 3 Hetero Central Neighborhood Set$(G,x,y,l,maxSize)$

Input: G: a graph; x: starting node; y: ending node; l:maximum path length; *maxSize*:Central Neighbourhood Set size threshold

Output: *HCNS*, Heterogeneous Central Neighbourhood Set between x and y;

Step 1: Compute meta-paths consisting of homogeneous and heterogeneous edges of length $\leq l$ between x and y.

Step 2: Find occurrence count O_k of each node k in paths between x and y.

Step 3: Compute *frequency-score* F_p, of each meta-path p as follows:

$$F_p = \sum_{k \in p} O_k$$

frequency-score of a path is the sum of the occurrence counts of all nodes along the paths.

Step 4: Sort the paths in increasing order of path length and then in decreasing order of *frequency-score*. Let the ranked list of meta-paths be P.

Step 5:

while size(HCNS) $\leq$ *maxSize* **do**

Add nodes of path $p \in P$ to HCNS

end while

return HCNS

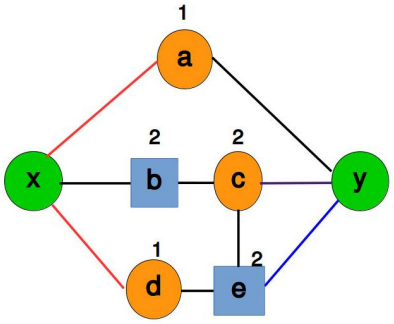


Figure 4.7.: A toy example for illustrating computation of HCNS between nodes x and y. Node weights are the occurrence counts.

meta-path(p)	**frequency-score**(F_p)
x-a-y	$O_a = 1$
x-b-c-y	$O_b + O_c = 4$
x-d-e-y	$O_d + O_e = 3$
x-b-c-e-y	$O_b + O_c + O_e = 6$

Table 4.2.: All paths between nodes x and y in Fig.4.7 and their frequency scores.

The procedure of computation of HCNS on a toy example in Fig.4.7 is shown in Table 4.2. Table 4.2 shows all the meta-paths between the nodes x and y of the graph shown in Fig.4.7 along with their *frequency-scores*. For the heterogeneous graph in Fig.4.7, CNS$(x,y) = \{x,a,b,c,y\}$, if *maxSize* is taken as 5.

4.5.2.3. Construction of local MRF

After computing the HCNS, the H-cliques containing only nodes of HCNS are extracted. This forms the clique graph of local MRF. MRF construction needs computation of clique potentials. The clique potential table of a H-clique is computed using the NDI.

A snapshot of a H-clique extracted with its associated potential table is shown in Fig.4.8 and Table.4.3.

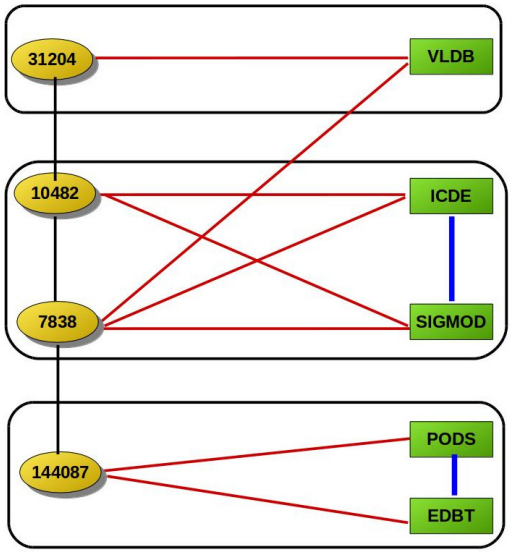


Figure 4.8.: A snapshot of DBLP Heterogeneous Network

10482	7838	ICDE	SIGMOD	$\phi_C(F)$
0	0	0	0	0.00000
0	0	0	1	0.00000
...	...	...	...	...
0	1	0	1	0.00000
...	...	...	...	...
0	1	1	0	0.23000
...	...	...	...	...
1	1	0	1	0.23000
1	1	1	0	0.03128
...	...	...	...	...
1	1	1	1	0.03128

Table 4.3.: A partial Clique Potential Table $\phi_C(F)$ of H-clique $C = \{10482, 7838, ICDE, SIGMOD\}$

4.5.2.4. Computation of *Hetero*-COP score

Once the local MRF of a pair of nodes x and y is constructed, the *Hetero*-COP score between the nodes is obtained using junction tree inference algorithm. Note that *Hetero*-COP score for a link x - y cannot be computed if x and y are in disjoint cliques as there exists no path connecting these cliques.

4.6. Implementation and Results

The proposed measure Hetero-COP is evaluated on two coauthorship benchmark datasets DBLP and HiePh-collab for link prediction. The details of these datasets are given in 2.6.

The performance of *Hetero*-COP is compared with four **neighbourhood-based** measures CommonNeighbours CN(x,y), JaccardCoefficient JC(x,y), AdamicAdar AA(x,y), PreferentialAttachment(PA), a **path-based** measure Katz(x,y) with damping factor 0.05 and within path lengths 5, three **random-walk based** measures PageRank PR(x,y), RootedPageRank RPR(x,y) with restart parameter as 0.15, PropFlow PF(x,y) with path length threshold 5, and the **probabilistic** measure COP.

Note that the bipartite and heterogeneous versions of these scores except COP have been discussed in chapter 3.

The link prediction scores of CN, JC, AA, PA, KZ, PR, RPR, PF are computed using the software tools LPmade [60]. The software libDAI [71] is used or inference using junction tree algorithm to compute *Hetero*-COP score. In computation of Heterogeneous Central Neighbourhood Set (HCNS), the threshold for meta-path length is taken as 5 and size of HCNS is considered as 6.

4.6.1. Results

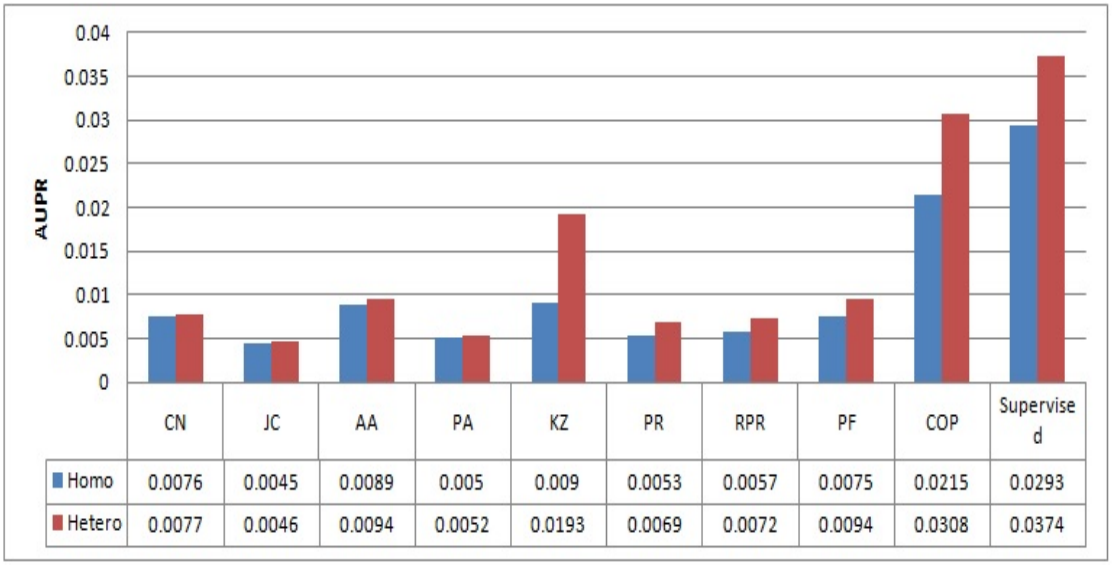
We predict two types of links specified below in the coauthorship networks of DBLP and HiePh-collab same as in Chapter 3.

- Prediction of homogeneous links (author-author) in homogeneous and heterogeneous networks.
- Prediction of heterogeneous links (author-conference) in bipartite as well as heterogeneous environments. For considering the bipartite network, we suppress the homogeneous links of *author-author* and $conf$-$conf$.

4.6.1.1. Prediction of Homogeneous links

The prediction results of coauthor homogeneous links in homogeneous networks of DBLP and HiePh-collab are given in Fig.4.9, 4.10, 4.11 and 4.12.

A slight improvement of 1%-2% is observed when the heterogeneous linkages are used for prediction in comparison to using only homogeneous links. *Hetero*-COP shows improved performance over all the other link prediction measures for both DBLP and HiePh-collab networks.

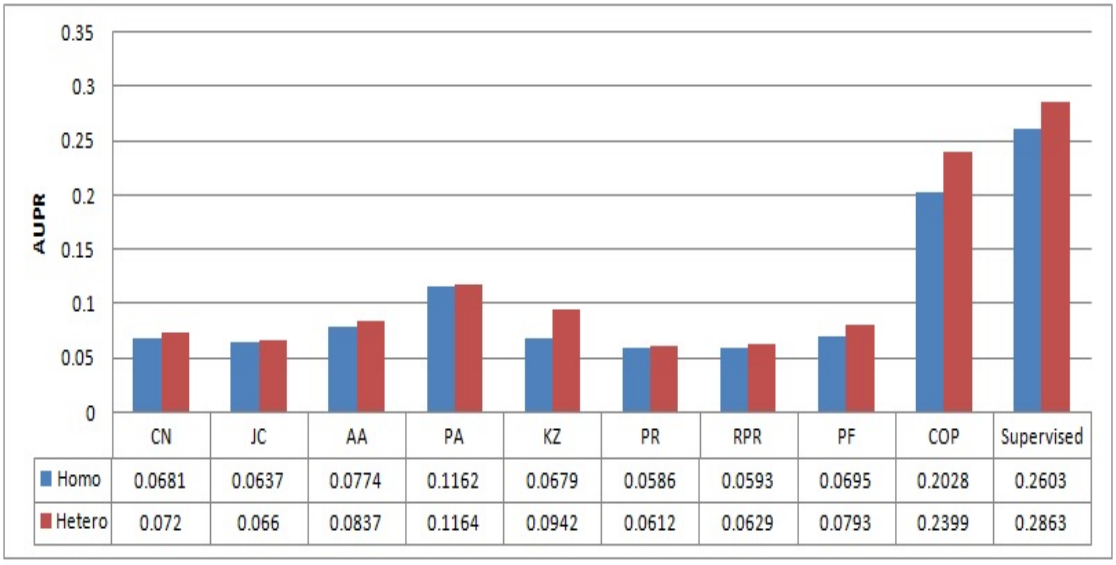


	CN	JC	AA	PA	KZ	PR	RPR	PF	COP	Supervised
Homo	0.0076	0.0045	0.0089	0.005	0.009	0.0053	0.0057	0.0075	0.0215	0.0293
Hetero	0.0077	0.0046	0.0094	0.0052	0.0193	0.0069	0.0072	0.0094	0.0308	0.0374

Figure 4.9.: AUPR score for *author-author* link prediction of *Hetero*-COP Vs baseline measures for **HiePh-collab** network. *COP* and *KZ* have improved performance significantly with heterogeneous information. Prediction of *Hetero*-COP is better over all other baseline measures.

	CN	JC	AA	PA	KZ	PR	RPR	PF	COP	Supervised
Homo	0.5846	0.5167	0.6064	0.5466	0.619	0.5389	0.5494	0.584	0.7153	0.7405
Hetero	0.592	0.5169	0.6253	0.55	0.6412	0.5639	0.5745	0.6043	0.739	0.7593

Figure 4.10.: AUROC score for *author-author* link prediction : *Hetero*-COP Vs baseline measures for **HiePh-collab** network. *Hetero*-COP has improved 2% over its homogeneous version and 9% over other heterogeneous measures and 1.5% over its homogeneous version.

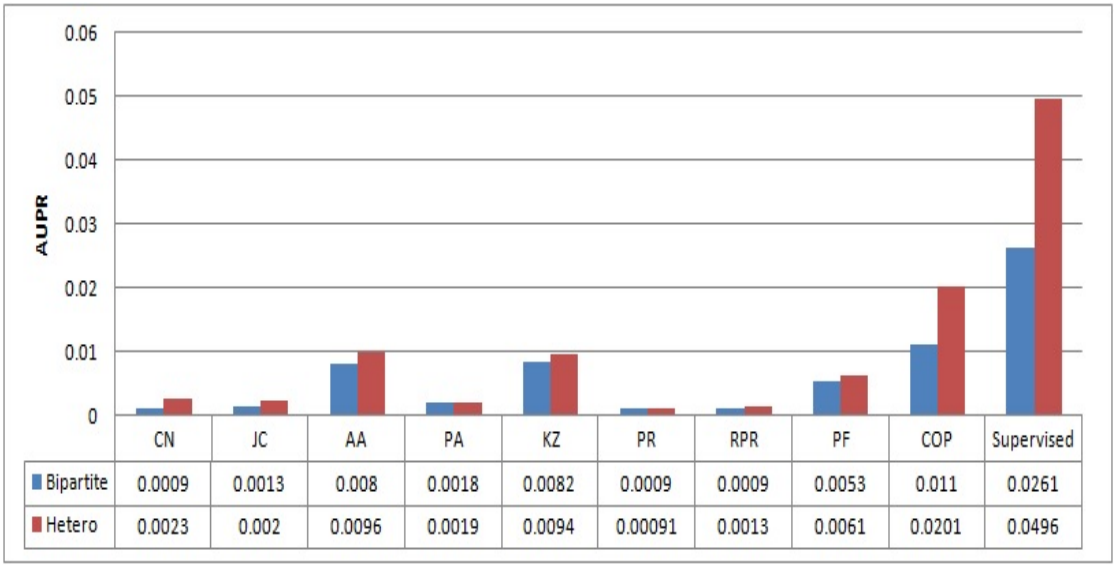


	CN	JC	AA	PA	KZ	PR	RPR	PF	COP	Supervised
Homo	0.0681	0.0637	0.0774	0.1162	0.0679	0.0586	0.0593	0.0695	0.2028	0.2603
Hetero	0.072	0.066	0.0837	0.1164	0.0942	0.0612	0.0629	0.0793	0.2399	0.2863

Figure 4.11.: AUPR of *author-author* link prediction of *Hetero*-COP Vs base-line measures for **DBLP** network. *COP* has improved predictions in both homogeneous as well as heterogeneous networks.

	CN	JC	AA	PA	KZ	PR	RPR	PF	COP	Supervised
Homo	0.6504	0.5942	0.6777	0.7415	0.6436	0.5734	0.579	0.6264	0.8379	0.85
Hetero	0.6811	0.6012	0.7011	0.7423	0.6843	0.5804	0.5985	0.6532	0.8439	0.8745

Figure 4.12.: AUROC of **homogeneous** link prediction of *Hetero*-COP Vs baseline measures for **DBLP** network. *Hetero*-COP improved 10% over other heterogeneous measures and a slight improvement over its homogeneous version.



	CN	JC	AA	PA	KZ	PR	RPR	PF	COP	Supervised
Bipartite	0.0009	0.0013	0.008	0.0018	0.0082	0.0009	0.0009	0.0053	0.011	0.0261
Hetero	0.0023	0.002	0.0096	0.0019	0.0094	0.00091	0.0013	0.0061	0.0201	0.0496

Figure 4.13.: AUPR scores of **heterogeneous** link prediction for **HiePh-collab** network. *Hetero*-COP dominated all other measures in homogeneous and heterogeneous networks. The performance of *COP* is doubled with the usage of heterogeneous links.

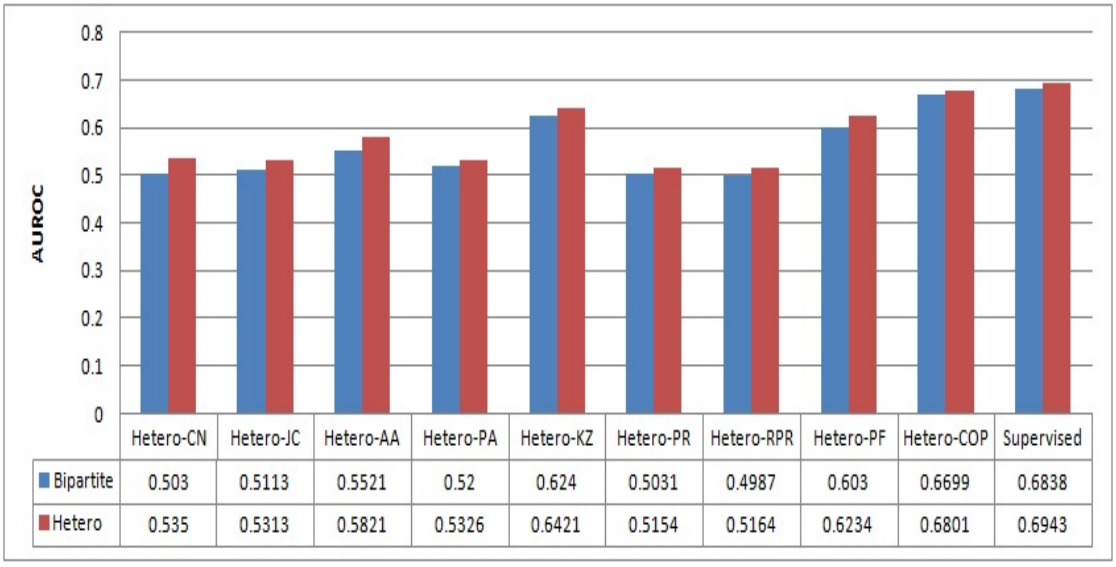
The AUPR score has improved from 0.0193 to 0.0308 for HiePh-collab network and from 0.1162 to 0.2399 for DBLP. Similarly, an average improvement of 9% in AUROC score is achieved by *Hetero*-COP over standard link prediction measures for HiePh-collab and DBLP networks.

4.6.1.2. Prediction of Heterogeneous links

We generate the author-conference pairs and predict future links by applying the proposed measures. The results are given in Fig.4.13, 4.14, 4.15 and 4.16. *Hetero*-COP performs best among all the other measures used for link prediction for both datasets.

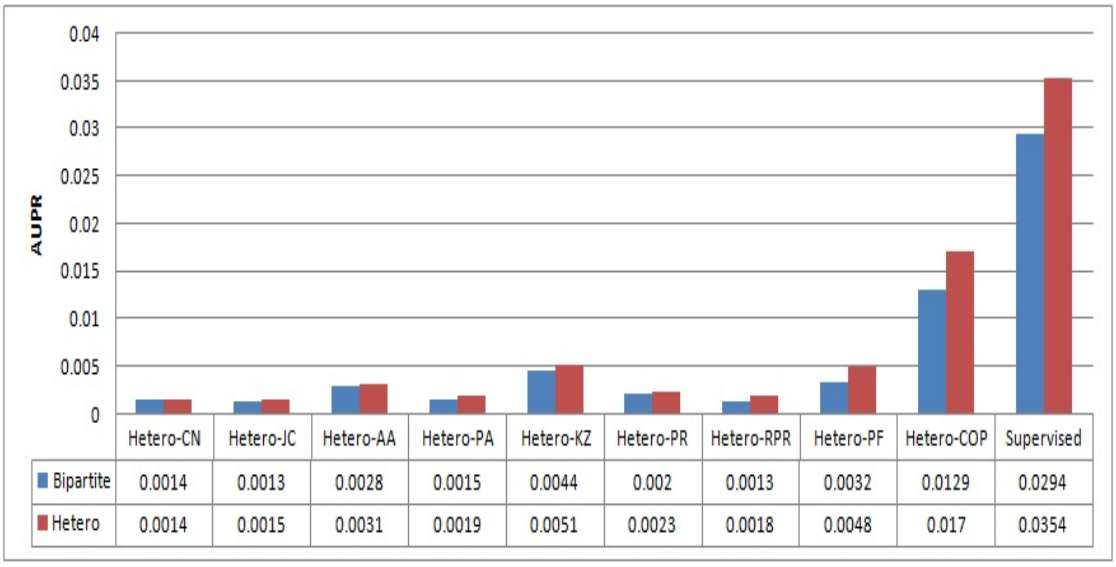
In predicting *author-conference* heterogeneous links in the bipartite network of DBLP, 8% improvement over existing link prediction measures is observed in terms of AUROC. When heterogeneous information is utilized, a further 4% improvement is observed. In case of HiePh-collab network, the improvements are 4% and 2% respectively.

AUPR score is improved from 0.0044 (best among existing measures) to 0.0129 in bipartite environment and the score is further improved to 0.0170 when co-author and co-conference homogeneous links are also used in computation. Finally, *Hetero*-COP obtains an AUPR score of 0.0170 over all other measures in predicting *author-conference* heterogeneous links.



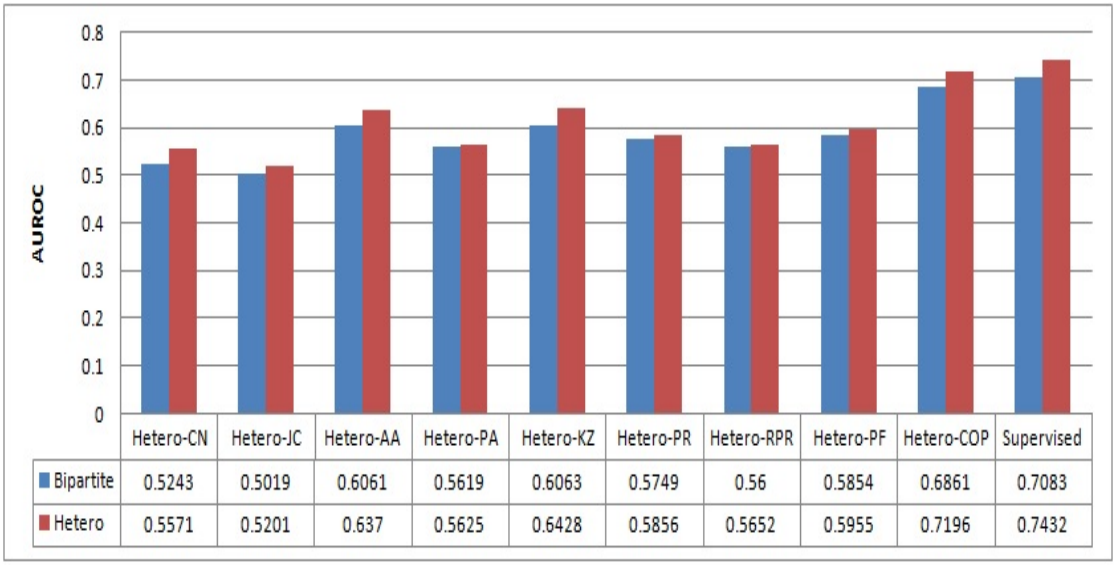
	Hetero-CN	Hetero-JC	Hetero-AA	Hetero-PA	Hetero-KZ	Hetero-PR	Hetero-RPR	Hetero-PF	Hetero-COP	Supervised
Bipartite	0.503	0.5113	0.5521	0.52	0.624	0.5031	0.4987	0.603	0.6699	0.6838
Hetero	0.535	0.5313	0.5821	0.5326	0.6421	0.5154	0.5164	0.6234	0.6801	0.6943

Figure 4.14.: AUROC of **heterogeneous** link prediction for **HiePh-collab** network. *Hetero*-COP achieves 8% improvement over *Hetero*-KZ in bipartite network and further 2% in heterogeneous network.



	Hetero-CN	Hetero-JC	Hetero-AA	Hetero-PA	Hetero-KZ	Hetero-PR	Hetero-RPR	Hetero-PF	Hetero-COP	Supervised
Bipartite	0.0014	0.0013	0.0028	0.0015	0.0044	0.002	0.0013	0.0032	0.0129	0.0294
Hetero	0.0014	0.0015	0.0031	0.0019	0.0051	0.0023	0.0018	0.0048	0.017	0.0354

Figure 4.15.: AUPR of **heterogeneous** link prediction for **DBLP** network. The improvement in *Hetero*-COP is clearly visible in both bipartite and heterogeneous networks.



	Hetero-CN	Hetero-JC	Hetero-AA	Hetero-PA	Hetero-KZ	Hetero-PR	Hetero-RPR	Hetero-PF	Hetero-COP	Supervised
Bipartite	0.5243	0.5019	0.6061	0.5619	0.6063	0.5749	0.56	0.5854	0.6861	0.7083
Hetero	0.5571	0.5201	0.637	0.5625	0.6428	0.5856	0.5652	0.5955	0.7196	0.7432

Figure 4.16.: AUROC of **heterogeneous** link prediction for **DBLP** network. *Hetero*-COP has improvement of 8% over *Hetero-KZ* and 2% in heterogeneous network.

4.6.2. Discussion

The improvement of prediction performance in heterogeneous environment over bipartite environment is obvious because in heterogeneous environment, additional information is used in the form of co-author and co-conference homogeneous links. But even in bipartite network, *Hetero*-COP gives improved performance over the other measures. Therefore, *Hetero*-COP can be applied to the natural bipartite networks.

DBLP network is sparse compared to HiePh-collab network. The increase in accuracy is more for DBLP compared to HiePh-collab network. This is due to the reason that in the case of sparse networks, global information does not contribute much. As COP uses pure local information the rate of false positives is reduced and thus the prediction accuracy is improved. Therefore, the increment in accuracy is more in the case of sparse networks compared to dense networks for link prediction in bipartite networks.

4.7. Conclusion

A probabilistic measure called *Hetero*-COP is proposed and evaluated against standard link prediction algorithms in this chapter. *Hetero*-COP gives better results in terms of AUPR and AUROC

for two types of links *author-author* and *author-conference* in the homogeneous, bipartite and heterogeneous coauthorship networks. The time information available for formation of links is not utilized in this work. In the next chapter, we utilize the information of time of formation of links and propose a new temporal measure called Temporal Co-occurrence Probability measure for link prediction.

It is possible that the co-occurrence probability feature has not been explored much in the literature since it is computationally expensive. On the other hand, it has been designed as a local model with parameters that can be chosen to keep the computation cheap. It can be seen that this is a rich feature with a great potential to improve the performance of link prediction algorithms to a significant extent.

5. Temporal Measures for Link Prediction

The existing literature does not lay emphasis on utilizing temporal information of the edges. By differentiating recent collaborations from very old interactions, one can avoid obtaining spurious results. Several temporal measures, viz. Time Score (TS) [43] have been proposed in the literature that work for homogeneous networks. The temporal measures also utilize the weight of interaction.

In this chapter, we propose a new measure called Temporal Co-occurrence Probability (TCOP) for homogeneous networks, that extends computations on Markov Random Fields by effectively utilizing the temporal information available in the network. Further, we extend the temporal measures existing in literature along with the newly proposed measure TCOP to bipartite and heterogeneous networks. The newly proposed heterogeneous measures are evaluated on four bibliographic networks of DBLP, Condmat, HiePh-collab and HiePh-cite in all environments. In the previous chapters, results have been shown on two of these datasets, DBLP and HIePH-collab, In this chapter, results for all the four datasets are evaluated.

We extend the definition of link-prediction problem [8] given by Liben Nowell et al. to include temporal edges as follows : A homogeneous social network $G(V,E,w,t)$ with vertex set V, edge set E, weight function w on E and time function $t : E \rightarrow 2^{\mathbb{N}}$, $t(u,v)$ denotes set of years of interaction of the nodes u and v is given. The aim of link prediction problem is to predict interactions that are more likely to occur in the network at a future time t'. Here, $t(u,v)$ is defined to be a discrete set and the definition could be extended to continuous intervals.

5.1. Related Literature

This section presents the link prediction literature for homogeneous networks. A temporal measure called Time-score is defined in [43] for homogeneous networks. Time-score is an extension

of Common neighbour measure. The authors utilize Time-score as an unsupervised measure to perform link prediction. The results obtained by the authors show that Time-score measure predicts the future links more accurately, compared to common neighbourhood based measures. A mechanism for dynamic link inference in heterogeneous networks using temporal information has been presented in [42]. In [32], Wang et al. constructed a time-constrained probabilistic factor graph model (TPFG), for a collaboration network and mined the hidden advisor-advisee relation between the authors. Co-author relations have been predicted using meta-path based features in [51].

Probabilistic graphical models efficiently utilize the higher order topological information and thus are efficient in link prediction task [31]. No probabilistic measure is reported in literature which uses time information. In the next section, we present a probabilistic measure called TCOP, by incorporating temporal information in the graphical model framework to perform link prediction for homogeneous networks.

5.2. Proposed Measure : Temporal Co-occurrence Probability (TCOP)

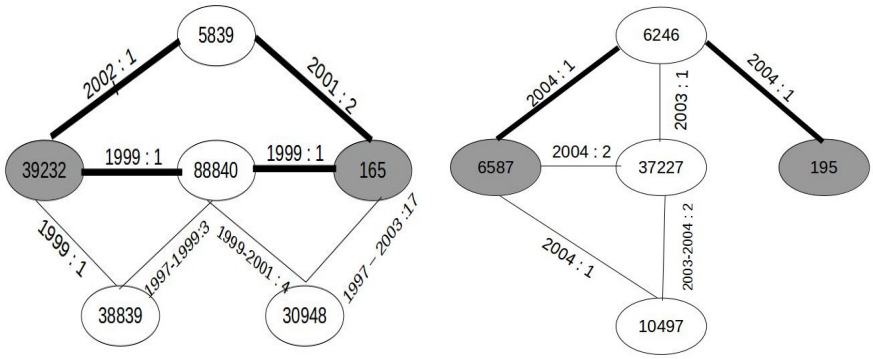
Let $G = (V,E)$ be a temporal homogeneous network. A temporal weighted homogeneous network is represented as a graph, $G = (V,E,w,t)$, where V represents nodes, E represents interaction between pairs of nodes, $w : E \rightarrow \mathbb{N}$, $w(u,v)$ corresponds to weight of interaction between nodes u and v, and $t : E \rightarrow 2^{\mathbb{N}}$, $t(u,v)$ represents the set of time instants of interaction between nodes u and v. $t_{max}(u,v)$ and $t_{min}(u,v)$ denote the most recent and the oldest time of interaction respectively.

5.2.1. Motivation for TCOP

The probabilistic measure Cooccurrence Probability(COP) proposed by Wang et.al [31] is described in 4.4.1.

COP does not take into consideration the age of the link, which may lead to spurious results, that is illustrated here. Consider two snapshots extracted from DBLP co-authorship network during 1997-2005 in Fig. 5.1. Nodes denote authors and edges are labelled as $t : w$, where t denotes the year(s) of publication and w denotes the number of publications during that period. Considering the prediction year as 2006, COP predicts a link between authors 39232 and 165 for the year

2006, whereas DBLP does not contain such a link. Note that many of the links in the cliques in Fig. 5.1(a) are very old links. Now consider graph snapshot in Fig 5.1(b). DBLP shows a link between authors 6587 and 195 in the year 2006, that is not predicted by COP. Thus, a few fresh links may contribute to the link formation rather than presence of many old links. Therefore, we extend the model of COP to TCOP (Temporal Co-occurrence Probability), by incorporating time information into the model.



(a) (b)

Figure 5.1.: A snapshot from DBLP collaboration network with temporal information

5.2.2. Computation of TCOP

The algorithm for computation of Temporal Co-occurrence Probability(TCOP) between two nodes x and y follows three steps, similar to computation of COP. The algorithm for computing TCOP is given below.

Step 1: **Compute central neighbourhood set (CNS) of** x **and** y **:** In this computation, path length (l) and size of CNS ($MaxSize$) are pre-set parameters. Consider all the nodes that lie on shortest paths of length less than l between x and y. The nodes are ordered according to the frequency of occurrence and the number of nodes included in $CNS(x,y)$ is bounded by the size of CNS.

Step 2: **Construct Markov Random Field(MRF) with the nodes in CNS**(x,y)**:** Find all the cliques C in G, containing only the nodes of CNS(x,y). The MRF construction involves computation of clique potentials. Clique potential is an assignment of probabilistic weights

to all subsets of C, called factors F of C. We define temporal weight for a factor F as follows:

$$Temporal\text{-}Weight(F) = \frac{w(F).\beta^{r(F)}}{|t_{max}(F) - t_{min}(F)| + 1} \tag{5.1}$$

where

$$w(F) = n \left(\frac{1}{\sum_{\substack{(u,v)\in F \\ w(u,v)\neq 0}} \frac{1}{w(u,v)}} \right)$$

$t_{max}(F) = \max_{(u,v)\in F}(t_{max}(u,v))$ $\quad$ $\beta < 1$ is a damping factor

$r(F) = Current\ Year$ - $t_{max}(F)$,

$t_{min}(F) = \min_{(u,v)\in F}(t_{min}(u,v))$ $\quad$ captures recency of factor F and

current year stands for the year of prediction.

$Temporal\text{-}Weight(F)$ gives more weight to recent cliques, with $w(F)$ using harmonic mean instead of raw weights, as described in Time-score of [43]. The total weight of the factor graph is damped by β, which lowers the value if the interactions are not recent. Further, this value is normalized by the total period of interaction of the authors in the factor graph. Thus, Temporal-Weight(F) yields a positive value less than 1, for any factor F.

Step 3: **Infer the joint probability of x and y:**

Use the standard junction tree inference algorithm [70], [72] to train the constructed MRF in order to infer the joint probability between nodes i and j.

The computation of TCOP measure is summarised in the Algorithm 4.

Algorithm 4 Temporal Co-occurrence Probabilistic measure for Link Prediction

Input: $G = (V, E, w, t)$, missing link $(x, y) \notin E$
Output: TCOP score for missing link (x, y)

Step 1 : Compute Non Derivable Itemsets(NDI) of G.
Step 2 : Compute **CNS** (x, y).
Step 3 : Extract set of cliques *CliqSet* formed with the nodes of CNS from NDI
Step 4 :
for each $C \in CliqSet$ **do**
  for each $F \subseteq C$ **do**
    Compute *Temporal-Weight*(F) as in eq 5.1.
  end for
end for
Step 5 : Construct MRF with the nodes in **CNS**.
Step 6 : Obtain the joint probability of link (x, y) using junction tree algorithm and return the score as TCOP score.

Computation of clique potentials on an example is illustrated in Section 5.2.3.

5.2.3. Example Illustration

Consider the clique $\{6587, 10497, 37227\}$ in graph 5.1(b) and the factor $F = (10497, 37227)$. From the DBLP dataset, the two authors in factor F have one paper published in 2003 and one paper in 2004. Also consider the factor $F = (38839, 88840)$ of clique $\{88840, 38839, 39232\}$ in the graph of Fig.5.1(a). The two authors 38839 and 88840 have coauthored in the years 1997, 1998 and 1999. If the training period is taken as 1997-2005 and the predicting year as 2006, the computation of *Temporal-Weight*(F)s of the two factors is shown in Table. 5.1.

Table 5.1.: Computation of *Temporal-Weight*(F) of Factors of cliques

Clique(C)	**Factor**(F)	t_{max}	t_{min}	r_F	w_F	**Potential(F)**
{6587, 10497, 37227}	(10497, 37227)	2004	2003	2006-2004=2	$2\left(\frac{1}{2}+\frac{1}{2}+\frac{0}{2}\right)$ =2*1=2	$\frac{2*0.5^2}{\lvert 2004-2003\rvert+1}$ =0.25
{88840, 38839, 39232}	{38839, 88840}	1999	1997	2006-1999=7	$3\left(\frac{1}{3}+\frac{1}{3}+\frac{1}{3}\right)$ =2.999	$\frac{2.999*0.5^7}{\lvert 1999-1997\rvert+1}$ = 0.00778

We can see in Table 5.1 that the recently formed edge $(10497, 37227)$ has more weight compared to the old edge $(38839, 88840)$.

The joint probability inference of the nodes 195 and 6587 using belief propagation in junction tree is illustrated below. The junction tree of Figure 5.1(b) is shown in Figure 5.2.

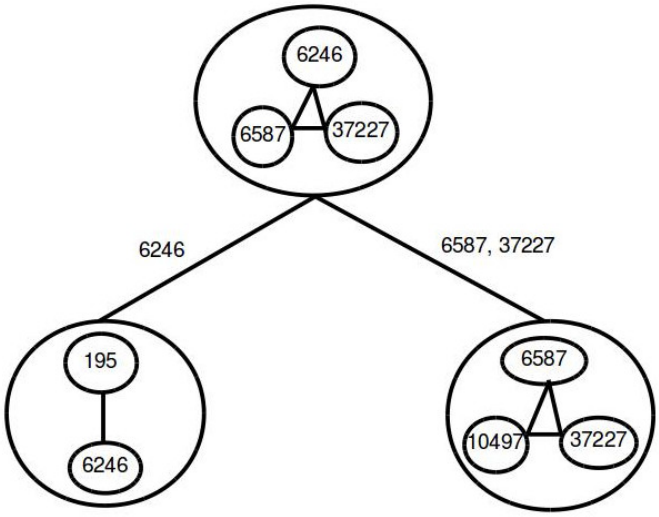


Figure 5.2.: Junction tree of Fig 5.1(b)

We can see in Fig 5.2 that there are 3 cliques and 2 separators.

Cliques :
$C_1 = \{6246, 6587, 37227\}$ $C_2 = \{195, 6246\}$ $C_3 = \{6587, 10497, 37227\}$
Separators :
$S(C_1, C_2) = \{6246\}$ $S(C_1, C_3) = \{6587, 37227\}$

The initial and final clique potential tables of cliques C_1, C_2 and C_3 computed using the Algorithm 4 are given in Tables. 5.2 and 5.6.

To compute the joint probability of nodes 195 and 6587, first pick the clique containing 195, i.e, C_2 and marginalize the two entries corresponding to the value 1 of node 195, which is 0.2158, and then pick the clique containing node 6587, i.e, C_3 and marginalize the entries corresponding to the assignment 1 for node 6587, which is 0.01438. The joint probability is simply the product of 0.2158 * 0.01438 = 0.00316, where as the joint probability obtained for the pair $(39232, 165)$ is 0.000017 much lower than 0.00316.

We evaluate TCOP on four benchmark datasets, DBLP, Condmat, HiePh-collab and HiePh-cite for link prediction. We consider AUROC as well as AUPR as evaluation techniques in this

Table 5.2.: Initial clique potential tables

Table 5.3.: ϕ_{C_1}

6246	6587	37227	$\phi_c(F)$
0	0	0	0
0	0	1	0.25
0	1	0	0.25
0	1	1	0.07916
1	0	0	0.5833
1	0	1	0.05
1	1	0	0.0416
1	1	1	0.0078

Table 5.4.: ϕ_{C_2}

195	6246	$\phi_c(F)$
0	0	0.0546
0	1	0.0428
1	0	0
1	1	0.5833

Table 5.5.: ϕ_{C_3}

6587	10497	37227	$\phi_c(F)$
0	0	0	0.5833
0	0	1	0.05
0	1	0	0
0	1	1	0.25
1	0	0	0.0416
1	0	1	0.079
1	1	0	0.25
1	1	1	0.25

Table 5.6.: Final clique potential tables

Table 5.7.: $\phi(C_1)$

6246	6587	37227	$\phi_c(F)$
0	0	0	0
0	0	1	0.05
0	1	0	0.0039
0	1	1	0.00142
1	0	0	0.2169
1	0	1	0.0096
1	1	0	0.0077
1	1	1	0.00164

Table 5.8.: $\phi(C_2)$

195	6246	$\phi_c(F)$
0	0	0.00962
0	1	0.0158
1	0	0
1	1	0.2158

Table 5.9.: $\phi(C_3)$

6587	10497	37227	$\phi_c(F)$
0	0	0	0.2168
0	0	1	0.00225
0	1	0	0
0	1	1	0.011
1	0	0	0.0016
1	0	1	0.00073
1	1	0	0.00975
1	1	1	0.0023

work.

5.2.4. Results

The implementation set up given in 4.6 is adopted here. Link prediction is carried out using TCOP as an individual measure as well as one of the features to form feature vector in supervised framework.

The performance of TCOP is compared with the best result of neighbourhood based measures, the path based method Katz(KZ), the best random walk based measure PropFlow(PF), the probabilistic measure COP and three temporal measures Time-score (TS), Link-score (LS) and T_Flow (TF). The AUROC and AUPR results obtained for four datasets are given in Table 5.10.

Table 5.10.: AUROC and AUPR results of *TCOP* Vs existing LP measures

Dataset→	Condmat		DBLP		HiePh-collab		HiePh-cite	
Predictor↓	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR
Non-Temporal measures								
Neighbourhood	0.6633	0.0133	0.7415	0.1162	0.6064	0.0089	0.8286	0.0423
Path based	0.6300	0.0018	0.6436	0.0679	0.6190	0.0090	0.7940	0.0378
Random-walk based	0.5825	0.0009	0.6264	0.6795	0.5840	0.0075	0.7933	0.0293
COP	0.7392	0.0266	0.8379	0.2028	0.7153	0.0215	0.8862	0.0901
Temporal measures								
TS	0.6995	0.0172	0.7913	0.1625	0.6752	0.0092	0.8544	0.0613
LS	0.7223	0.0190	0.8016	0.1721	0.6836	0.0110	0.8714	0.0752
TF	0.7168	0.0177	0.8125	0.1785	0.6921	0.0114	0.8662	0.0700
TCOP	**0.7970**	**0.0309**	**0.8590**	**0.2421**	**0.7392**	**0.0320**	**0.9147**	**0.1991**
Supervised								
Supervised	**0.8317**	**0.0489**	**0.9351**	**0.4364**	**0.8434**	**0.0623**	**0.9453**	**0.3733**

It can be seen in Table 5.10 that if we do not include COP, TS gives better accuracy as compared to the standard non-temporal neighbourhood-based measures in terms of AUROC and AUPR. LS and TF give similar performance in case of all the four datasets. TS performs slightly better than LS on sparse networks of DBLP and HiePh-collab and in case of dense networks of Condmat and HiePh-cite, LS is better than TF.

However, TCOP gives the best result out of all other measures, for all datasets, whether the network is sparse or dense. For Condmat dataset, the probabilistic measure COP gives good performance of 73.92% among non-temporal measures and when time information is added to, the performance of TCOP reaches 79.70%. AUPR also improved from 0.026 to 0.031. For DBLP dataset also, TCOP gives the performance of 85.9% over all other measures. In the case of supervised framework for DBLP network, the base performance obtained by using the 5 non-temporal measures, as features, the prediction performance obtained is 89%, (as shown in Table 5.10). With temporal measures of TS, LS, TF and TCOP further added as features, the accuracy further improved to 92.8%. Similar improvement in accuracy is observed for HiePh-collab and HiePh-cite datasets as well. Clearly, TCOP achieves a superior performance for link prediction for all datasets Condmat, DBLP, HiePh-collab and HiePh-cite.

In Chapter 3, standard measures have been extended to bipartite and heterogeneous networks. We proposed *Hetero*-CN, *Hetero*-JC, *Hetero*-PA, *Hetero*-AA, *Hetero*-KZ, *Hetero*-PR, *Hetero*-PF. In the next section we extend all the above mentioned temporal measures to bipartite

and heterogeneous networks.

5.3. Extension of Temporal Measures to Heterogeneous Networks

The notation is given in detail in 3.3. To recall the important definitions, a meta-path is a path connecting two nodes through homogeneous or heterogeneous links. $\Gamma_k(x)$ represents the k-hop neighbourhood of node x, which contains k-hop distance neighbours of x. $P_l(x,y)$ denotes set of all meta-paths of length ranging from 2 to l connecting x to y.

5.3.1. Hetero-Time-Score(Hetero-TS)

We extend Time-Score measure proposed for homogeneous networks [43] to heterogeneous environment as follows:

$$Hetero-TS(x,y) = \sum_{p \in P_3(x,y)} \frac{w(p) * \beta^{r(p)}}{|latest(p) - oldest(p)| + 1} \tag{5.2}$$

where $w(p)$ is equal to the harmonic mean of edge weights of edges in p, β is a damping factor ($0<\beta<1$), $latest(p) = \max_{e \text{ on } P_3}(t(e))$, $oldest(p) = \min_{e \text{ on } P_3}(t(e))$ and r is a recency factor, defined as $r(p)= current\_time - latest(p)$.

5.3.2. Hetero-Link-Score(Hetero-LS)

Choudhary et al. extend the Time-score measure to obtain a path based measure called *Link-score* [44] for homogeneous networks. To obtain the *Link-score* between a pair of nodes x and y which are not directly connected, the authors define a Time Path Index (TPI) on each path p between the nodes x and y. TPI evaluates path weight based on time stamps of links involved in a path. Link-score is the sum of TPI of each path between the nodes x and y.

We extend *Link-score* to heterogeneous network by considering *meta-paths* between two nodes instead of paths containing only homogeneous links. The modified definitions of TPI and *Link-Score* are given in equation 5.3.

$$TPI_p = \frac{w(p) * \beta^{current\_time - avg(p)}}{|current\_time - latest(p)| + 1} \tag{5.3}$$

where $avg(p)$ is the average active year, which is the average of years of recent interaction of edges on meta-path p and all other are as defined in equation 5.2.

$$Hetero-LS(x,y) = \sum_{l=2}^{L} \frac{Avg(TPI_{P_l(x,y)})}{l-1} \tag{5.4}$$

where L is the maximum length of meta-path between nodes i and j.

$Hetero$-LS is applicable to bipartite environment by considering the heterogeneous links between nodes x and y because there are no homogeneous links in such environment.

5.3.3. Hetero-T_Flow(Hetero-TF)

T_Flow [45] is a random-walk based measure, which is an extension of PropFlow measure defined in [18]. Munasinghe et.al [45] define T_Flow that computes the information flow between a pair of nodes x and y through all random-walks starting from node x to node y including link weights as well as activeness of links by giving more weight to recently formed links recursively and take the summation.

We extend T_Flow measure to heterogeneous and bipartite networks by considering heterogeneous edges for bipartite networks and both homogeneous and heterogeneous links for each meta-path for heterogeneous networks. $Hetero$-TF is defined as:

$$Hetero-TF(x,y) = \sum_{i=2}^{l} \sum_{p \in P_l(x,y)} \sum_{\forall e \in p} Hetero-TF(z_1,z_2) * (1-\alpha)^{r(p)} \tag{5.5}$$

If $(x,y) \in E$, then $Hetero-TF(x,y)$ is given by

$$Hetero-TF(x,y) = Hetero-TF(a,x) * \frac{w(x,y)}{\sum_{z \in \Gamma(x)} w(x,z)} * (1-\alpha)^{r(p)} \tag{5.6}$$

where t_x is the time stamp of the link when the random walk visits the node x and t_y is the time stamp of the link when the random walk visits node y.

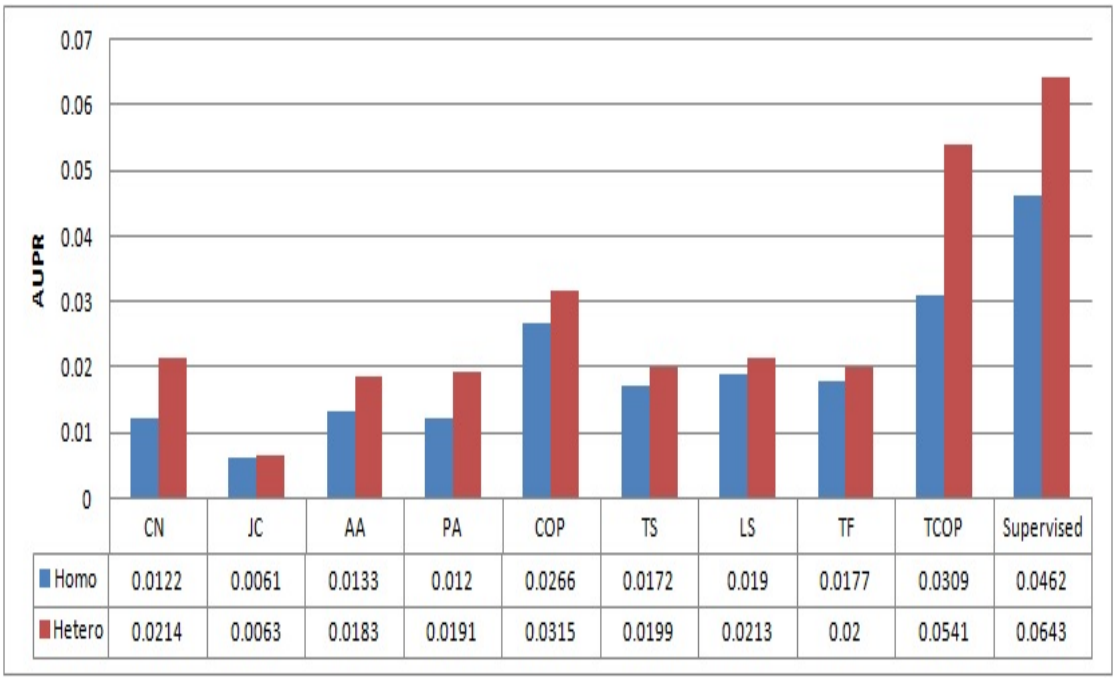
5.3.4. Hetero-TCOP

We extend the newly proposed measure TCOP in section 5.2 to heterogeneous environment in this section. We extend $TCOP$ to heterogeneous environment following the same approach given in

4.5 with a modification of computing temporal weights instead of non-temporal weights to the factors.

5.3.5. Results

The performance of *Hetero-TCOP* is compared with Hetero-Common Neighbor (CN), Hetero-Jaccard Coefficient (JC), Hetero-Adamic Adar (AA), Hetero-Preferential Attachment (PA), Hetero-Co-occurrence Probability (COP), Hetero-Time-score (TS), Hetero-Link-score (LS) and Hetero-T_Flow (TF). Similarly, the prediction performance of *Bipartite-TCOP* is compared with all bipartite versions specified in Table.3.1. The AUROC and AUPR results obtained for Condmat, DBLP, HIePh-collab and HiePh-cite bibliographic networks are tabulated in Tables 5.11,5.12. The improvement in AUPR is more visible in a histogram and therefore, the histograms are given in Fig.5.3, 5.4, 5.5, 5.6, 5.7, 5.8, 5.9.



	CN	JC	AA	PA	COP	TS	LS	TF	TCOP	Supervised
Homo	0.0122	0.0061	0.0133	0.012	0.0266	0.0172	0.019	0.0177	0.0309	0.0462
Hetero	0.0214	0.0063	0.0183	0.0191	0.0315	0.0199	0.0213	0.02	0.0541	0.0643

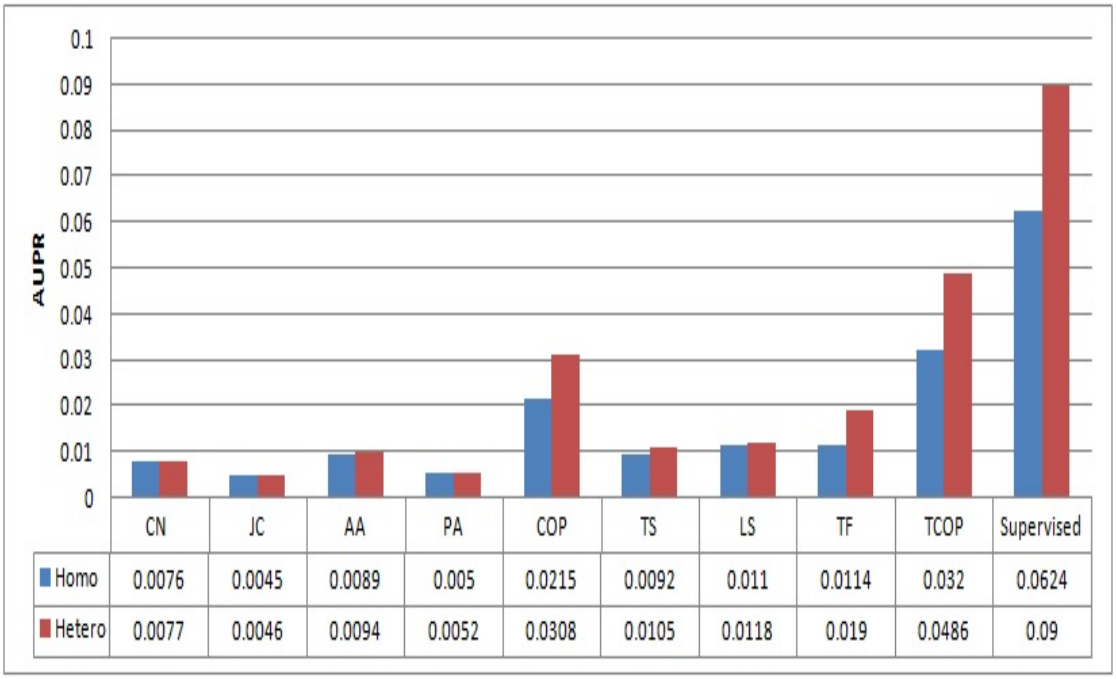
Figure 5.3.: AUPR for **homogeneous** (author-author) link prediction in Condmat network

Table 5.11.: Link Prediction performance of author-conf/journal heterogeneous link on **Condmat**, **DBLP**, **HiePh-collab** and **HiePh-cite** networks

Relation →	Condmat				DBLP				HiePh-collab				HiePh-cite			
Evaluation Measure →	AUROC		AUPR		AUROC		AUPR		AUROC		AUPR		AUROC		AUROC	
LP ↓	Bipartite	Hetero	Bipartite	Hetero	Bipartite	Hetero	Bipartite	Hetero	Bipartite	Hetero	Bipartite	Hetero	Bipartite	Hetero	Bipartite	Hetero
Hetero-CN	0.5312	0.5630	0.0016	0.0023	0.5243	0.5571	0.0014	0.0014	0.5030	0.5350	0.0009	0.0023	0.7615	0.7832	0.0201	0.0410
Hetero-JC	0.4921	0.5111	0.0007	0.0015	0.5019	0.5201	0.0013	0.0015	0.5113	0.5313	0.0013	0.0020	0.7201	0.7315	0.0098	0.0103
Hetero-AA	0.6314	0.6623	0.0025	0.0036	0.6061	0.6370	0.0028	0.0031	0.5521	0.5821	0.0080	0.0086	0.7813	0.8153	0.0313	0.0517
Hetero-PA	0.5810	0.6121	0.0018	0.0025	0.5619	0.5625	0.0015	0.0019	0.5200	0.5326	0.0018	0.0019	0.7751	0.8000	0.0305	0.0412
Hetero-COP	0.7154	0.7513	0.00581	0.0197	0.6861	0.7196	0.0170	0.0123	0.6699	0.6801	0.0110	0.0201	0.8615	0.8953	0.293	0.437
Hetero-TS	0.6812	0.6941	0.0039	0.0054	0.6692	0.6783	0.0098	0.0142	0.6313	0.6601	0.0074	0.0091	0.8013	0.8324	0.0483	0.081
Hetero-LS	0.6995	0.7416	0.0045	0.0098	0.6714	0.6899	0.0132	0.0155	0.6521	0.6712	0.0089	0.0093	0.8421	0.8753	0.0901	0.5130
Hetero-TF	0.6900	0.7312	0.0043	0.0069	0.6790	0.6913	0.0147	0.0193	0.6666	0.6799	0.0094	0.0099	0.8215	0.8500	0.0732	0.0992
Hetero-TCOP	**0.8014**	**0.8513**	**0.0532**	**0.0747**	**0.7093**	**0.7530**	**0.0251**	**0.0410**	**0.6890**	**0.7104**	**0.0230**	**0.0314**	**0.9301**	**0.9642**	**0.6215**	**0.6800**
Supervised	**0.8621**	**0.9015**	**0.0789**	**0.0910**	**0.6591**	**0.6825**	**0.0089**	**0.0099**	**0.5900**	**0.6103**	**0.0121**	**0.0398**	**0.9615**	**0.9701**	**0.6920**	**0.7321**

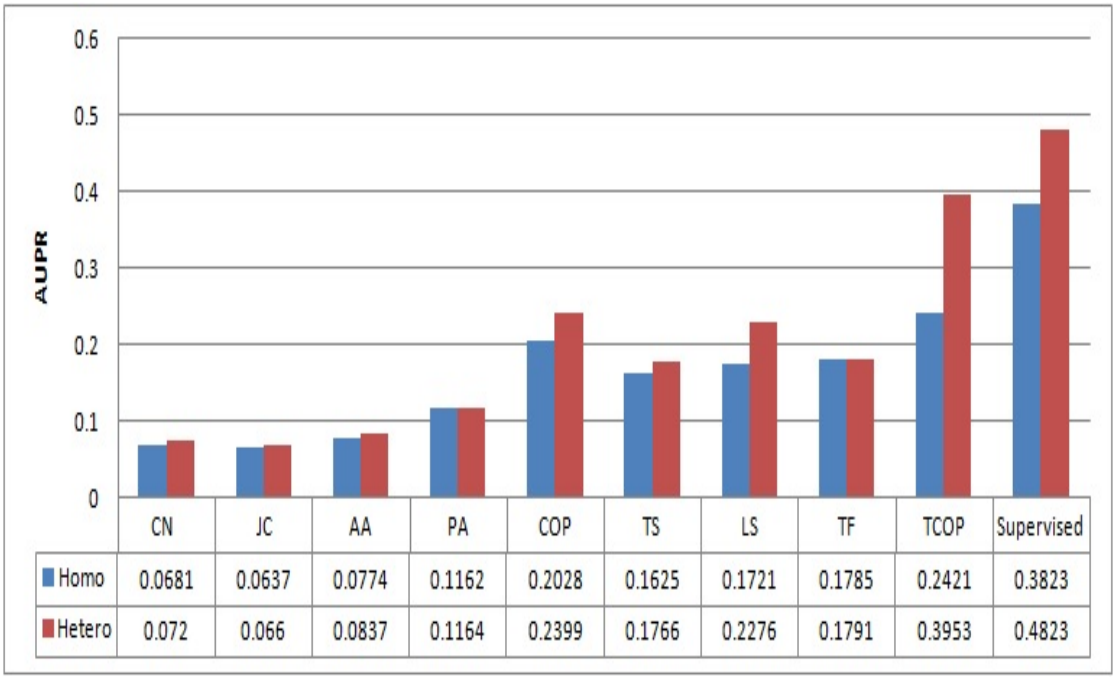
Table 5.12.: Link Prediction performance of co-author relation on **Condmat**, **DBLP**, **HiePh-collab** and **HiePh-cite** networks

Relation →	Condmat				DBLP				HiePh-collab				HiePh-cite			
Evaluation Measure →	AUROC		AUPR		AUROC		AUPR		AUROC		AUPR		AUROC		AUROC	
LP ↓	Homo	Hetero	Homo	Hetero	Homo	Hetero	Homo	Hetero	Homo	Hetero	Homo	Hetero	Homo	Hetero	Homo	Hetero
Hetero-CN	0.6193	0.6201	0.0122	0.0214	0.6504	0.6811	0.0681	0.0720	0.5846	0.5920	0.0076	0.0077	0.8147	0.8413	0.0381	0.0453
Hetero-JC	0.5086	0.5113	0.0061	0.0063	0.5942	0.6012	0.0637	0.0660	0.5167	0.5169	0.0045	0.0046	0.8272	0.8595	0.0385	0.0.0494
Hetero-AA	0.6633	0.6843	0.0133	0.0183	0.6777	0.7011	0.0774	0.0837	0.6064	0.6253	0.0089	0.0094	0.8260	0.8613	0.0383	0.0555
Hetero-PA	0.5853	0.6121	0.0120	0.0191	0.7415	0.7423	0.1162	0.1164	0.5466	0.5500	0.0050	0.0052	0.8286	0.8600	0.0423	0.0583
Hetero-COP	0.7392	0.7751	0.0266	0.0315	0.8379	0.8439	0.2028	0.2399	0.7153	0.7390	0.0215	0.0308	0.8862	0.9201	0.0901	0.1999
Hetero-TS	0.6995	0.7123	0.0172	0.0199	0.7913	0.8290	0.1625	0.1766	0.6752	0.6801	0.0092	0.0105	0.8544	0.8843	0.0613	0.0649
Hetero-LS	0.7223	0.7461	0.0190	0.0213	0.8016	0.8376	0.1721	0.2276	0.6836	0.6900	0.0110	0.0118	0.8714	0.8991	0.0715	0.0752
Hetero-TF	0.7168	0.7240	0.0177	0.0200	0.8125	0.8263	0.1785	0.1791	0.6921	0.7000	0.0114	0.0190	0.8662	0.8793	0.0700	0.0693
Hetero-TCOP	**0.7970**	**0.8303**	**0.0309**	**0.0541**	**0.8590**	**0.8934**	**0.2421**	**0.3953**	**0.7392**	**0.7575**	**0.0320**	**0.0486**	**0.9147**	**0.9526**	**0.1991**	**0.3640**
Supervised	**0.8431**	**0.8814**	**0.0462**	**0.0643**	**0.7521**	**0.7724**	**0.1543**	**0.1598**	**0.6591**	**0.6834**	**0.0096**	**0.0106**	**0.9386**	**0.9618**	**0.3619**	**0.4125**



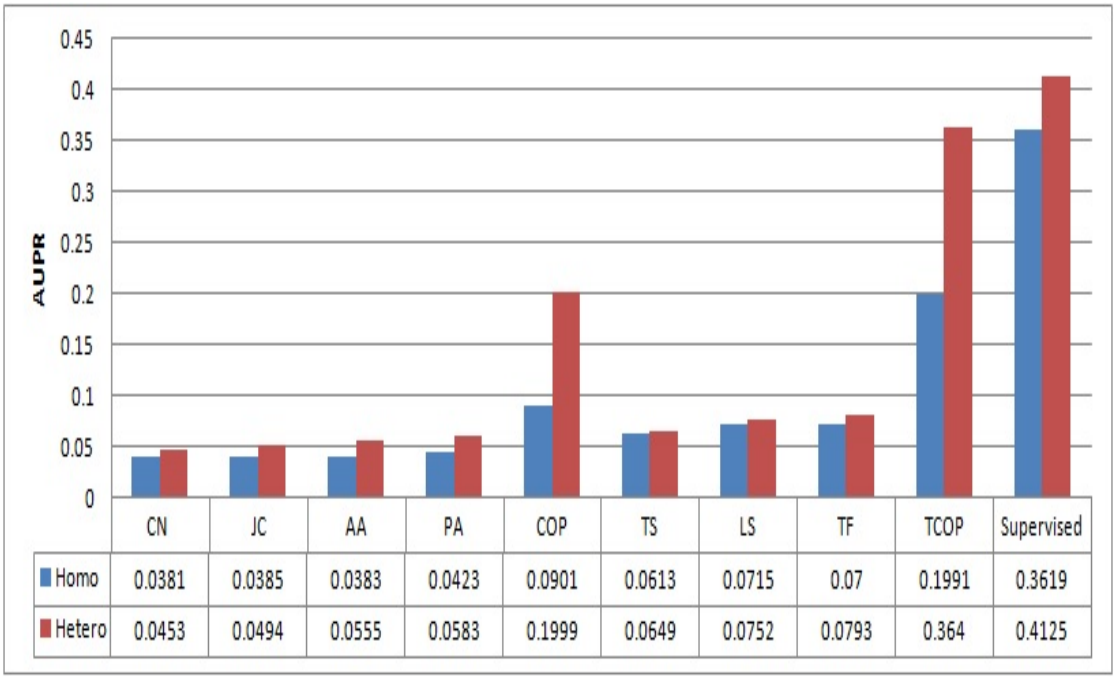
	CN	JC	AA	PA	COP	TS	LS	TF	TCOP	Supervised
Homo	0.0076	0.0045	0.0089	0.005	0.0215	0.0092	0.011	0.0114	0.032	0.0624
Hetero	0.0077	0.0046	0.0094	0.0052	0.0308	0.0105	0.0118	0.019	0.0486	0.09

Figure 5.5.: AUPR scores of **homogeneous links** using temporal info for HiePh-collab



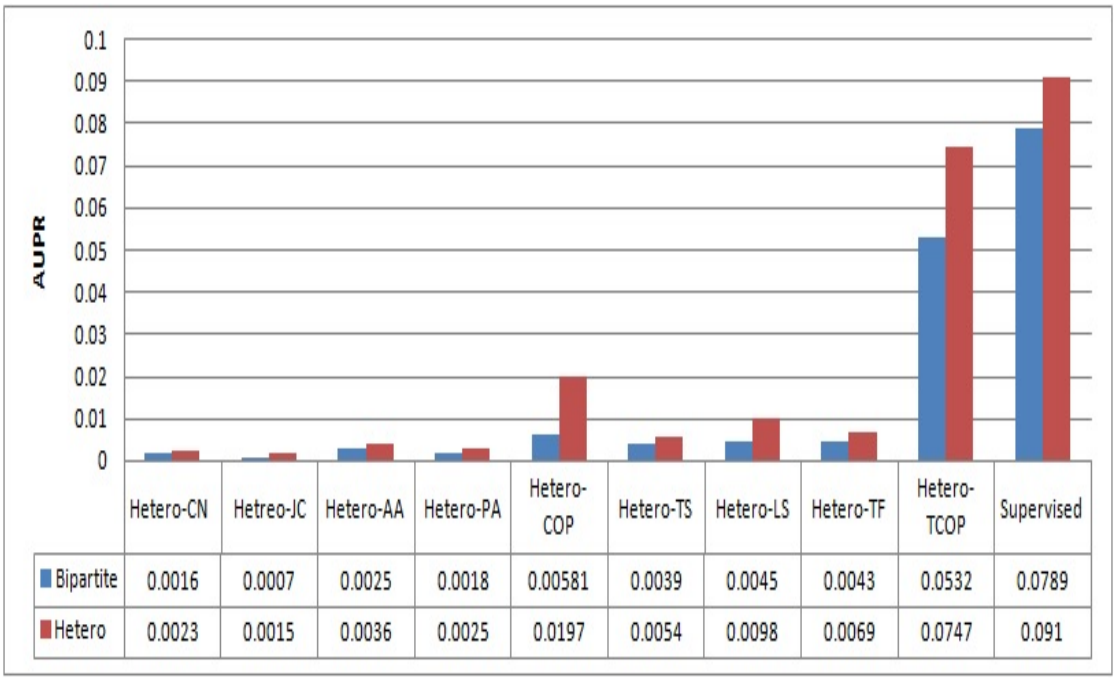
	CN	JC	AA	PA	COP	TS	LS	TF	TCOP	Supervised
Homo	0.0681	0.0637	0.0774	0.1162	0.2028	0.1625	0.1721	0.1785	0.2421	0.3823
Hetero	0.072	0.066	0.0837	0.1164	0.2399	0.1766	0.2276	0.1791	0.3953	0.4823

Figure 5.4.: AUPR for **homogeneous** link prediction in DBLP network



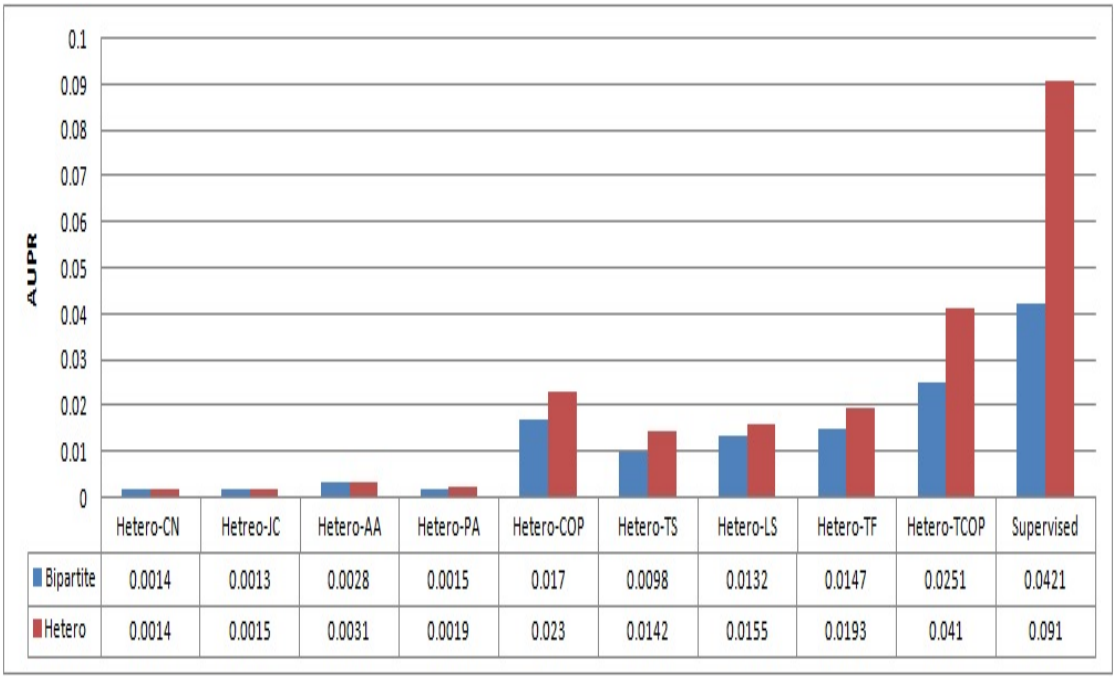
	CN	JC	AA	PA	COP	TS	LS	TF	TCOP	Supervised
Homo	0.0381	0.0385	0.0383	0.0423	0.0901	0.0613	0.0715	0.07	0.1991	0.3619
Hetero	0.0453	0.0494	0.0555	0.0583	0.1999	0.0649	0.0752	0.0793	0.364	0.4125

Figure 5.6.: AUROC scores of **homogeneous links** using temporal info for Condmat



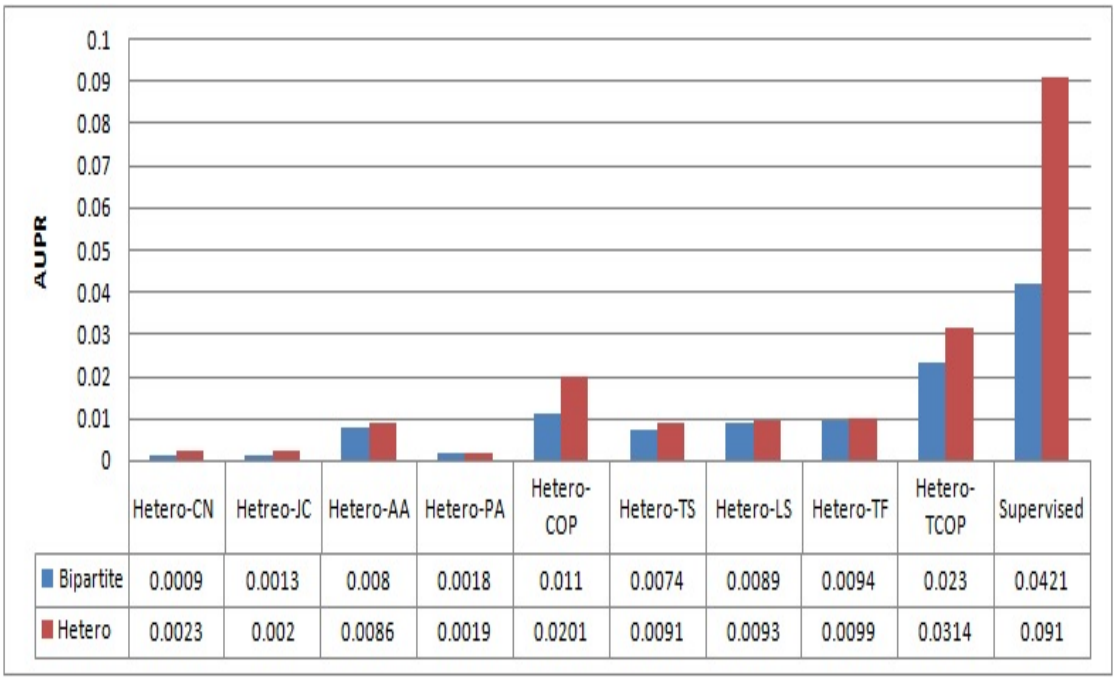
	Hetero-CN	Hetreo-JC	Hetero-AA	Hetero-PA	Hetero-COP	Hetero-TS	Hetero-LS	Hetero-TF	Hetero-TCOP	Supervised
Bipartite	0.0016	0.0007	0.0025	0.0018	0.00581	0.0039	0.0045	0.0043	0.0532	0.0789
Hetero	0.0023	0.0015	0.0036	0.0025	0.0197	0.0054	0.0098	0.0069	0.0747	0.091

Figure 5.7.: AUPR scores of **heterogeneous links** using temporal info for Condmat



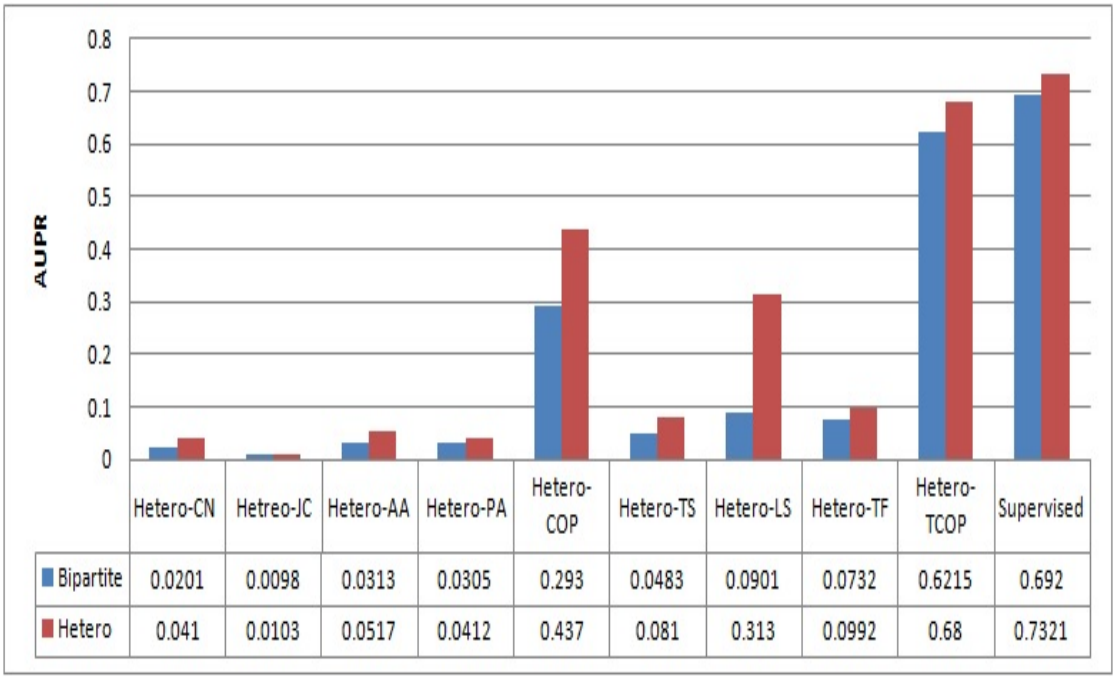
	Hetero-CN	Hetreo-JC	Hetero-AA	Hetero-PA	Hetero-COP	Hetero-TS	Hetero-LS	Hetero-TF	Hetero-TCOP	Supervised
Bipartite	0.0014	0.0013	0.0028	0.0015	0.017	0.0098	0.0132	0.0147	0.0251	0.0421
Hetero	0.0014	0.0015	0.0031	0.0019	0.023	0.0142	0.0155	0.0193	0.041	0.091

Figure 5.8.: AUPR scores of **heterogeneous links** using temporal info for DBLP



	Hetero-CN	Hetreo-JC	Hetero-AA	Hetero-PA	Hetero-COP	Hetero-TS	Hetero-LS	Hetero-TF	Hetero-TCOP	Supervised
Bipartite	0.0009	0.0013	0.008	0.0018	0.011	0.0074	0.0089	0.0094	0.023	0.0421
Hetero	0.0023	0.002	0.0086	0.0019	0.0201	0.0091	0.0093	0.0099	0.0314	0.091

Figure 5.9.: AUPR scores of **heterogeneous links** using temporal info for HiePh-collab



	Hetero-CN	Hetreo-JC	Hetero-AA	Hetero-PA	Hetero-COP	Hetero-TS	Hetero-LS	Hetero-TF	Hetero-TCOP	Supervised
Bipartite	0.0201	0.0098	0.0313	0.0305	0.293	0.0483	0.0901	0.0732	0.6215	0.692
Hetero	0.041	0.0103	0.0517	0.0412	0.437	0.081	0.313	0.0992	0.68	0.7321

Figure 5.10.: AUPR scores of **heterogeneous links** using temporal info for HiePh-cite

Fig.5.11,5.12 show the ROC curves of DBLP network are shown in Fig.

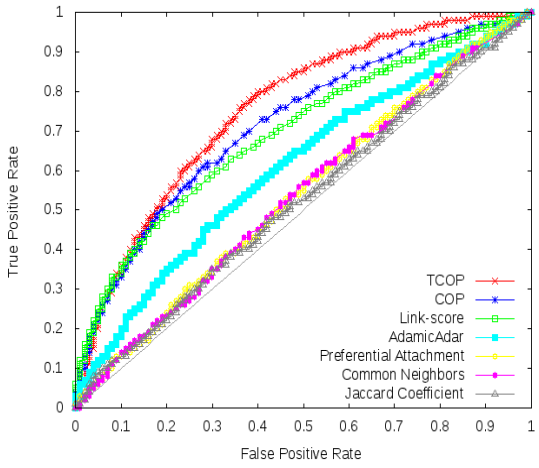


Figure 5.11.: ROC curve for predicting auth-conf heterogeneous links in DBLP TCOP Vs Non-temporal measures

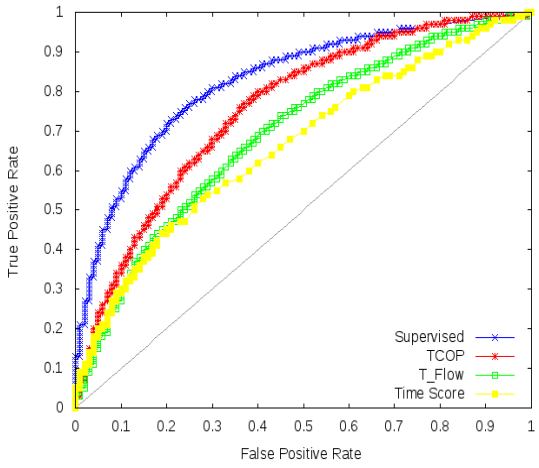


Figure 5.12.: ROC curve for predicting auth-conf heterogeneous links in DBLP TCOP Vs Temporal measures

5.3.6. Discussion

COP performs best among all the non-temporal measures CN, JC, AA and PA, while TCOP proves to be better over all the temporal as well as non-temporal measures. The proposed *Hetero*-TCOP shows superior performance over all 8 measures for all the four bibliographic networks.

We analyse a few True Positives discovered by *Hetero*-TCOP that are missed by *COP* as well as the other measures; and False Positives of other measures which are rightly rejected by *Hetero*-TCOP. Consider a snapshot of DBLP heterogeneous network in Fig.5.13.

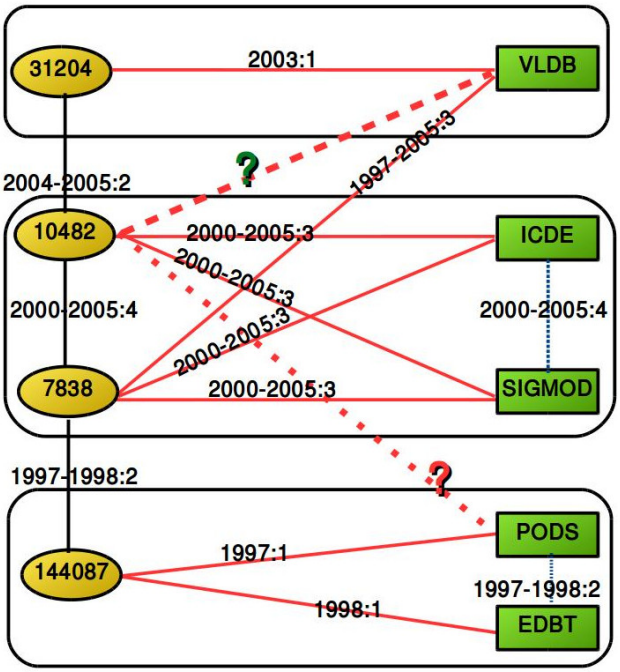


Figure 5.13.: A snapshot of DBLP heterogeneous network with time information

10482	7838	ICDE	SIGMOD	$\phi_C(F)$
0	0	0	0	0.00000
0	0	0	1	0.00000
...	...	...	...	...
0	1	0	1	0.00000
...	...	...	...	...
0	1	1	0	0.50000
...	...	...	...	...
1	1	0	1	0.50000
1	1	1	0	0.08335
...	...	...	...	...
1	1	1	1	0.08335

Table 5.13.: A partial Clique Potential Table $\phi_C(F)$ of H-clique $C = \{10482, 7838, ICDE, SIGMOD\}$

Table 5.14.: A H-clique extracted from DBLP heterogeneous network

DBLP network contains a link between the author node *10482* and the conference node *VLDB* in the year 2006. *Hetero*-TCOP predicts a link between the author *10482* and the conference *VLDB* as the links involved are latest, but the standard link prediction measures compute a low score between *10482* and *VLDB*, as there are more meta-paths of length greater than 2 between them. In the other case, DBLP does not contain a link between the author node *10482* and the conference node *PODS* in the year 2006. Neighbourhood-based measures as well as COP predict

a link between the author *10482* and the conference *PODS*, as many meta-paths exist between them through author nodes *7838* and *144087* which are old links. *Hetero*-TCOP ranks this low as the links on meta-paths are old.

More improvement in prediction performance is observed for probabilistic measures COP and TCOP over non-probabilistic measures in both bipartite and heterogeneous networks. From Table.5.11, one can see that the performance of *Hetero-TCOP* is improved by 5% over *Bipartite-TCOP* for DBLP network and *Hetero-COP* is improved by 4% over *Bipartite-COP*. Similar is the case with HiePh-collab dataset also. Prediction accuracy is increased in DBLP which is sparse network over the dense network HiePh-collab. In machine learning framework, the prediction accuracy is improved from 66% to 68% for heterogeneous links for DBLP heterogeneous network and from 59% to 62% for HiePh-collab network.

In the case of homogeneous(author-author) link prediction, the improvement is less (Table.5.12). More improvement in the prediction performance is observed for homogeneous links (author-author) when compared to the performance of heterogeneous links (author-conference). *Hetero-TCOP* has shown an accuracy of 9% over neighbourhood-based measures and 6% over temporal measures in heterogeneous environment.

An average improvement of around 5% is observed for temporal measures over non-temporal measures, in the prediction of both homogeneous as well as heterogeneous links.

5.4. Conclusion

Common neighbour based methods are computationally efficient in the field of link prediction in social networks. A new measure called Temporal Co-occurrence Probability is proposed for link prediction in social networks, which is an extension of COP. We make use of temporal information of links which is readily available in collaborative networks in a natural way to positively weigh recent author cliques in comparison to old cliques. Through an extensive experimentation, it is demonstrated that TCOP performs better than state-of-the art prediction algorithms. The results obtained on two sparse and two dense collaborative networks show that, TCOP predicts future collaborations more accurately compared to all the other measures in literature, whether the network is dense or sparse. When TCOP is used as a feature in supervised framework, the prediction accuracy is further enhanced. The results obtained on HiePh-cite are prominently higher than all the other networks which needs further investigation from the perspective of network analysis.

Most of the existing social networks are heterogeneous in nature. This work shows the importance of including temporal information for link prediction. Utilizing both temporal and heterogeneous information proved to be a very successful strategy in enhancing the performance of the probabilistic measures like *COP*.

In the next chapter, we propose a scalable approach for link prediction based on community discovery.

6. Community based Approach for Link Prediction

Various link prediction measures for homogeneous and heterogeneous networks are discussed in previous chapters. In large social networks, time and memory are major challenges for link prediction. In this chapter, a scalable approach is proposed based on the network structure. The proposed method called Community based Link Prediction (*CBLP*), computes communities of the network and computes prediction scores within each community in parallel. The idea is applicable for a heterogeneous environment, if a community discovery algorithm is available for a non-trivial heterogeneous network. To the best of our knowledge, such algorithms are not available. Hence, we restrict the discussion of the ideas underlying community based approach to only multi-relational networks. The results of implementation of *CBLP* on bench-mark datasets show that community information does significantly help in improving the performance of link prediction for multi-relational networks.

6.1. Motivation

Social networks exhibit a natural community structure. The communities in social networks are groups of nodes sharing common properties[73]. For example, the group of web-pages representing same content may be treated as a community[74]. In biological networks, a community forms by the functional modules sharing same cycles/pathways [75]. In friends networks, a community may represent the group of people sharing similar interests[76]. The community detection algorithms discover a pertinent community structure in the social graphs.

The nodes in a community are densely connected to one another and sparsely connected to the nodes in other communities. The neighbours of a node within community affects future collaborations of the node differently from the neighbours in the other communities. The links

tend to form between nodes lying within community than nodes in different communities. The existing link prediction measures treat the whole graph as a single community and generate pairs of nodes belonging to different communities as candidate pairs for prediction. This may lead to many false positives due to the fact that the two nodes in the pair may be present in different communities and may not form an edge in future as shown in Fig.6.1.

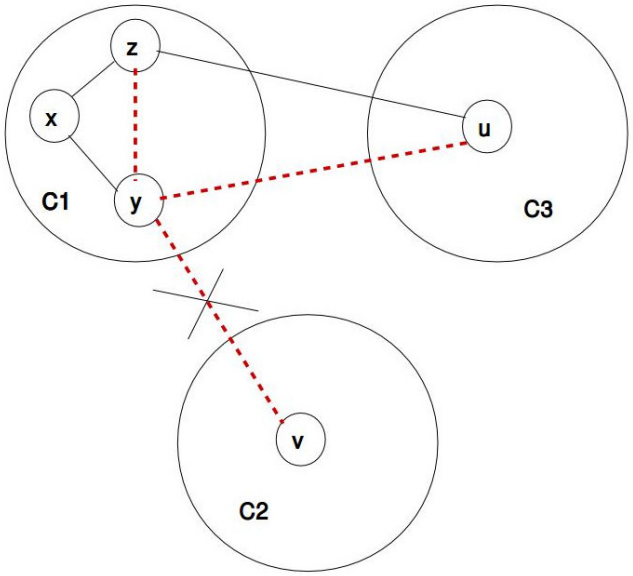


Figure 6.1.: Community Structure in graphs

In this work, it is proposed that the rate of false positives can be reduced significantly if the link prediction is performed between the nodes within the community. The time taken by the link prediction algorithms can be significantly reduced by data parallelization based on community information.

6.2. Existing Community Discovery Algorithms

Plenty of community discovery algorithms for homogeneous networks exist in the literature [77, 73, 78, 79, 80, 81, 82]. In the case of heterogeneous networks, community discovery is performed on each homogeneous projection of the network. This approach may lead to loss of information flowing through multiple node and edge types. A community discovery algorithm for dynamic networks is proposed in [83]. Tang et al. propose a community discovery algorithm for multi-

relational networks in [61]. The authors represent a multi-relational graph as a set of adjacency matrices($A^{(s)}$), one for each type of interaction s, and define four types of integrations on these matrices namely partition integration, network integration, utility integration and feature integration. Partition Integration combines multiple clustering results of the same data from a variety of sources into a single consensus clustering based on majority rule. K-means clustering algorithm that is applied to each dimension can introduce more uncertainty as it is highly sensitive to initial conditions. This is seen as consensus clustering[84]. Network Integration approach handles a multi-dimensional network is featured as single dimensional network and calculates the average interaction network among the nodes as follows.

$$A' = \frac{1}{m}\sum_{s=1}^{m} A^{(s)} \tag{6.1}$$

Any community discovery algorithm can be applied on average network, A'. The limitation of this approach is simply averaging over all the dimensions would overwhelm the structural information in other dimensions.

The third type of integration is utility integration. Utility Matrix(Modularity Matrix) is computed for each interaction s as follows:

$$B_s = A_s - \frac{d_i d_s^T}{2m_s} \tag{6.2}$$

Then the total modularity matrix is computed as follows.

$$B = \frac{1}{m}\sum_{s=1}^{m} B_s \tag{6.3}$$

Then community discovery algorithm is applied on B. This approach maximizes the total modularity among all dimensions. The limitation of this approach is ambiguity on whether modularity can be compared in different dimensions.

The last one is Feature Integration (also called as Principal Modularity maximization(PMM)). Principal Modularity Maximization (PMM) method has the following steps: Firstly, PMM computes modularity matrix for each relation type. Then Principal Component Analysis (PCA) is applied on these matrices to select the top eigenvectors. Once the data is projected onto the principal vectors, a lower-dimensional embedding is obtained, which captures the principal pattern across all the types of relations in the network. Then K-means clustering method is applied on this

embedding to discover the community labels.

Tang. et al have shown that Principal Modularity Maximization based community division is superior to other methods in multi-relational networks [61].

6.3. Proposed Approach : Community based Link Prediction (CBLP)

The proposed method, computes communities of the multi-relational network and computes prediction scores within each community in parallel. The main idea in this work is to divide the multi-relational network into clusters and then apply multi-relational supervised link prediction algorithm on each cluster separately. The flow diagram of the proposed approach is shown in Figure 6.2.

This is meaningful only if the communities are dense and fewer edges exist between the clusters. Since social networks exhibit precisely this kind of structure for communities, the clusters formed using the community information are meaningful [85]. The community discovery algorithm of Tang et al. based on PMM is one of the latest algorithms that is shown to have good performance in multi-relational networks [61]. Hence in this work, PMM is used for obtaining clusters of multi relational network.

The link prediction framework for multi relational networks called MR-HPLP has been proposed by Davis et al. in [46, 86] that has been used in the previous chapters is used here. The details for the case of multi-relational networks are given explicitly here.

- Compute features for a node pair (x,y) of each edge type s as : $(F_1(x,y),F_2(x,y),\ldots)$, where $F_i(x,y)$ may be $AA(x,y)$, $CN(x,y)$, $PA(x,y)$, $PF(x,y)$, $\ldots$
- Compute a label for (x,y) of each edge type s as follows.

$$\begin{aligned} l_s(x,y) &= 1 \quad if\ (x,y) \in E_s \\ &= 0 \quad if\ (x,y) \notin E_s. \end{aligned} \tag{6.4}$$

- Now combine all the feature vectors of (x,y) for each edge type s along with their labels. Therefore, the the overall feature vector of (x,y) is

$$Feature\_vector(x,y) = \cup_{s=1}^{m}(F_1,F_2,\ldots,F_k)_s \tag{6.5}$$

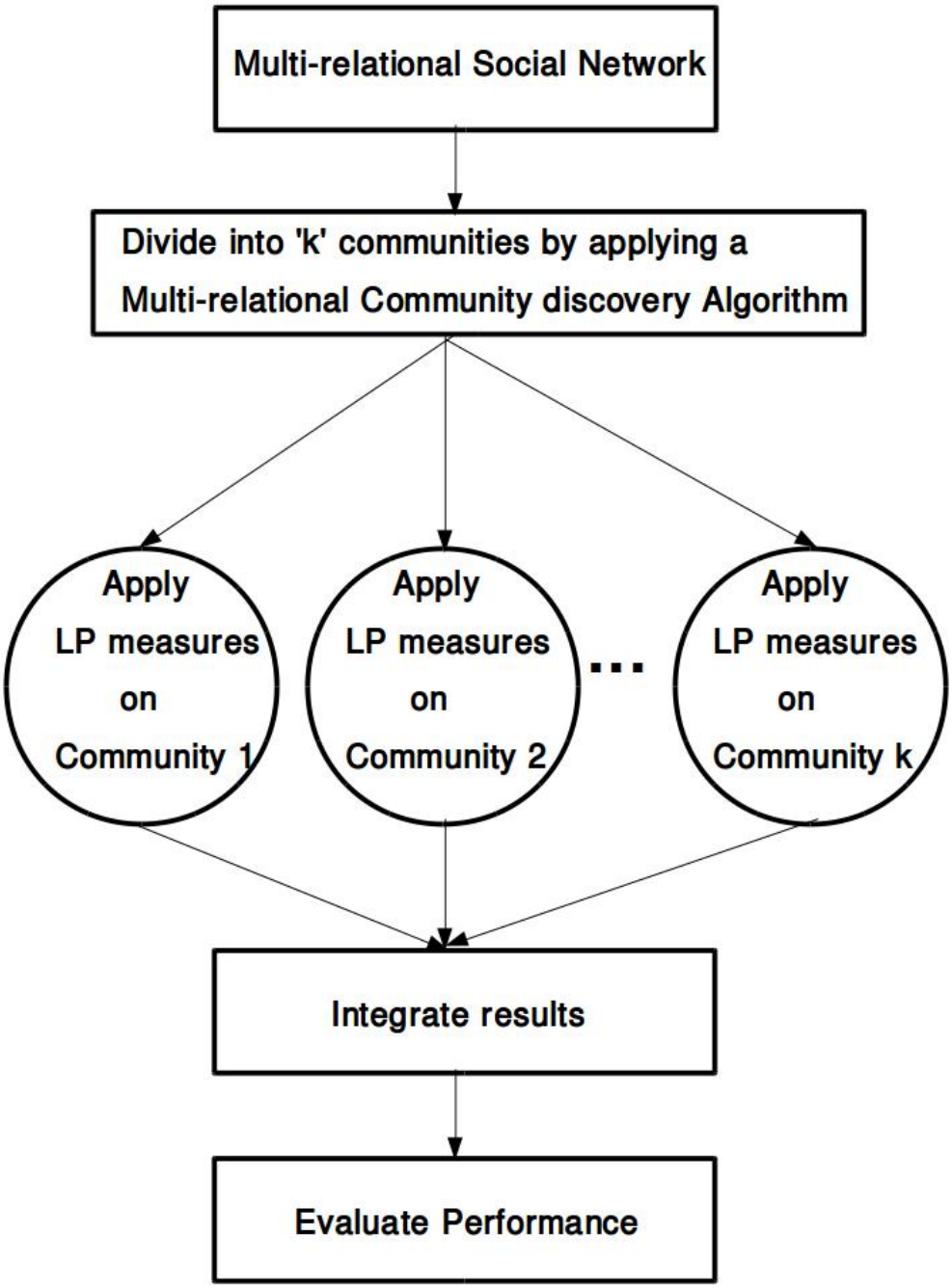


Figure 6.2.: Community Based Link Prediction (CBLP)

- If the target edge type to be predicted between (x,y) is s, then the label l_s is taken as class label.
- As in HPLP, Random forest classification algorithm with bagging and under-sampling is considered.

The MR-HPLP described above is applied on each community in parallel. The proposed method called community based link prediction (CBLP) is given in Algorithm 5.

Algorithm 5 Community Based Link Prediction (CBLP)

Input:

- Network graph $G = (V, E = E_1 \cup E_2 \cup \ldots E_m)$, where V is the set of nodes, E denotes the set of m types of edges
- c : The number of communities

Output: E'_s, the set of predicted links of type s in G.

Step 1: Apply PMM on G to obtain the community labels for nodes in V.
Step 2: $V = (C_1 \cup C_2 \cup \ldots \cup C_c)$, where C_i is i^{th} community and G_i is the induced multi-relational sub graph on C_i.
Step 3:
for $i = 1 \ldots c$ **in parallel do**

  for $s = 1 \ldots m$ **do**
    for each node pair $(x,y) \in G_i$ **do**
      Compute $feature\_vector$, $F_s(x,y) = (f_1, f_2, \ldots)_s$ and label $l_s(x,y)$
    end for
  end for
  Compute overall $feature\_vector$, $F(x,y) = (F_s(x,y))_{s=1,\ldots,m}$ and $l(x,y) = l_s(x,y)$
  Apply MR-HPLP on G_i and obtain E'_s.
end for

6.4. Experimental Evaluation

To evaluate CBLP, a dataset with the availability of community information is needed. Therefore, two datasets are considered for experimentation. The first one is a synthetic benchmark dataset created by Tang et al. [61] and the second dataset is DBLP coauthorship network which has been used in previous chapters. As there is no time of formation of link available on the synthetic dataset, the non-temporal measures are used to evaluate *CBLP* approach and for the *DBLP* dataset, the temporal measures are also evaluated.

The synthetic network is multi-relational, with 350 nodes and four types of relations among nodes. This network having three communities is originally created by the authors for evaluating the multi-relational community discovery algorithm proposed by them. The community information required by the proposed algorithm is available for this data set and hence this dataset is considered for experimentation. The details of this data set are given in 2.6 and provided in the Table 6.1 for ready reference.

Table 6.1.: Synthetic dataset with 350 nodes distributed among 3 communities

Edge Type	$\|Edges\|$
Relation1	12430
Relation2	16850
Relation3	15756
Relation4	15206
Total	60242

The other dataset is DBLP coauthorship dataset used in the previous chapters and statistics are available in 2.6.

6.4.1. Results of Synthetic dataset

For *CBLP*, the whole network is divided into 3 communities as, the bench mark dataset used in [61], is designed for three communities. For discovering communities, PMM algorithm is used. It is interesting to see that when the network is divided into 3 communities, the candidate edges as well as the graph size is drastically reduced. The drastic reduction in the number of candidate edges when divided into communities can be seen in Table 6.2.

Table 6.2.: Candidate node pairs for prediction in the whole network and community wise

Rel.Type	Whole	CommunityWise
Relation1	94680	35124
Relation2	105822	41198
Relation3	80594	35756
Relation4	103970	40074

The proposed *CBLP* is compared with the 8 unsupervised baseline link prediction methods

Table 6.3.: AUROC of LP measures using *CBLP* approach Vs non-*CBLP*

Relation→ LP↓	Relation 1		Relation 2		Relation 3		Relation 4	
	Non-CBLP	CBLP	Non-CBLP	CBLP	Non-CBLP	CBLP	Non-CBLP	CBLP
CN	0.7012	0.7299	0.6078	0.6253	0.7737	0.7953	0.6594	0.6776
JC	0.7273	0.7471	0.6195	0.6200	0.7830	0.7800	0.6805	0.6872
AA	0.7131	0.7395	0.6089	0.6415	0.7747	0.8013	0.6715	0.6866
PA	0.6236	0.7296	0.5108	0.5112	0.5600	0.5713	0.5597	0.6456
PR	0.6063	0.7235	0.5064	0.5066	0.5216	0.5201	0.5549	0.6605
RPR	0.7043	0.7294	0.5617	0.5832	0.7631	0.7715	0.6368	0.6752
KZ	0.6462	0.7232	0.6087	0.6213	0.7728	0.7834	0.6462	0.6942
PF	0.7394	0.7399	0.6166	0.6225	0.7459	0.7513	0.6829	0.6979
COP	0.7589	0.8295	0.6425	0.6923	0.8215	0.8516	0.7934	0.8361
Supervised	0.8416	**0.9055**	0.6853	**0.7216**	0.8593	**0.8932**	0.8874	**0.9231**

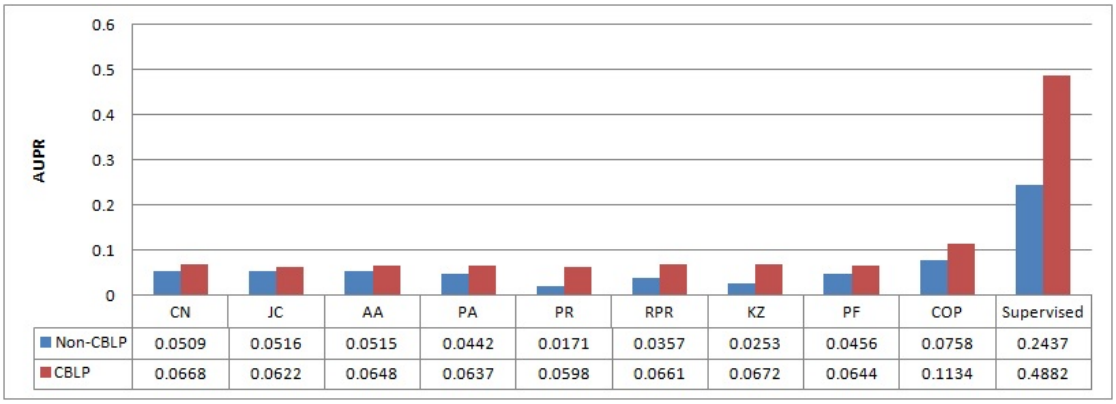
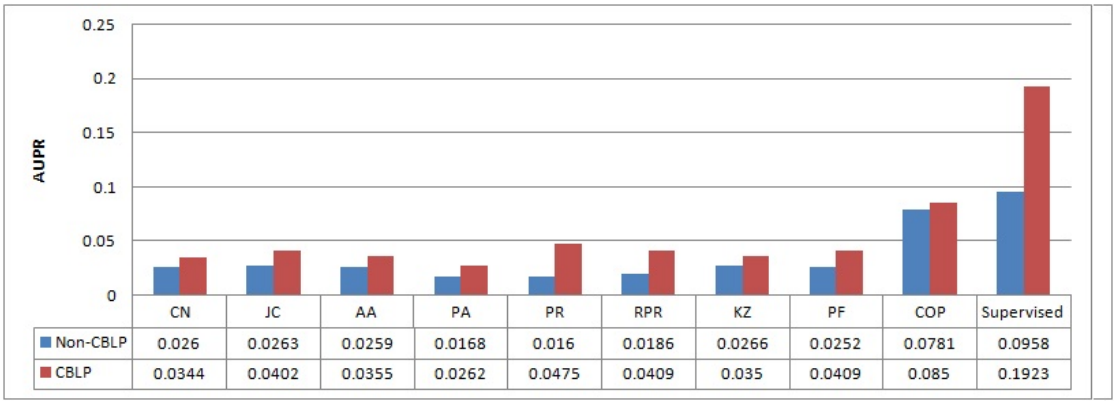


Figure 6.3.: AUPR results of *CBLP* Vs *Non-CBLP* for Relation 1 of **Synthetic Network**

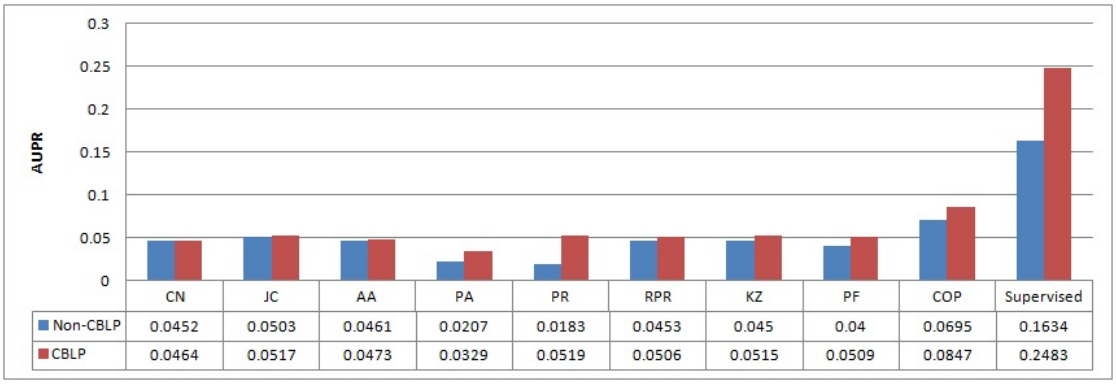
along with supervised framework. The results for AUROC for the relations 1, 2, 3, and 4 are shown in Tables 6.3 and those of AUPR are shown in Fig. 6.3 - 6.6.

For all the four types of relations, COP gives best AUROC and AUPR when networks are considered as a whole (non-*CBLP*) as well as when the network is considered after dividing into communities (*CBLP*). This improvement is anticipated as explained in 6.1 that the links tend to form between the intra-community nodes than inter-community nodes.The false positive rate is also drastically reduced which lead to increase in accuracy. It can be clearly observed from the confusion matrices of Relation 4, given in Table 6.4 and 6.5. Out of 122150 possible edges, 108426 are candidate edges, but in *CBLP*, these are reduced to 40074, after ignoring inter community candidate edges. The improvement of accuracy with community division on Relation



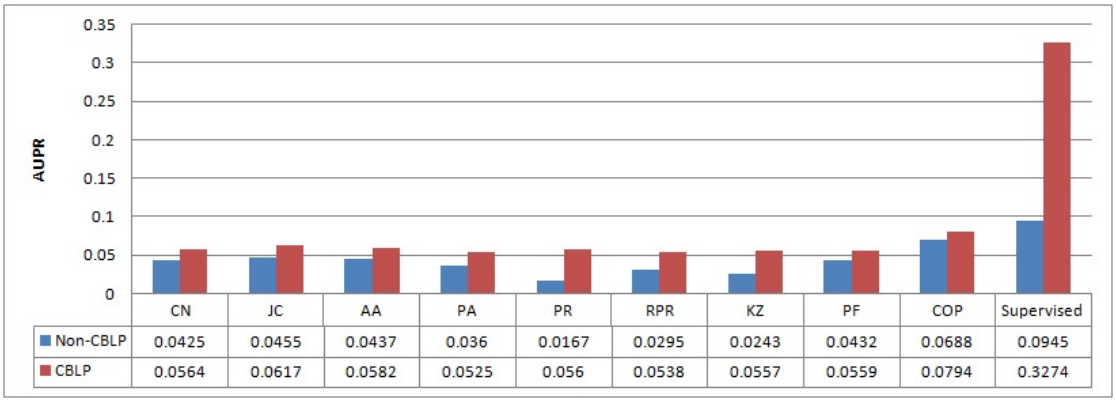
	CN	JC	AA	PA	PR	RPR	KZ	PF	COP	Supervised
Non-CBLP	0.026	0.0263	0.0259	0.0168	0.016	0.0186	0.0266	0.0252	0.0781	0.0958
CBLP	0.0344	0.0402	0.0355	0.0262	0.0475	0.0409	0.035	0.0409	0.085	0.1923

Figure 6.4.: AUPR results of *CBLP* Vs *Non-CBLP* for Relation 2 of **Synthetic Network**



	CN	JC	AA	PA	PR	RPR	KZ	PF	COP	Supervised
Non-CBLP	0.0452	0.0503	0.0461	0.0207	0.0183	0.0453	0.045	0.04	0.0695	0.1634
CBLP	0.0464	0.0517	0.0473	0.0329	0.0519	0.0506	0.0515	0.0509	0.0847	0.2483

Figure 6.5.: AUPR results of *CBLP* Vs *Non-CBLP* for Relation 3 of **Synthetic Network**



	CN	JC	AA	PA	PR	RPR	KZ	PF	COP	Supervised
Non-CBLP	0.0425	0.0455	0.0437	0.036	0.0167	0.0295	0.0243	0.0432	0.0688	0.0945
CBLP	0.0564	0.0617	0.0582	0.0525	0.056	0.0538	0.0557	0.0559	0.0794	0.3274

Figure 6.6.: AUPR results of *CBLP* Vs *Non-CBLP* for Relation 4 of **Synthetic Network**

Table 6.4.: Confusion matrix considering whole network

Predicted → (108426) / Actual (108426) ↓	Pos (103970)	Neg (4456)
Pos (1482)	TP (1444)	FN (38)
Neg (106944)	FP (102526)	TN (4418)

Table 6.5.: Confusion matrix after community division

Predicted → (108426) / Actual (108426) ↓	Pos (40074)	Neg (68352)
Pos (1482)	TP (1088)	FN (394)
Neg (106944)	FP (38986)	TN (67958)

4 is summarized in Table 6.6.

In Table 6.6, we can observe that, precision is improved along with improvement in AUPR and FPR is drastically reduced resulting in increase of AUROC.

CBLP approach not only increases accuracy, but also reduces time and memory requirements. As we are dividing the network into three communities, we can expect that the computation time will be reduced by three times. But When computed on a machine with 12GB RAM and i7 processor, the average time to compute *COP* between a pair of nodes has reduced by 18 times and the memory needed has reduced by 3 times. This is because of two reasons. (a) The network size of each community considered separately is small, from which local MRF of central neighbourhood

Table 6.6.: Improvement of accuracy with community division on Relation 4

Measure	Whole	Community wise
FPR	0.95	0.36
Precision	0.01	0.03
AUROC	0.7934	0.8361
AUPR	0.0688	0.0794

Table 6.7.: AUROC of LP measures for different number of communities for *Relation*1 of synthetic dataset

LP	C = 1	C = 2	C = 3	C = 4	C = 5
CN	0.7012	0.7270	0.7299	0.7016	0.6738
JC	0.7273	0.7311	0.7471	0.6961	0.6719
AA	0.7131	0.7293	0.7395	0.6958	0.6672
PA	0.6236	0.7104	0.7296	0.6582	0.4981
PR	0.6063	0.7064	0.7235	0.7099	0.6896
RPR	0.7043	0.7279	0.7294	0.7040	0.6819
KZ	0.6462	0.7218	0.7232	0.7023	0.6752
PF	0.7394	0.7395	0.7399	0.6955	0.6733
Supervised	0.7412	0.7518	0.7763	0.7267	0.7132

set of two nodes is constructed. (b) The candidate links between communities are ignored.

One more interesting observation is, *CBLP* not only improves prediction accuracy, but also gives a hint on the number of communities. To confirm the fact, experiments are conducted by dividing the network into different number of communities and the AUROC scores are tabulated in Table.6.7 and the line graph showing the AUPR results of AA, KZ, PF and supervised LP measures in Fig.6.7.

A trend is observed that the scores are increasing when number of communities is increased till 3 and the performance is decreasing when number of clusters is 4 and further decreased when it is 5. In all cases for all prediction scores, the method gave best prediction score when the number of clusters is 3. As the nodes tend to be linked with other nodes within communities than between communities, when the number of communities is 4 or 5, the inter community edges are increasing. Therefore when inter-community common neighbours are ignored, the performance is decreasing.

Note that, the inter community edges are ignored by *CBLP*. It is counter intuitive that when a much reduced dataset is given for training and testing, there can be any improvement in accuracy. But we find that in Table.6.7, with respect to the measure *CN*, *CBLP* gives 73% accuracy for edge type Relation 1 with community discovery where as without community divison, gives 70% AUROC. Similar trend can be seen for AUPR in Fig.6.7. As explained for the synthetic dataset experimentation, clearly the TP's get enhanced with community information being given and FP's get reduced, which improves AUROC and AUPR.

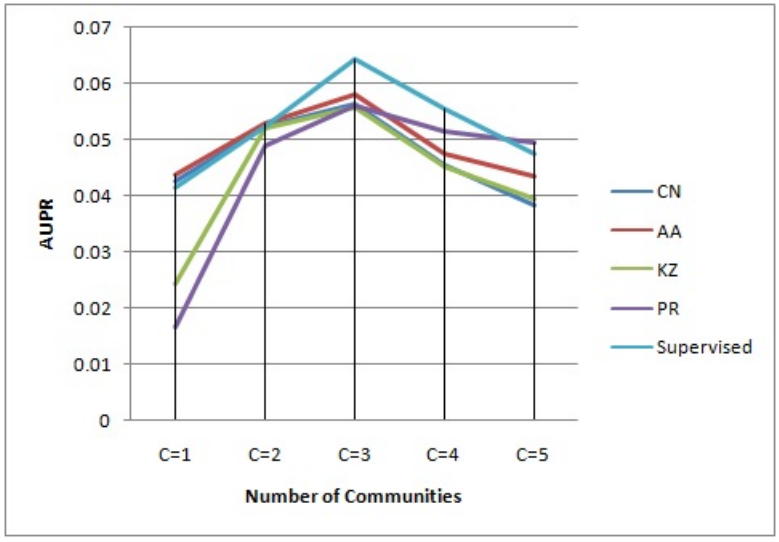


Figure 6.7.: AUPR of CN, AA, KZ, PR and Supervised LP measures for different no. of communities C: $Relation1$ of synthetic dataset

6.4.2. Results for Coauthorship network

We experimented the *CBLP* approach on the real-world coauthorship network of DBLP. To create a multi-relational network of DBLP, the publications in three areas are considered : Databases (DB), DataMining (DM) and Machine Learning (ML). These three areas are considered as three relations and the corresponding edge lists are computed. For example, there is an edge between two author nodes a_i and a_j under relation DB, if the two authors publish a paper together in any of the DB conferences. This construction of multi-relational dataset of DBLP is shown in Fig.6.8.

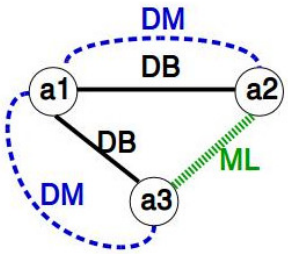


Figure 6.8.: Construction of DBLP Multi-relational Network

DBLP dataset does not contain the information of communities. Therefore, PMM algorithm is applied on the DBLP dataset by varying the number of communities, $C = 1,2,\ldots,20$. In each case, *CBLP* based link prediction algorithm is applied. The results obtained are given in

Table.6.8. It can be seen that the AUROC score for link prediction increases upto C=4 and then starts decreasing in the case of all LP measures as seen in Table. 6.8.

Table 6.8.: AUROC of LP measures for DBLP coauthorship network using *CBLP* approach for different number of communities on all types of edges

Communities→ LP↓	**C=1**	**C=2**	**C=3**	**C=4**	**C=5**	**C=6**	**C=7**	**C=8**	**C=9**	**C=10**	**C=15**	**C=20**
CN	0.6504	0.6414	0.6519	0.6678	0.6511	0.6439	0.6311	0.6029	0.5832	0.5613	0.5521	0.4913
JC	0.5942	0.5814	0.5900	0.6010	0.5900	0.5823	0.5801	0.5615	0.5411	0.5123	0.5000	0.4621
AA	0.6777	0.6623	0.6799	0.6982	0.6800	0.6713	0.6611	0.6214	0.6111	0.5923	0.4999	0.4912
PA	0.7415	0.7311	0.7492	0.7513	0.7311	0.7201	0.7000	0.6401	0.6400	0.6242	0.5930	0.5612
COP	0.8379	0.8214	0.8399	0.8411	0.8312	0.8211	0.8194	0.7802	0.8000	0.7311	0.7214	0.6823
TS	0.7913	0.7764	0.8011	0.7923	0.7811	0.7715	0.7698	0.7500	0.7413	0.7211	0.6478	0.5256
LS	0.8016	0.7821	0.7942	0.7999	0.7812	0.7732	0.7635	0.7467	0.7400	0.7219	0.6201	0.5521
TF	0.8125	0.8031	0.8112	0.8132	0.8046	0.7811	0.7823	0.7500	0.7421	0.7224	0.6311	0.5745
TCOP	0.8590	0.8100	0.8621	0.8713	0.8314	0.8298	0.8242	0.7864	0.8214	0.7431	0.7321	0.6841
Supervised	0.9281	0.9103	0.9300	0.9474	0.9324	0.9217	0.8499	0.8993	0.8342	0.8251	0.7867	0.6940

Therefore, we consider the number of communities in DBLP as 4 and apply *CBLP* approach to the DBLP network. The AUROC results obtained on DBLP network are given in Table.6.9, and AUPR results are shown in Fig.6.9, 6.10, 6.11.

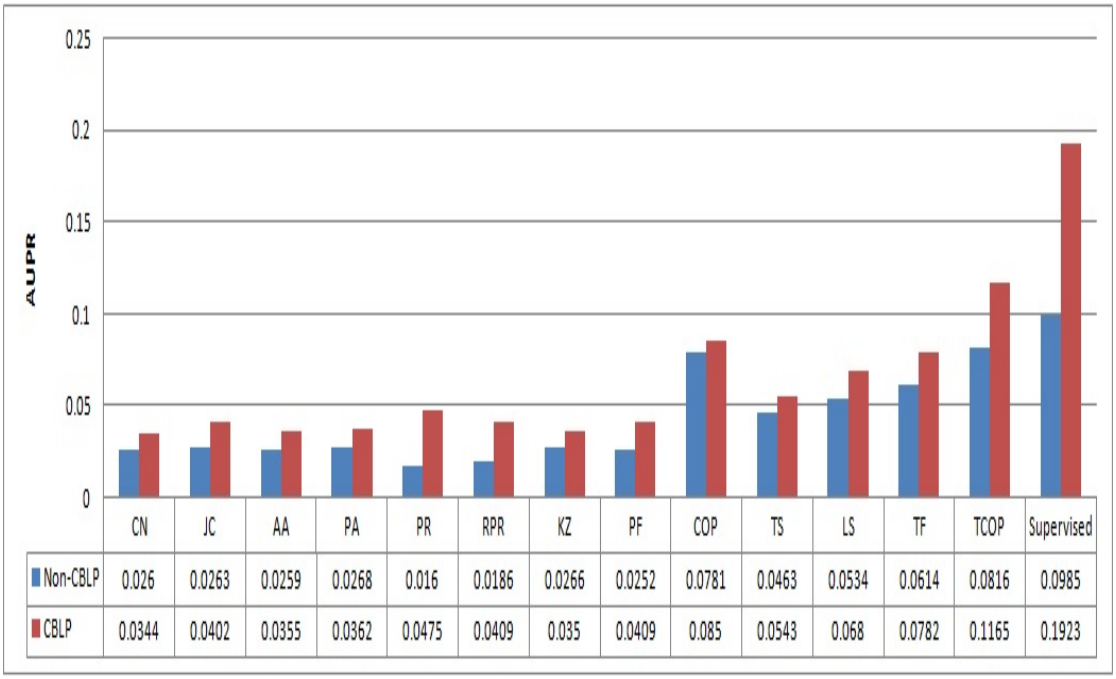


Figure 6.9.: AUPR results of *CBLP* Vs *Non-CBLP* of link type **DB** for **DBLP** network

Table 6.9.: AUROC of LP measures for three Relations of DBLP dataset using *CBLP* approach

Relation→ LP↓	Relation 1 (DM)		Relation 2 (DB)		Relation 3 (ML)	
	Non-*CBLP*	*CBLP*	Non-*CBLP*	*CBLP*	Non-*CBLP*	*CBLP*
CN	0.6642	0.6932	0.5472	0.5720	0.5831	0.5843
JC	0.6497	0.6522	0.4921	0.5296	0.5934	0.6482
AA	0.6953	0.7320	0.6392	0.6723	0.6823	0.7012
PA	0.7217	0.7230	0.6509	0.6592	0.6823	0.6923
PR	0.6250	0.6578	0.5492	0.5823	0.5723	0.6201
RPR	0.6316	0.6612	0.5500	0.5823	0.5923	0.6102
KZ	0.6926	0.7302	0.6439	0.6621	0.6810	0.6899
PF	0.6722	0.7011	0.6391	0.6502	0.6691	0.6932
COP	0.8142	0.8523	0.7493	0.7733	0.7623	0.8023
TS	0.7742	0.8321	0.6834	0.6932	0.6888	0.7021
LS	0.7954	0.8492	0.7034	0.7421	0.7453	0.7823
TF	0.8000	0.8621	0.7283	0.7512	0.7600	0.7821
TCOP	0.8365	0.8823	0.7810	0.8012	0.8000	0.8211
Supervised	0.8943	0.9270	0.8295	0.8634	0.8534	0.9232

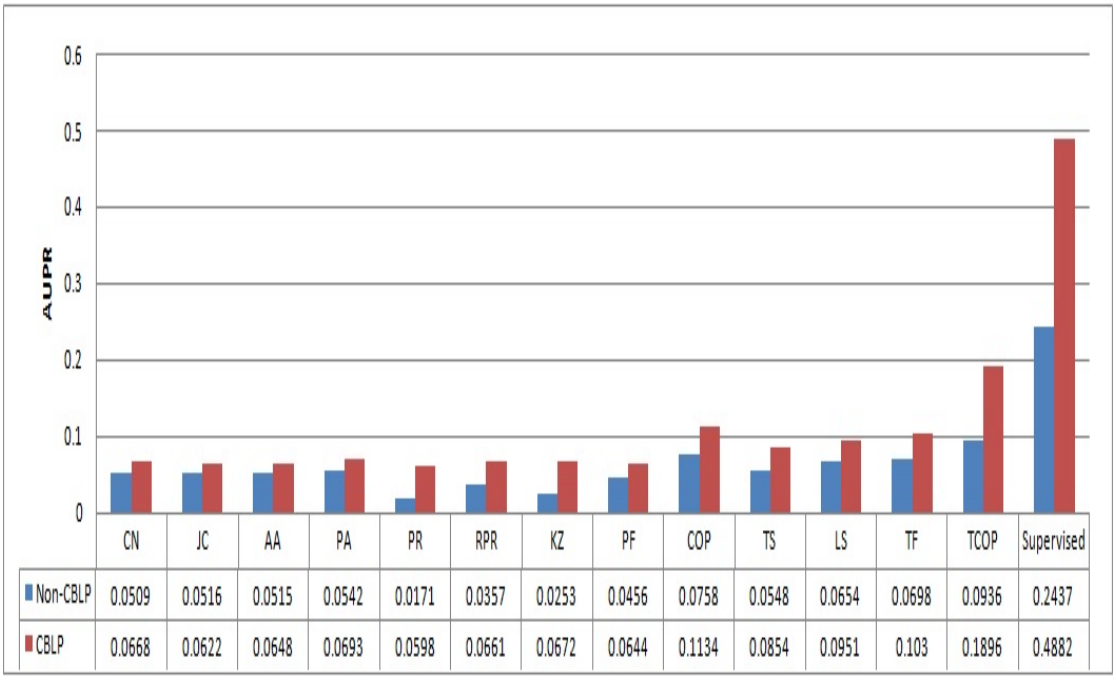
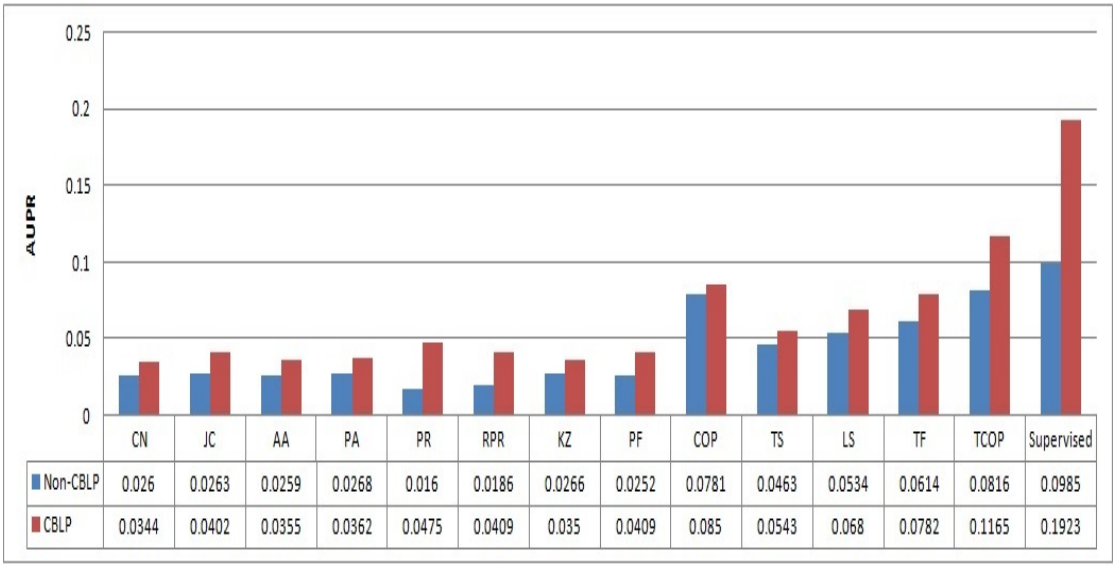


Figure 6.10.: AUPR results of *CBLP* Vs *Non-CBLP* of link type **DM** for **DBLP** network



	CN	JC	AA	PA	PR	RPR	KZ	PF	COP	TS	LS	TF	TCOP	Supervised
Non-CBLP	0.026	0.0263	0.0259	0.0268	0.016	0.0186	0.0266	0.0252	0.0781	0.0463	0.0534	0.0614	0.0816	0.0985
CBLP	0.0344	0.0402	0.0355	0.0362	0.0475	0.0409	0.035	0.0409	0.085	0.0543	0.068	0.0782	0.1165	0.1923

Figure 6.11.: AUPR results of *CBLP* Vs *Non-CBLP* of link type **ML** for **DBLP** network

There is significant improvement seen in the AUROC as well as AUPR scores for *CBLP* approach over non-*CBLP* approach for all the link types, *DM*, *DB* and *ML*. As expected, *TCOP* performed better for *CBLP* approach also. Supervised classification shows maximum improvement in the case of link type *ML*.

6.5. Conclusion

In this chapter, a community based approach for link prediction is proposed and evaluated. The community based approach divides the network into communities and the link prediction measures are applied on each community in parallel. The results from each of the communities are integrated and evaluated. Though our focus is on heterogeneous network mining, we limited our experimentation to multi-relational networks because the community discovery algorithms for pure heterogeneous networks are not available in the literature to the best of our knowledge. The community based approach improves accuracy of prediction and memory efficient. But the success of this approach depends on the availability of efficient community discovery algorithms as the time taken for community division is also a major issue. This approach is applicable only to well clustered networks.

7. Application of Link Prediction in Recommender Systems

The main objective of this chapter is to find the effectiveness of the proposed heterogeneous link prediction measures in a totally different application like link recommender systems. Many e-commerce websites provide a wide range of products to the users. The users commonly have different needs and tastes based on which they buy the products. Recommending the most appropriate products to the users make the buying process efficient and improves the user satisfaction. The enhanced user satisfaction keeps the user loyal to the website and improves the sales and thus profits of the retailers. E-commerce leaders like Amazon and Netflix use recommender systems to recommend products to the users.

Recommender systems recommend items to users. Items include products and services such as movies, music, books, web pages, news, jokes and restaurants. The recommendation process utilizes data including user demographics, product descriptions and the previous history of users on items like buying, rating, and watching. The information can be acquired explicitly by collecting ratings given by users on items or implicitly by monitoring user's behaviour such as songs listened in music websites, news/movies watched in news/movies websites, items bought in e-commerce websites or books read in book-listing websites in the past.

Recommender system is a typical application of heterogeneous link prediction. Graph-based recommendation algorithms compute recommendations on a bipartite graph where two types of nodes are present in the graph representing users and items. All the link prediction methods proposed in this thesis can be applied directly to the bipartite network in order to predict the possible recommendation links between the items and users [87, 88]. But the application is limited to the scope of only obtaining 'top-k' ranked items for a user. Recommender systems are more general and predict the actual ratings given by a user for an item. But LP systems will not be able to predict ratings. Hence, we compare our application only upto ranking aspect of the recommendation systems.

7.1. Background

Schafer et al [89] define the problem of recommender systems as follows:

Definition 7.1. Given a set of users $U = \{u_1, u_2, \ldots u_m\}$, and items $I = \{I_1, I_2, \ldots, I_n\}$ and the ratings $R = [r_{ij}]$ representing the ratings given by user u_i to item I_j, the task of recommender systems is to recommend a new item i_j to a user u_i which the user u_i has not bought.

For example, consider the matrix given in Fig.7.1, where rows correspond to the users and the columns denote movies. The matrix entry R_{ij} represents the rankings given by user u_i on movie m_j. The main task of the recommender systems is to predict the unrated entries in the rating matrix.

$$\mathbf{R} = \begin{array}{c} \begin{array}{cccc} \text{Harry Potter} & \text{Avengers} & \text{Jurassic Park} & \text{Terminator 5} \end{array} \\ \left[\begin{array}{cccc} 5 & & & \\ 4 & 4 & & \\ & 3 & 2 & \\ & & & 1 \\ & 2 & & \\ & & & 2 \end{array}\right] \end{array} \begin{array}{l} \text{User 1} \\ \text{User 2} \\ \text{User 2} \\ \text{User 3} \\ \text{User 4} \\ \text{User 5} \end{array}$$

Figure 7.1.: Rating matrix of movie dataset containing 4 movies rated by 6 users

7.1.1. Classical Approaches for RS

Recommender systems is seeing an explosive growth in the literature with many novel algorithms being proposed almost everyday. Our intention is not to compare our approach to any of the latest state-of-the-art algorithms but to apply the newly proposed LP measures like *COP*, *TCOP*, *Hetero*-TCOP etc on a completely new domain other than bibliographic networks. Hence we present some of the traditional approaches to recommender systems and compare our proposed LP measures with these few algorithms.

Content-based recommendation, Collaborative Filtering (CF) and Hybrid Approach are some of the popular approaches to solve the problem of recommender systems. Content-based recommendation uses item descriptions and constructs user profiles which contain the information about user preferences [90]. The recommendation of an item to the user is then based on the similarity between the item description and the user profile. This approach has the advantage of being able to recommend previously unrated items to users with unique interests and to provide

explanations for its recommendations [91]. Content-based recommendation systems require complete descriptions of items and detailed user profiles. This is one of the the main limitations of such systems.

Collaborative filtering (CF) techniques depend only on the user's past behaviour and provide personalized recommendation of items to users [92]. CF techniques take a rating matrix as input where the rows of the matrix correspond to users, the columns correspond to items and the cells correspond to the rating given by the user to the item [93]. CF methods are classified into *Latent factor* CF models [94] and *Neighbourhood-based* CF.

Latent factor models ratings by characterizing both items and users on some number of factors inferred from Matrix factorization methods. Data sparsity is a serious problem of CF methods. Dimensionality reduction techniques, such as Principal Component Analysis (PCA), Singular Value Decomposition (SVD) are some of the popular techniques that address the problem of sparsity. However, when certain users or items are discarded, useful information for recommendations related to them may get lost and recommendation quality may be degraded [95]. The limitation of collaborative filtering systems is the *cold start problem.* i.e, these methods can not recommend new items to the existing users as there is no past buying history to the item. At the same time, it is difficult to recommend items to a new user without knowing the user's interests in the form of ratings.

Neighbourhood based methods compute a set of nearest neighbours for users as well as items using similarity measures like Pearson coefficient, cosine distance and Manhattan distance. Neighbourhood based CF techniques are further classified into *user based* and *item based*. User based methods compute the similarity between the users based on their ratings on items [96]. These methods associate a set of nearest neighbours with each user and then predict the user's rating for unscored items using the ratings given by the neighbours on that item. Similarly, item neighbourhood depicts the number of users rating the same items. The item rating for a given user can then be predicted based upon the ratings given in their user neighbourhood and the item neighbourhood.

Hybrid approach uses both types of information, collaborative and content-based. Content-boosted CF algorithm [97], uses the item profile information to recommend the items to new users.

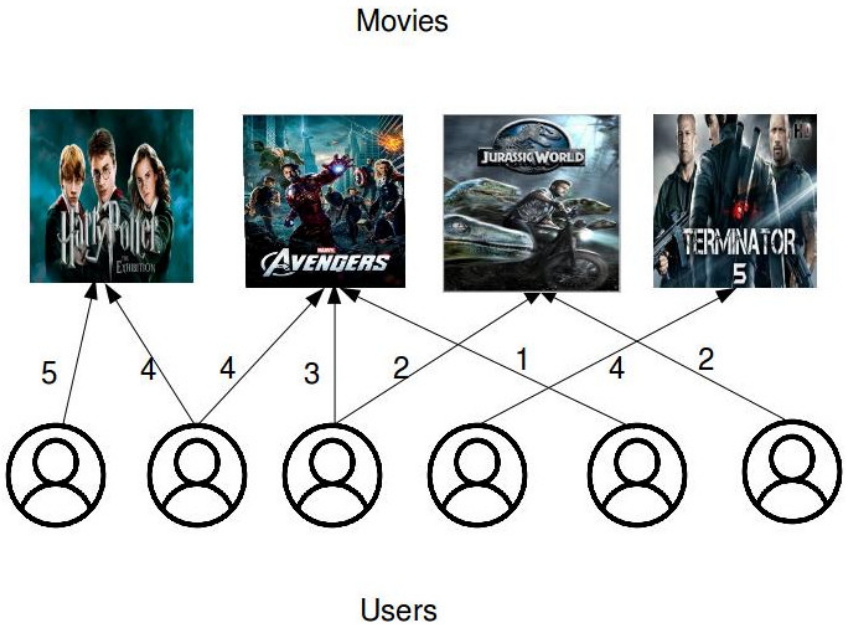


Figure 7.2.: Movie-User bipartite graph

7.2. Proposed Approach : Application of LP measures to Recommender Systems

The networks underlying recommendation systems can be modelled as bipartite graphs, $G = (U \cup I, E)$ where U is the set of user nodes, I is the set of item nodes and E represents the set of heterogeneous edges. The edge (u,i) between $u \in U$ and $i \in I$ exists if the user u buys the item i. The weights of the edge represent the rating given by u on i. The entries in the rating matrix can be treated as weights of the links between users and items. For example consider the graph representation of matrix in Fig.7.1, shown in Fig.7.2. One can see that there are four movie nodes corresponding to the four movie columns in the matrix and six user nodes, one for each user in matrix of Fig.7.1. There is an edge between a user node and movie node, if the user watched the movie and the weight on the edge denotes the rating that the user has given to the movie.

The standard link prediction methods can predict the recommendation link between a user and an item, but will not be able to predict the actual rating. Hence we adopt the ranking-oriented recommendation approach in which the recommendation is treated as a ranking task recommending top-k ranked items to a user [98]. We follow the unsupervised learning approach used in

the previous chapters to predict a link between an item and a user using the standard as well as proposed link prediction measures in a bipartite environment. Our main interest is to apply our proposed measures like $Hetero$-COP and $TCOP$ to this problem and obtain comparative performance both with other LP measures and some of the classical recommender systems.

The probabilistic measures COP and TCOP work on cliques of the graph and in the previous chapters, in the case of bipartite networks, H-cliques were used by suppressing the homogeneous links. In this application, an algorithm for extracting bipartite cliques (B-cliques) is being proposed. In bibliographic networks, author cliques are available in event logs which are utilized to extract the H-cliques containing the author nodes and conference nodes. In the case of recommender systems, user cliques and item cliques are not readily available in the event logs, but can be retrieved using simple sorting techniques. Since the number of users is huge in comparison to the set of items which is a much smaller set, the extraction of B-cliques can start from the item cliques. The proposed algorithm for extracting B-cliques is given in Algorithm 6.

Algorithm 6 Extraction of B-Cliques from user-item bipartite graph

Input: $G = (V,E)$ where $V = U \cup I$, U is set of user nodes and I is the set of item nodes and E represents weighted heterogeneous edges.

Output: $Bcliq$, set of maximal B-cliques of G.

Step 1: Extract the set of all item cliques, $Item\_Cliq$ as follows:
For each user u, select all the items to which u gives a rating to form I_u
Step 2: Extract the set of all user cliques, $User\_Cliq$ as follows:
For each item i, take all users who have rated that item i as user clique U_i.
Step 3:
for each item i in I **do**
  $J \leftarrow I$
  for each user v in U_i **do**
    $J \leftarrow J \cap I_v$
  end for
  // $J = \bigcap_{v \in U_i} I_v$
  $Bcliq_i = J \cup U_i$
  $Bcliq = \cup_i Bcliq_i$
end for
return $Bcliq$

The B-clique extraction algorithm first extracts all homogeneous cliques of items $Icliq$ and users $Ucliq$. A homogeneous user clique is formed with all the users who rate/buy the same item.

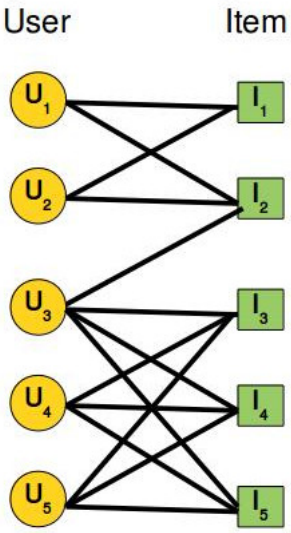


Figure 7.3.: User-Item bipartite graph

Similarly, a homogeneous clique of items is formed with all the items a user rate/buy. To extract a B-clique, first consider an item i. For each user v who have rated i, compute the common items rated by user v. The union of U_i along with all the common items rated by U_i forms a B-clique. The process of extracting B-cliques for toy example in Fig.7.3 is illustrated below:

U=$\{u_1, u_2, u_3, u_4, u_5\}$ I=$\{i_1, i_2, i_3, i_4, i_5\}$

Item cliques :

Item clique corresponding to user $u_1 = \{i_1, i_2\}$

Item clique corresponding to user $u_2 = \{i_1, i_2\}$

Item clique corresponding to user $u_3 = \{i_2, i_3, i_4, i_5\}$

Item clique corresponding to user $u_4 = \{1_3, i_4, i_5\}$

Item clique corresponding to user $u_5 = \{1_3, i_4, i_5\}$

User cliques :

User clique corresponding to item $i_1 = \{u_1, u_2\}$

User clique corresponding to item $i_2 = \{u_1, u_2, u_3\}$

User clique corresponding to item $i_3 = \{u_3, u_4, u_5\}$

User clique corresponding to item $i_4 = \{u_3, u_4, u_5\}$

User clique corresponding to item $i_5 = \{u_3, u_4, u_5\}$

The the extraction of B-cliques from the above homogeneous cliques is shown in Fig.7.4.

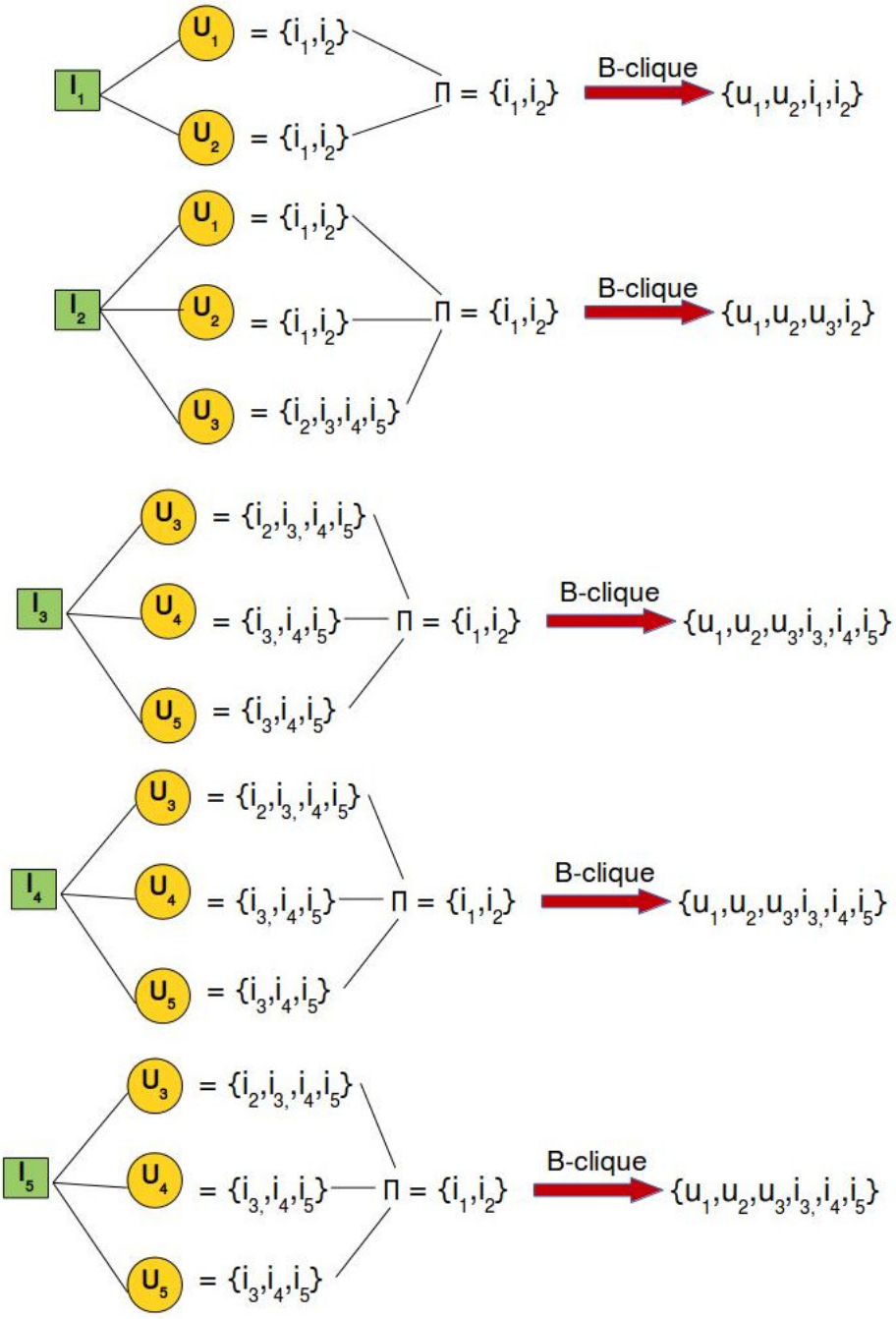


Figure 7.4.: Illustration of extracting *B*-cliques from User-Item event logs

After extracting the B-cliques from the user-item bipartite graph, the *Hetero*-COP measure can be computed using the procedure specified in section 4.5 and *Hetero*-TCOP scores can be computed using the Algorithm 4. The experimental evaluation of proposed approach is given in next section.

7.3. Experimental Evaluation

To see the applicability of the link prediction measures proposed in this thesis, they are used to recommend movies to users. The experimentation is carried out on a benchmark MovieLens recommender system whose details are given below.

7.3.1. Dataset

This data set used for experimental evaluation contains more than ten million ratings given by 71,567 users on 10,681 movies of the online movie recommender service MovieLens [99]. In the graph recommendation systems, movies are considered as nodes and items are users and rating given by users to the movies are considered as the weights on links. In MovieLens dataset, the train-test splits are given in [99]. The users who have rated atleast 20 movies have been chosen randomly to be included in this dataset. The benchmark data set [100] is given with 80% - 20% split with 80% given as training set and test set containing 20%. The training and test sets are formed by splitting the ratings data such that, for every user, 80% of his/her ratings is taken in training and the rest are taken in the test set. In this experimentation, 5 fold cross validation is used. All the 5 sets of training and test datasets are made available at [100]. The evaluation metrics used for recommender systems are given in the following section.

7.3.2. Evaluation Metrics

Evaluation metrics in recommender systems can be classified as

- Accuracy measures: Mean Absolute Error (MAE), Root of Mean Square Error (RMSE), Normalized Mean Average Error (NMAE).
- Set recommendation metrics : Precision, Recall and Area Under Receiver Operating Characteristic (AUROC), Area Under Precision-recall curve (AUPR)
- Rank recommendation metrics: Half-life, discounted cumulative gain and Rank-score

Most of the measures listed above, use rating to calculate the error and hence are not applicable in our context. We use AUROC, AUPR are used for evaluating performance.

Rank-score Rank-score metric measures the ability of a recommendation algorithm to produce a ranked list of recommended items. The recommender system method is efficient, if the ranking given by the method matches with the user's order of buying the items in the recommended list. Rank-score is defined as follows : For a user u, the list of items i recommended to u, that is predicted by the algorithm is captured by $rankscore_p$

$$\begin{aligned} rankscore_{max} &= \frac{1}{\sum_{j=1}^{|T|} 2^{\frac{j-1}{\alpha}}} \\ rankscore_p &= \frac{1}{\sum_{j\in|T|} 2^{\frac{ranl(j)-1}{\alpha}}} \\ Rankscore &= \frac{rankscore_p}{rankscore_{max}} \end{aligned} \tag{7.1}$$

where $rank(j)$ is the rank given by the recommender algorithm to item j. $|T|$ is the number of items of interest and α is ranking half-life, an exponential reduction factor.

7.3.3. Results

The performance of various link prediction measures are compared with the standard User-based and Item-based collaborative filtering (CF) methods. Pearson correlation coefficient is used to find the similar users in User-based CF and cosine similarity is used for finding item similarity in Item-based CF. The code given in [101] is used for implementation of User-based CF and Item-based CF algorithms. The results obtained for AUROC, AUPR and Rank-score for MovieLens dataset are given in Table.7.1.

First observation in this experimentation is that some of the link prediction measures like Katz, PropFlow could produce better recommendation compared to Item-based CF. The usage of temporal measures seem to help in improving the quality of recommendations. The time of formation of link or the time of rating given by a user to movie plays crucial role, as the user's preferences change over time. The temporal measures TS, LS, TF and TCOP assign more weight to the recent ratings. Therefore, temporal measures performed better than all the other measures including User-based CF and Item-based CF. TCOP outperformed all the other link prediction measures and the user-based and item-based collaborative filtering methods. The recommendation

Table 7.1.: Performance of LP measures for recommending movies to users in **MovieLens** Bipartite Network

LP measure	**Rank-score**	**AUROC**	**AUPR**
Non-temporal LP measures			
Hetero-CN	2.0501	0.5123	0.0092
Hetero-JC	1.3615	0.4956	0.0085
Hetero-AA	1.7819	0.5749	0.0153
Hetero-PA	2.6804	0.6832	0.0251
Hetero-KZ	3.2546	0.6635	0.0193
Hetero-PF	3.2990	0.6846	0.0286
Hetero-COP	3.6661	0.7231	0.0613
Temporal LP measures			
Hetero-TS	4.0015	0.7016	0.0365
Hetero-LS	5.2134	0.7340	0.0861
Hetero-TF	5.3684	0.7532	0.0960
Hetero-TCOP	**8.6304**	**0.8165**	**0.2351**
Classical Collaborative Filtering methods			
User-based CF	3.9942	0.6925	0.0415
Item-based CF	3.0274	0.7136	0.0491

performance is improved by 6% in terms of AUROC over TF and nearly 10% over CF methods. It can be observed from Table.7.1 that the AUPR score also have shown great improvement from 0.0960(of TF) to 0.2351. Similar trend is observed for the measure the evaluation measure Rank-score. All the temporal measures perform better than User-based CF and Item-based CF. TCOP is rated as highest by Rank-score.

7.4. Conclusion

In this chapter, the recommender systems problem is solved using link prediction approach. Though it is not fair to compare the graph based LP approach with the regular RS algorithms since LP algorithms predict only the link but not the rating on the link, we demonstrate the applicability of LP algorithms to recommender systems. The standard recommender system methods suffer from the problems of sparsity and scalability. As the link prediction measures are local, these methods reduce the problem of sparsity and are scalable. However, link prediction approach does not address the cold start problem.

Some insights gained during this experimentation are that by considering the temporal information, the prediction task enhances the performance of the recommendations. Temporal measures have proved to perform well in this scenario. If the method can somehow be extended to predict the rating as well, then it will be interesting to assess the performance of the temporal heterogeneous link prediction measures in recommendation systems.

8. Conclusions

8.1. Conclusion

In this thesis, link prediction problem is studied for heterogeneous social networks. Five major contributions have been made for link prediction. Some of the baseline link prediction measures available for homogeneous networks are extended for heterogeneous networks. The proposed measures are evaluated on four benchmark bibliographic datasets of Condmat, DBLP, HiePh-cite and HiePh-collab. The results meet the basic expectation of an improvement in accuracy when heterogeneous information is included.

As a second contribution, the significance of PGM's for link prediction task is explained. A probabilistic measure existing in the literature called Co-occurrence probabilistic measure (COP) [31] for homogeneous networks is extended to bipartite and heterogeneous social networks. *Hetero*-COP is shown to exhibit improved performance over neighbourhood, path and random-walk based measures. A new measure called Temporal Co-occurrence Probability (TCOP) is proposed for homogeneous networks, that extends computations on Markov Random Fields by effectively utilizing the temporal information available in the network.

It is interesting to find that all the measures improved when heterogeneous information is utilized and further improved with the usage of temporal information in the computation. In order to appreciate the improvement along temporal and heterogeneous dimensions, a summarized performance of all the measures with respect to these two aspects are given for DBLP bibliographic network in Table.8.1. It is to be noted that TimeScore, LinkScore, T_Flow and TCOP are the temporal extensions of CN, Katz, PropFlow and COP respectively, whose results are given under the temporal column. *PA* does not change for all types of networks. Similarly, the results for heterogeneous extensions Hetero-CN till Hetero-COP are given in the respective columns.

An overall trend is observed that the temporal measures significantly improve the prediction performance over non-temporal versions. Temporal measures in heterogeneous networks are

Table 8.1.: Performance of LP measures for DBLP network in various environments

Type of NW →	Non-temporal				Temporal			
	Homogeneous		Heterogeneous		Homogeneous		Heterogeneous	
LP ↓	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR
Neighbourhood-based (CN)	0.6504	0.0681	0.7913	0.1625	0.6811	0.0720	0.8290	0.1766
Neighbourhood-based (PA)	0.7415	0.1162	0.7423	0.1164	0.7415	0.1162	0.7423	0.1164
Path-based (KZ)	0.6436	0.0679	0.6843	0.0942	0.8016	0.1721	0.8376	0.2276
Random-walk based (PF)	0.6264	0.0695	0.6532	0.0793	0.8125	0.1785	0.8263	0.1791
Probabilistic (COP)	0.8379	0.2028	0.8439	0.2399	0.8590	0.2421	0.8934	0.3953

overall winners.

Among neighbourhood-based measures, for homogeneous networks, *PA* performed better with a 74% AUROC score and 0.11 AUPR score and for heterogeneous networks *CN* is proved to be better with an AUROC score of 83% AUROC and 0.17 AUPR. Among path and random walk based measures in temporal environment, *PF* has shown better performance for homogeneous and *KZ* proves to be better for heterogeneous networks. The probabilistic measure $TCOP$ performed better than all the other measures for all types of networks with an AUROC score of 89% and AUPR score of 0.39. These summarised results clearly indicate that the usage of temporal and heterogeneous information significantly improve prediction performance.

A scalable approach is proposed based on the network structure called Community based Link Prediction (*CBLP*) for multi-relational networks. *CBLP* computes communities of the network and computes prediction scores within each community in parallel. The results of implementation of *CBLP* on bench-mark datasets show that community information does significantly help in improving the performance of link prediction for multi-relational networks. But the limitation of this method is *CBLP* works for networks with well defined communities. The behaviour of these measures for networks with overlapping communities are still to be explored.

The proposed heterogeneous link prediction measures have been applied to a totally different application of link recommender systems. Some of the link prediction measures like Katz, PropFlow produce better recommendation compared to the standard collaboration filtering methods. The usage of temporal measures helped in improving the quality of recommendations. The time of buying/rating plays crucial role, as the users preferences change over time. More rigor-

ous experimentation is needed to compare the performance with the recent collaboration filtering techniques.

8.2. Future Scope

There can be many variations for link prediction problem that merit further investigation.

Link prediction is in general used for predicting probable link formation in the immediate next period of time. In supervised setting, one assumes certain amount of information being available for training say up to y years, and testing to be done on the data for the year $(y+1)$. This can be certainly termed as short-term link prediction. We propose a new problem, in which the model is trained on data available up to say y years and then to predict the potential link formation for $(y+t)^{th}$ year, t>1 (after a gap of t years) may be named as long-term link prediction.

Long-term link prediction problem may not be always meaningful, for example, in weather prediction scenarios. But in collaboration networks and disease networks, where it is natural to expect a gestation period for the formation of a new link, it is very much meaningful to assess if the existing algorithms are capable of long-term prediction. A preliminary work that has been done for predicting links in long term, is reported in [102]. This problem needs further investigation.

In all our contributions we made in this thesis, we have treated all edge types with equal preference. But in some of the applications, one edge type may be more dominating than others and more weightage may need to be given for such edge types. This needs to be further explored.

Social networks are dynamic in nature. The link prediction in dynamic social networks, where new nodes get added and deleted is still to be addressed for the proposed measures. On-line algorithms for link prediction in heterogeneous networks can be an interesting future direction. The existing link prediction measures cannot predict the year of formation of link as well as strength of the future link. These problems are worth further investigation.

List of Publications

1. T. Jaya Lakshmi and S. Durga Bhavani. Link Prediction Measures in Various Types of Information Networks : A Review. **IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM**, Barcelona, Spain, August, 2018. (DBLP) (Presented.)
2. T. Jaya Lakshmi and S. Durga Bhavani. Link Prediction in Temporal Heterogeneous Networks. **In Proceedings of the 12th Pacific Asia Workshop, (PAKDD Workshop)**, Jeju Island, South Korea. pp 83-98, May 2017. (DBLP)
3. T. Jaya Lakshmi and S. Durga Bhavani, Temporal probabilistic measure for link prediction in collaborative networks". **J. Applied Intelligence**, Volume 47, Issue 1, pp 83-95, July 2017. (DBLP, SCI)
4. T. Jaya Lakshmi and S. Durga Bhavani. Enhancement to community-based multi-relational link prediction using co-occurrence probability feature. **In Proceedings of the Second ACM IKDD Conference on Data Sciences**, pp 86-91, March 2015. (DBLP)
5. T. Jaya Lakshmi and S. Durga Bhavani. Heterogenous link prediction based on multi-relational community information. In **Sixth International Conference on Communication Systems and Networks**, COMSNETS, pp 1-4, January 2014. (DBLP)

A. Illustration of Junction Tree Inference

The junction tree algorithm is a method used to infer joint probability between two nodes in a graph. This algorithm performs belief propagation on a modified graph of original grapg, called a Junction Tree.

For performing Junction Tree inference, first a clique graph is constructed for the given graph. Clique graph is a graph with maximal cliques of G as nodes. Two cliques can overlap, and the overlapped part is called their separator. There exists an edge between two clique nodes if there are common nodes between them. The weight of each edge in the clique graph is the size of their seperator set. Junction tree is a Maximum Spanning Tree of the clique graph.

Consider a snapshot extracted from DBLP coauthorship network in Fig.A.1 and the corresponding clique graph in Fig.A.2. In this example, the clique graph itself is tree. Therefore, junction tree of Fig.A.1 is same as clique graph.

We can see in Fig A.2 that there are 3 cliques and 2 separators.
Cliques :
$C_1 = \{6246, 6587, 37227\}$
$C_2 = \{195, 6246\}$ and
$C_3 = \{6587, 10497, 37227\}$
Separators :
$S(C_1, C_2) = \{6246\}$
$S(C_1, C_3) = \{6587, 37227\}$

There is one clique potential table corresponding to each clique. These clique potential tables are derived from NDI extracted from the event logs. In the example, the initial clique potentials of C_1,C_2,C_3 denoted by ϕ_{C_1}, ϕ_{C_2} *and* ϕ_{C_3} are given in Tables A.1, A.2 and A.3.

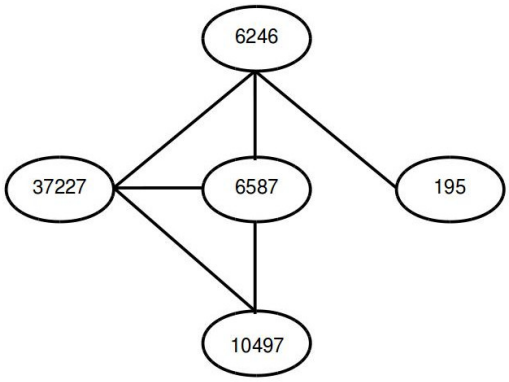


Figure A.1.: A snapshot of DBLP coauthorship network

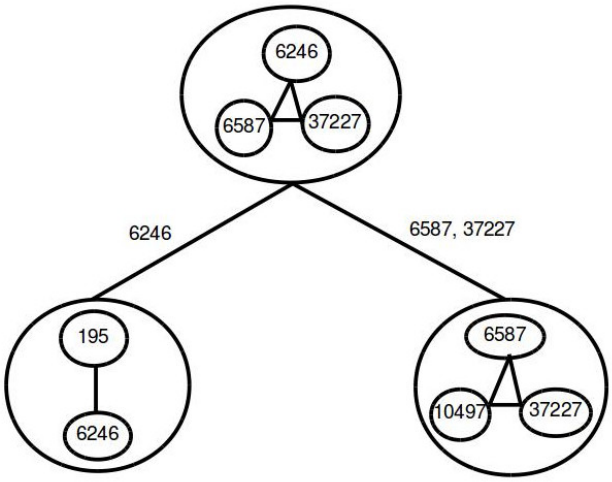


Figure A.2.: Junction tree of Fig.A.1

Table A.1.: ϕ_{C_1}

6246	6587	37227	$\phi_c(F)$
0	0	0	0
0	0	1	0.027778
0	1	0	0.027778
0	1	1	0.55556
1	0	0	0.19444
1	0	1	0.05556
1	1	0	0.11111
1	1	1	0.02778

Table A.2.: ϕ_{C_2}

195	6246	$\phi_c(F)$
0	0	0.6111
0	1	0.1944
1	0	0
1	1	0.1944

Table A.3.: ϕ_{C_3}

6587	37227	10497	$\phi_c(F)$
0	0	0	0.19444
0	0	1	0
0	1	0	0.05556
0	1	1	0.02778
1	0	0	0.11111
1	0	1	0.02778
1	1	0	0.55556
1	1	1	0.02778

The belief propagation in junction tree has two steps: Upward belief propagation and downward belief propagation.

Belief Propagation in Upward Direction:

– For each leaf in the junction tree, send a message to its parent. The message is the marginal of its table, summing out any variable not in the separator.

$$for\ x \in S(C,C'),\ \phi_{S(C,C')}(x) = \sum_{x \in C', x \notin S} P(x) \tag{A.1}$$

That means, every leaf clique node C' sends the potential table computed by equation A.1,

to its parent clique, separated by separator $S(C,C')$.

In the above example, Message from $C_2 \rightarrow C_1$: Marginalize ϕ_{C_2} to $S(C_1,C_2)$. In the same way, message from $C_3 \rightarrow C_1$ can be obtained by marginalizing ϕ_{C_3} to $S(C_1,C_3)$, i.e,6587, 37227, as shown in Table. A.4.

Table A.4.: Upward belief propagation

Message from $C_2 \rightarrow C_1$

6246	$\phi_c(F)$
0	0.6111
1	0.3888

Message from $C_3 \rightarrow C_1$

6587	37227	$\phi_c(F)$
0	0	0.19440
0	1	0.08334
1	0	0.13889
1	1	0.58334

– When a parent Clique(say C) receives a message from a child (say C'), it multiplies its table by the message table to obtain its new table and thus updates for all child nodes C'.

$$\phi_C(x) = \prod_{C'} \phi_C(x)\phi_{S(C,C')}(x) \tag{A.2}$$

Continuing the same example as above, Update C_1 by the message from its child C_2 to get table A.5 and Update C_1 by the message from its child C_3 to get table A.6.

Table A.5.: Updated C_1 after C_1 receives message from C_2

6246	6587	37227	$\phi_c(F)$
0	0	0	0 * 0.6111 = 0
0	0	1	0.027778 * 0.6111 = 0.01697
0	1	0	0.027778 * 0.6111 = 0.01697
0	1	1	0.55556 * 0.6111 = 0.33951
1	0	0	0.19444 * 0.3888 = 0.07559
1	0	1	0.05556 * 0.3888 = 0.0216
1	1	0	0.11111 * 0.3888 = 0.0432
1	1	1	0.02778 * 0.3888 = 0.0108

Table A.6.: Updated C_1 after C_1 receives message from C_3

6246	6587	37227	$\phi_c(F)$
0	0	0	0 * 0.1944 = 0
0	0	1	0.01697 * 0.08334 = 0.00141
0	1	0	0.01697 * 0.13889 = 0.00236
0	1	1	0.33951 * 0.58334 = 0.19805
1	0	0	0.07559 * 0.19440 = 0.01469
1	0	1	0.02160 * 0.08334 = 0.00180
1	1	0	0.04320 * 0.13889 = 0.00600
1	1	1	0.01080 * 0.58334 = 0.00630

– When a parent receives messages from all its children, it repeats the process. This process continues until the root receives messages from all its children. In the above example, the tree is of height 1. So, therefore, the process stops in one iteration. The final potential of clique C_1 is shown in Table A.7

Table A.7.: Final Potential table of clique C_1 after upward pass

6246	6587	37227	$\phi_c(F)$
0	0	0	0
0	0	1	0.00141
0	1	0	0.00236
0	1	1	0.19805
1	0	0	0.01469
1	0	1	0.00180
1	1	0	0.00600
1	1	1	0.00630

After updations to clique potentials are carried out in upward pass, the downward pass is initiated.

Belief Propagation in Downward Direction: This step reverses upward pass, starting at the root.

- The root(say C) sends a message to each of its children. More specifically, the root **divides** its **current table** by the message received from the child through the separator(say $S(C,C')$), **marginalizes** the resulting table to the **separator**, and sends the result to the child.

$$\phi_(x) = \sum_{x \in C, x \notin S(C,C')} \frac{\phi_{C'}(x)}{\phi_{S(C,C')}(x)} \tag{A.3}$$

  In the example, the message from clique $C_1 \rightarrow C_2$ is computed, first by dividing C_1 by $S(C_1,C_2)$ as shown in table A.8 and then marginalizing it to $S(C_1,C_2)$ as shown in table A.9.

Table A.8.: $C_1/S(C_1,C_2)$

6246	6587	37227	$\phi_c(F)$
0	0	0	0 / 0.6111= 0
0	0	1	0.00141 / 0.6111 = 0.00231
0	1	0	0.00236 / 0.6111 = 0.00386
0	1	1	0.19805 / 0.6111 = 0.32408
1	0	0	0.01469 / 0.3888 = 0.03778
1	0	1	0.00181 / 0.3888 = 0.00463
1	1	0	0.00600 / 0.3888 = 0.0154
1	1	1	0.00630 / 0.3888 = 0.0162

Table A.9.: Message from $C_1 \rightarrow C_2$

6246	$\phi_c(F)$
0	0.33025
1	0.07401

In the same way, the message from clique $C_1 \rightarrow C_3$ is computed, first by dividing C_1 by

$S(C_1, C_3)$ as shown in table A.10 and then marginalizing it to $S(C_1, C_3)$ as shown in table A.11.

Table A.10.: $C_1/S(C_1, C_3)$

6246	6587	37227	$\phi_c(F)$
0	0	0	0.00000 / 0.19440 = 0
0	0	1	0.04100 / 0.08334 = 0.01690
0	1	0	0.00390 / 0.13889 = 0.01670
0	1	1	0.00142 / 0.58334 = 0.33950
1	0	0	0.21690 / 0.19440 = 0.07556
1	0	1	0.00960 / 0.08334 = 0.02160
1	1	0	0.00770 / 0.13889 = 0.04320
1	1	1	0.00164 / 0.58334 = 0.01080

Table A.11.: Message from $C_1 \rightarrow C_3$

6587	37227	$\phi_c(F)$
0	0	0.7556
0	1	0.0385
1	0	0.0599
1	1	0.3503

– Each child(C') **multiplies** its table by its parent's(C) table and repeats the process (acts as a root) until leaves are reached.

$$\phi_{C'}(x) = \prod \phi_C(x)\phi_S(x) \tag{A.4}$$

Thus, final potential table of clique is obtained by multiplying the table of C_2 by message obtained by its parent C_1 through its separator $S(C_1, C_2)$, which is shown in table . Similar is the case for getting final potential of clique C_3.

Table A.12.: Final potential table of clique C_2

195	6246	$\phi_c(F)$
0	0	0.6111 * 0.33025 = 0.2018
0	1	0.1944 * 0.07401 = 0.0144
1	0	0.0000 * 0.33025 = 0.0000
1	1	0.1944 * 0.07401 = 0.0144

Table A.13.: Final potential table of clique C_3

6587	37227	10497	$\phi_c(F)$
0	0	0	0.19444 * 0.07556 = 0.0147
0	0	1	0.00000 * 0.07556 = 0.00000
0	1	0	0.05556 * 0.03850 = 0.00214
0	1	1	0.02778 * 0.03810 = 0.00106
1	0	0	0.11111 * 0.05990 = 0.00665
1	0	1	0.02778 * 0.05990 = 0.00166
1	1	0	0.55556 * 0.35030 = 0.19460
1	1	1	0.02778 * 0.35030 = 0.009731

After completing the upward and downward belied propagation, the potentials will be equal to Marginals. To infer the joint probability of two variables, we will pick those two variables from any cliques, and multiply their potentials.

For instance, to compute the joint probability of nodes 195 and 6587, first pick the clique containing 195, i.e, C_2 and marginalize the two entries corresponding to the value 1 of node 195, which is 0.0144, and then pick the clique containing node 6587, i.e, C_3 and marginalize the entries corresponding to value 1 for node 6587, which is 0.21264. The joint probability is simply the product of 0.0144 * 0.21264 = 0.00306.

Bibliography

[1] "Dblp publication statistics," May 2018. https://dblp.uni-trier.de/.

[2] "Facebook q1 2018 results," Apr 2018. https://investor.fb.com/investor-events/event-details/2018/Facebook-Q1-2018-Earnings/default.aspx.

[3] T. Falkowski, J. Bartelheimer, and M. Spiliopoulou, "Mining and visualizing the evolution of subgroups in social networks," in *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, WI '06, pp. 52–58, IEEE Computer Society, 2006.

[4] D. Chakrabarti and C. Faloutsos, "Graph mining: Laws, generators, and algorithms," *ACM Comput. Surv.*, vol. 38, jun 2006.

[5] J. Ferlez, C. Faloutsos, J. Leskovec, D. Mladenic, and M. Grobelnik, "Monitoring network evolution using mdl," in *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, pp. 1328–1330, IEEE, 2008.

[6] M. Toyoda and M. Kitsuregawa, "Extracting evolution of web communities from a series of web archives," in *Proceedings of the fourteenth ACM conference on Hypertext and hypermedia*, pp. 28–37, ACM, 2003.

[7] A. Gohr, A. Hinneburg, R. Schult, and M. Spiliopoulou, "Topic evolution in a stream of documents," in *Proceedings of the 2009 SIAM International Conference on Data Mining*, pp. 859–870, SIAM, 2009.

[8] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *Journal of The American Society For Information Science and Technology*, vol. 58, no. 7, pp. 1019–1031, 2007.

[9] L. A. N. Amaral, "A truer measure of our ignorance," *Proceedings of the National Academy of Sciences*, vol. 105, no. 19, pp. 6795–6796, 2008.

[10] M. P. H. Stumpf, T. Thorne, E. de Silva, R. Stewart, H. J. An, M. Lappe, and C. Wiuf,

"Estimating the size of the human interactome," *Proceedings of the National Academy of Sciences*, vol. 105, no. 19, pp. 6959–6964, 2008.

[11] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen, "Collaborative filtering recommender systems," in *The adaptive web*, pp. 291–324, Springer, 2007.

[12] G. Berlusconi, F. Calderoni, N. Parolini, M. Verani, and C. Piccardi, "Link prediction in criminal networks: A tool for criminal intelligence analysis," *PLOS ONE*, no. 4, pp. 1–21, 2016.

[13] E. Gündoğan and B. Kaya, "A recommendation method based on link prediction in drug-disease bipartite network," in *Advanced Information and Communication Technologies (AICT), 2017 2nd International Conference on*, pp. 125–128, IEEE, 2017.

[14] G. Crichton, Y. Guo, S. Pyysalo, and A. Korhonen, "Neural networks for link prediction in realistic biomedical graphs: a multi-dimensional evaluation of graph embedding-based approaches," *BMC bioinformatics*, vol. 19, no. 1, p. 176, 2018.

[15] Z. Zhang, W. Ma, Z. Zhang, and H. Zhou, "A node pair entropy based similarity method for link prediction in transportation networks," in *International Conference on Electrical and Information Technologies for Rail Transportation*, pp. 817–825, Springer, 2017.

[16] C. C. Aggarwal, ed., *Social Network Data Analytics*. Springer, 2011.

[17] M. A. Hasan, V. Chaoji, S. Salem, and M. Zaki, "Link prediction using supervised learning," in *Proceedings of SDM 06 workshop on Link Analysis, Counterterrorism and Security*, 2006.

[18] R. N. Lichtenwalter, J. T. Lussier, and N. V. Chawla, "New perspectives and methods in link prediction," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD'10, pp. 243–252, ACM, 2010.

[19] A. Anderson, D. Huttenlocher, J. Kleinberg, and J. Leskovec, "Effects of user similarity in social media," in *Proceedings of the fifth ACM international conference on Web search and data mining*, pp. 703–712, ACM, 2012.

[20] P. Bhattacharyya, A. Garg, and S. F. Wu, "Analysis of user keyword similarity in online social networks," *Social network analysis and mining*, vol. 1, no. 3, pp. 143–158, 2011.

[21] C. G. Akcora, B. Carminati, and E. Ferrari, "User similarities on social networks," *Social Network Analysis and Mining*, vol. 3, no. 3, pp. 475–495, 2013.

[22] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. New York, NY, USA: McGraw-Hill, Inc., 1986.

[23] L. Hamers, Y. Hemeryck, G. Herweyers, M. Janssen, H. Keters, R. Rousseau, and A. Vanhoutte, "Similarity measures in scientometric research: The jaccard index versus salton's cosine formula," *Information Processing Management*, vol. 25, no. 3, pp. 315 – 318, 1989.

[24] T. Sørensen, *A Method of Establishing Groups of Equal Amplitude in Plant Sociology Based on Similarity of Species Content and Its Application to Analyses of the Vegetation on Danish Commons*. Biologiske skrifter, I kommission hos E. Munksgaard, 1948.

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

[26] E. A. Leicht, P. Holme, and M. E. J. Newman, "Vertex similarity in networks," *Phys. Rev. E*, vol. 73, p. 026120, Feb 2006.

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

[28] L. A. Adamic and E. Adar, "Friends and neighbors on the web," *Social Networks*, vol. 25, pp. 211–230, 2001.

[29] Q. Ou, Y.-D. Jin, T. Zhou, B.-H. Wang, and B.-Q. Yin, "Power-law strength-degree correlation from resource-allocation dynamics on weighted networks," *Phys. Rev. E*, vol. 75, p. 021102, Feb 2007.

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

[31] C. Wang, V. Satuluri, and S. Parthasarathy, "Local probabilistic models for link prediction," in *Proceedings of Seventh IEEE International Conference on Data Mining*, ICDM '07, pp. 322–331, IEEE Computer Society, 2007.

[32] C. Wang, J. Han, Y. Jia, J. Tang, D. Zhang, Y. Yu, and J. Guo, "Mining advisor-advisee relationships from research publication networks," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pp. 203–212, ACM, 2010.

[33] H. Kashima, T. Kato, Y. Yamanishi, M. Sugiyama, and K. Tsuda, "Link propagation: A fast semi-supervised learning algorithm for link prediction," in *Proceedings of the 2009 SIAM International Conference on Data Mining*, pp. 1099–1110, Philadelphia, PA, USA, May 2009.

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

[35] J. Kunegis and A. Lommatzsch, "Learning spectral graph transformations for link prediction," in *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 561–568, ACM, 2009.

[36] X. Li and H. Chen, "Recommendation as link prediction in bipartite graphs: A graph kernel-based machine learning approach," *Decision Support Systems*, vol. 54, no. 2, pp. 880–890, 2013.

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

[38] T. Tylenda, R. Angelova, and S. Bedathur, "Towards time-aware link prediction in evolving social networks," in *Proceedings of the 3rd Workshop on Social Network Mining and Analysis*, SNA-KDD '09, pp. 1–10, ACM, 2009.

[39] P. R. da Silva Soares and R. B. C. Prudêncio, "Time series based link prediction," in *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7, IEEE, 2012.

[40] Y. Sun, J. Han, C. C. Aggarwal, and N. V. Chawla, "When will it happen?: relationship prediction in heterogeneous information networks," in *Proceedings of the fifth ACM international conference on Web search and data mining*, pp. 663–672, ACM, 2012.

[41] Y. Yang, N. V. Chawla, Y. Sun, and J. Han, "Predicting links in multi-relational and heterogeneous networks," in *12th IEEE International Conference on Data Mining, ICDM 2012, Brussels, Belgium, December 10-13, 2012*, pp. 755–764, 2012.

[42] C. C. Aggarwal, Y. Xie, and P. S. Yu, "A framework for dynamic link prediction in heterogeneous networks," *Statistical Analysis and Data Mining*, vol. 7, no. 1, pp. 14–33, 2014.

[43] L. Munasinghe and R. Ichise, "Time aware index for link prediction in social networks.," in *DaWaK*, vol. 6862 of *Lecture Notes in Computer Science*, pp. 342–353, Springer, 2011.

[44] P. Choudhary, N. Mishra, S. Sharma, and R. Patel, "Link score: A novel method for time aware link prediction in social network," *ICDMW*, 2013.

[45] L. Munasinghe, "Time-aware methods for link prediction in social networks," *PhD Thesis, The Graduate University for Advanced Studies*, 2013.

[46] D. A. Davis, R. Lichtenwalter, and N. V. Chawla, "Supervised methods for multi-relational link prediction," *Social Network Analysis and Mining*, vol. 3, no. 2, pp. 127–141, 2013.

[47] R. N. Lichtenwalter and N. V. Chawla, "Vertex collocation profiles: theory, computation, and results," *SpringerPlus*, vol. 3, no. 1, pp. 1–27, 2014.

[48] N. Benchettara, R. Kanawati, and C. Rouveirol, "Supervised machine learning applied to link prediction in bipartite social networks," in *Advances in Social Networks Analysis and Mining (ASONAM), 2010 International Conference on*, pp. 326–330, IEEE, 2010.

[49] V. Leroy, B. B. Cambazoglu, and F. Bonchi, "Cold start link prediction," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 393–402, ACM, 2010.

[50] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, "Pathsim: Meta path-based top-k similarity search in heterogeneous information networks," *Proceedings of the VLDB Endowment*, vol. 4, no. 11, pp. 992–1003, 2011.

[51] Y. Sun, R. Barber, M. Gupta, C. C. Aggarwal, and J. Han, "Co-author relationship prediction in heterogeneous bibliographic networks," in *Proceedings of the 2011 International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '11, pp. 121–128, IEEE Computer Society, 2011.

[52] G. Fu, Y. Ding, A. Seal, B. Chen, Y. Sun, and E. Bolton, "Predicting drug target interactions using meta-path-based semantic network analysis," *BMC bioinformatics*, vol. 17, no. 1, p. 1, 2016.

[53] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Computer networks and ISDN systems*, vol. 30, no. 1-7, pp. 107–117, 1998.

[54] J. O'Madadhain, J. Hutchins, and P. Smyth, "Prediction and ranking algorithms for event-based network data," *SIGKDD Explorations Newsletter*, vol. 7, no. 2, pp. 23–30, 2005.

[55] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques*. Elsevier, 2011.

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

[57] G. Davis, Jesse and Mark, "The relationship between precision-recall and roc curves," in *Proceedings of the 23rd international conference on Machine learning*, ICML '06, pp. 233–240, ACM, 2006.

[58] K. Boyd, K. H. Eng, and C. D. Page, "Area under the precision-recall curve: Point estimates and confidence intervals," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 451–466, Springer, 2013.

[59] N. Chawla, "Data mining for imbalanced datasets: An overview," pp. 853–867, 2005.

[60] R. N. Lichtenwalter and N. V. Chawla, "Lpmade: Link prediction made easy," *Journal of Machine Learning Research.*, vol. 12, pp. 2489–2492, 2011.

[61] L. Tang, X. Wang, and H. Liu, "Community detection via heterogeneous interaction analysis," *Data Mining and Knowledge Discovery*, vol. 25, no. 1, pp. 1–33, 2012.

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

[63] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[64] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[65] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: An update," *SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[66] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009.

[67] M. J. Druzdzel, "Some properties of joint probability distributions," in *Proceedings of the Tenth International Conference on Uncertainty in Artificial Intelligence*, UAI'94, pp. 187–194, Morgan Kaufmann Publishers Inc., 1994.

[68] D. Pavlov, H. Mannila, and P. Smyth, "Probabilistic models for query approximation with large sparse binary data sets," in *UAI-2000*, pp. 465–472, Morgan Kaufmann Publishers, 2000.

[69] T. Calders and B. Goethals, "Mining all non-derivable frequent itemsets," in *Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery*, PKDD '02, pp. 74–85, Springer-Verlag, 2002.

[70] F. Jelinek, "Continuous speech recognition," *SIGART Bulletin*, no. 61, pp. 33–34, 1977.

[71] J. M. Mooij, "libDAI: A free and open source C++ library for discrete approximate inference in graphical models," *Journal of Machine Learning Research*, vol. 11, pp. 2169–2173, 2010.

[72] D. J. S. S. L. Lauritzen, "Local computations with probabilities on graphical structures and their application to expert systems," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 50, no. 2, pp. 157–224, 1988.

[73] M. E. Newman, "Modularity and community structure in networks," *Procedings of National Academy of Sciences, USA*, vol. 103, no. 23, pp. 8577–8582, 2006.

[74] G. W. Flake, S. Lawrence, and C. L. Giles, "Efficient identification of web communities," in *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 150–160, ACM, 2000.

[75] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proceedings of the national academy of sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[76] G. Su, A. Kuchinsky, J. H. Morris, D. J. States, and F. Meng, "Glay: community structure analysis of biological networks," *Bioinformatics*, vol. 26, no. 24, pp. 3135–3137, 2010.

[77] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Physics Review E*, vol. 74, p. 036104, 2006.

[78] M. E. Newman, "Fast algorithm for detecting community structure in networks," *Physical review E*, vol. 69, no. 6, p. 066133, 2004.

[79] M. E. Newman, "Analysis of weighted networks," *Physical review E*, vol. 70, no. 5, p. 056131, 2004.

[80] A. Clauset, M. E. Newman, and C. Moore, "Finding community structure in very large networks," *Physical review E*, vol. 70, no. 6, p. 066111, 2004.

[81] B. W. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs," *Bell System Technical Journal*, vol. 49, no. 2, pp. 291–307, 1970.

[82] S. White and P. Smyth, "A spectral clustering approach to finding communities in graphs," in *Proceedings of the 2005 SIAM international conference on data mining*, pp. 274–285, SIAM, 2005.

[83] L. Tang, H. Liu, J. Zhang, and Z. Nazeri, "Community evolution in dynamic multi-mode networks," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 677–685, ACM, 2008.

[84] N. Nguyen and R. Caruana, "Consensus clusterings," in *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pp. 607–612, IEEE, 2007.

[85] J. C. Valverde-Rebaza and A. de Andrade Lopes, "Link prediction in complex networks based on cluster information," in *Proceedings of the 21st Brazilian Conference on Advances in Artificial Intelligence*, SBIA'12, pp. 92–101, Springer-Verlag, 2012.

[86] D. Davis, R. Lichtenwalter, and N. V. Chawla, "Multi-relational link prediction in heterogeneous information networks," pp. 281–288, July 2011.

[87] J. Li, L. Zhang, F. Meng, and F. Li, "Recommendation algorithm based on link prediction and domain knowledge in retail transactions," *Procedia Computer Science*, vol. 31, pp. 875 – 881, 2014.

[88] X. Li and H. Chen, "Recommendation as link prediction in bipartite graphs: A graph kernel-based machine learning approach," *Decision Support Systems*, vol. 54, no. 2, pp. 880 – 890, 2013.

[89] J. B. Schafer, J. A. Konstan, and J. Riedl, "E-commerce recommendation applications," *Data Mining and Knowledge Discovery*, vol. 5, no. 1, pp. 115–153, 2001.

[90] M. J. Pazzani and D. Billsus, "The adaptive web," ch. Content-based Recommendation Systems, pp. 325–341, Springer-Verlag, 2007.

[91] R. J. Mooney and L. Roy, "Content-based book recommending using learning for text categorization," in *Proceedings of the Fifth ACM Conference on Digital Libraries*, DL '00, pp. 195–204, ACM, 2000.

[92] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: item-to-item collaborative filtering," *IEEE Internet Computing*, vol. 7, no. 1, pp. 76–80, 2003.

[93] J. Leskovec, A. Rajaraman, and J. D. Ullman, *Recommendation Systems*, pp. 292–324. Cambridge University Press, 2 ed., 2014.

[94] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[95] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Analysis of recommendation algorithms for e-commerce," in *Proceedings of the 2Nd ACM Conference on Electronic Commerce*, EC '00, pp. 158–167, ACM, 2000.

[96] Z. Huang, W. Chung, and H. Chen, "A graph model for e-commerce recommender systems," *J. Am. Soc. Inf. Sci. Technol.*, vol. 55, no. 3, pp. 259–274, 2004.

[97] P. Melville, R. J. Mooney, and R. Nagarajan, "Content-boosted collaborative filtering for improved recommendations," in *Eighteenth National Conference on Artificial Intelligence*, pp. 187–192, American Association for Artificial Intelligence, 2002.

[98] P. Cremonesi, Y. Koren, and R. Turrin, "Performance of recommender algorithms on top-n recommendation tasks," in *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, pp. 39–46, ACM, 2010.

[99] https://grouplens.org/datasets/, 2009.

[100] http://files.grouplens.org/datasets/movielens/ml-10m-README.html, 2009.

[101] `https://www.kaggle.com/gspmoreira/recommender-systems-in-python-101`.

[102] T. Jaya Lakshmi and S. Durga Bhavani, "Temporal probabilistic measure for link prediction in collaborative networks," *Applied Intelligence*, vol. 47, pp. 83–95, Jul 2017.