

The Pennsylvania State University
The Graduate School

LINK AGE: A FACTOR IN LINK PREDICTION IN A SOCIAL NETWORK

A Thesis in
Electrical Engineering
by
Samet Akcay

© 2015 Samet Akcay

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

August 2015

The thesis of Samet Akcay was reviewed and approved* by the following:

David J. Miller
Professor of Electrical Engineering

Kenneth Jenkins
Professor of Electrical Engineering

Kultegin Aydin
Professor of Electrical Engineering
Head of the Department of Electrical Engineering

*Signatures are on file in the Graduate School.

Abstract

This work extends a previous one that investigated link age and its effect on network evolution. Whether aging adversely influences prediction power of links in network evolution is the fundamental question partially answered in the previous work. Additionally, this study argues whether reliable old connections in a network have a great impact on future link predictions. One of our hypotheses is that aging of a link is a crucial factor in link prediction. The other one is that prediction power of a link usually lessens over time. Using logistic regression and mixture extension, younger links are observed to dominate the link prediction process in most cases. However, this is not always the case. We cannot ignore the links that are old but still powerful. In addition to prediction power of the links, using a mixture model improves the overall link prediction accuracy. The findings of this research support the implications of the previous work that some old and unstable links might be removed from the network.

Table of Contents

List of Figures	vi
List of Tables	vii
List of Symbols	viii
Acknowledgments	ix
Chapter 1	
Introduction	1
Chapter 2	
Related Work	5
Chapter 3	
Methodology	8
3.1 DBLP Dataset	8
3.2 Features	9
3.3 Model: Logistic Regression and a Mixture Extension for Link Prediction	10
3.3.1 Mixture of Logistic Regression	12
3.3.2 Maximum Likelihood	13
3.3.3 EM Algorithm	13
3.3.3.1 Expectation Step	14
3.3.3.2 Maximization Step	15
3.4 Evaluation Methods	19
Chapter 4	
Experimental Results	21
4.1 Optimization Results	21
4.2 Relative Performance of the Classifiers	24
4.2.1 Relative Performance of C_1	25
4.2.2 Relative Performance of C_2	25
4.2.3 Relative Performance of C_3	25
4.2.4 Relative Performance of C_4	25

4.2.5	Comparison of the Relative Performance of the Classifiers	26
Chapter 5		
	Conclusion, Discussion and Future Work	34
5.1	Summary of the Thesis	34
5.2	Discussion	35
5.3	Future Work	35
5.4	Conclusion	36
Appendix		
	Feature Extraction	37
1	Fundamentals of the Data	37
2	Generating the Co-Author Network	37
2.1	Updating Co-Authorship Networks	41
3	Generating Mutual Friends	43
4	Generating Feature Matrix	44
Bibliography		47

List of Figures

1.1	A Simple Network showing the Connections with respect to years	1
1.2	Visualization of Source and Target Networks	3
2.1	Network Generation and Evolution	5
3.1	Generating labels	10
4.1	Objective Function with Different Initial Values	22
4.2	Objective Function of C1 with different number of components.	22
4.3	Relative Performance of C1 with different number of components in the range between 2004 and 2007	27
4.4	Relative Performance of C2 with different number of components in the range between 2004 and 2007	28
4.5	Relative Performance of C3 with different number of components in the range between 2004 and 2007	29
4.6	Relative Performance of C4 with different number of components in the range between 2004 and 2007	30
4.7	Relative Performance of C1,C2,C3,C4 when M=1 in the range between 2004 and 2007	31
4.8	Relative Performance of C1,C2,C3,C4 when M=4 in the range between 2004 and 2007	32
4.9	Relative Performance of C1,C2,C3,C4 when M=7 in the range between 2004 and 2007	33
A.1	Data visualization of table (A.5)	39
A.2	Author #1's network in year 2000.	40
A.3	Mutual Friends of 5-76 in 2003. Table (A.10)	43

List of Tables

3.1	Data Characteristics	8
3.2	Features with respect to years	9
3.3	Classifiers	10
3.4	Contingency Table	20
4.1	Learned Coefficients of the Classifier C1 in the year 2003	23
4.2	Mixing Coefficients of the Classifier C1 in the year 2003	23
4.3	Learned coefficients of classifiers C2,C3 and C4 in the year of 2003	24
4.4	Mixing coefficients of the classifiers C2,C3 and C4 in the year 2003.	24
A.1	Data Characteristics	37
A.2	A Sample data	38
A.3	Sample data: Authors who do not have at least 5 papers.	38
A.4	Sample data: Authors with at least 5 papers.	39
A.5	Sample data of the year 2000.	40
A.6	Co-Authors of the sample data of year 2000	41
A.7	Co-Authors of the sample data of year 2001	41
A.8	Co-Authors of the sample data of year 2002	41
A.9	Co-Authors of the sample data of year 2003	42
A.10	Co-Author Network of 2000 and 2001.	42
A.11	Co-Author Network of 2002 and 2003.	42
A.12	Mutual Friends of co-author network in 2000.	44
A.13	Features with respect to years	45
A.14	Feature Matrix of years 2000,2001,2002	46

List of Symbols

θ_k	Coefficient of the k^{th} component of the logistic function, p. 11
\mathbf{X}	Feature vector, p. 11
p	Logistic Function when $y = 1$ and given feature vector and logistic regression coefficient, p. 12
α_k	Mixing coefficient of the k^{th} component, p. 12
L	Likelihood function of the mixture of the logistic function, p. 13
ℓ	Log-Likelihood function of the mixture of the logistic function, p. 13
ℓ_c	Complete Data Log-Likelihood, p. 14
ℓ_{ic}	Incomplete Data Log-Likelihood, p. 16
$V_{ij,k}$	Binary random variable, p. 14
RP_n	Relative Performance of the first n predictions, p. 19
TPR_n	True positive rate of the first n predictions, p. 19
TPR_{rand}	True positive rate of a random naive guessing, p. 19

Acknowledgments

First of all, I would like to express my gratitude to my supervisor Dr. David J. Miller for being not only an advisor but also a mentor. Throughout my thesis process, he supported me with his patience and profound knowledge. He also encouraged me to finish the research on time by always motivating me during our meetings. Hence, he deserves the greatest appreciation.

During the times I worked at Robust Machine Intelligence and Control Lab, I had the greatest opportunity to have Hossein Soleimani and Zhicong Qiu as PhD fellow students. I really enjoyed the discussions and arguments about network theory and machine learning applications that are related to this work.

I also would like to thank Dr. Kenneth Jenkins for willingly agreeing on being a committee member.

I finally want to thank my wife, parents and sisters for supporting me throughout the master level. My wife always encouraged me even when I am in the toughest part of the work. Also, my parents and sisters always motivated me to conduct and finish this research even though they are thousands of miles away!

Introduction

With whom would an author be interested in collaborating in his academic research? Another co-author of those with whom he published a paper 10 years ago, or last year? Or, is it likely to expect that he is willing to receive LinkedIn connection recommendations based on the one he met 5 years ago? Or, let us ask a question from his daily routine. Would he still want to receive advertisements related to an item that he added on his wish list 3 years ago?

Most of the popular network properties are based on network growth. These properties indicate that a network always evolves by new vertex additions. This means that a link always remain in the network after being connected. At this point, however, the stability of the links becomes questionable. In other words, one cannot guarantee that links will always be active once they are added. In this work, we study how the effectiveness of links in a network changes over time. Specifically, we answer this problem in terms of their efficacy in predicting possible future link formations.

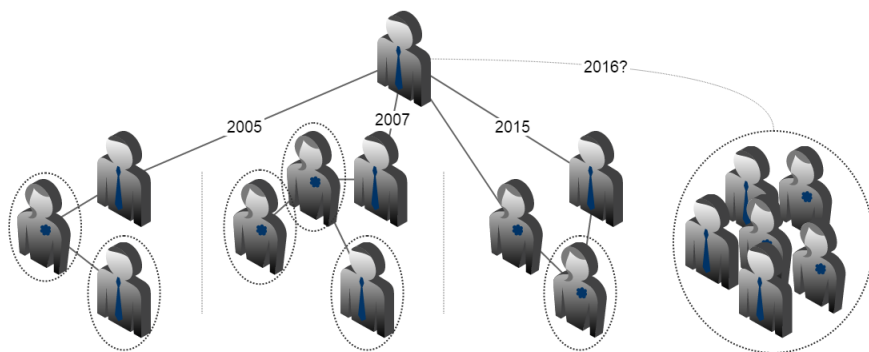


Figure 1.1: A Simple Network showing the Connections with respect to years

Studies in network theory have proposed many models that capture the overall statistics of a network. Thanks to these models, networks with high clustering coefficient, scale free power distributions can be generated. Some studies have also revealed that degree information

of networks increases with new nodes, while the diameter of the network does decrease. This implies, hence, that networks expand over time. Since networks gradually develop, predicting future nodes and edges has attracted scientist's interest. Many models accurately predicting future links have been introduced in recent years. Even though this has provided scientist with the opportunity to observe the factors affecting link prediction, few researches have investigated how aging has an impact on a link's power in predicting new connections.

Most of the related works propose models that generate networks, and which predict and recommend possible future links to these networks. This is where the question arises: Does a network always grow? Are the links in the network always active? If not, is there any way to realize which links do not contribute to the network evolution? [1] states that as a link ages, it is getting less powerful in predicting new links. However, this study does have some limitations such that there could be important old links in network, and their prediction power might not decrease. This means, [1] illuminates the problem to some extent, but we still need a broader solution.

Conducting this work is significant because main goal is to observe whether links are always active once they are connected to a network. We investigate links' contribution to network evolution, and consider the aging of the links as a factor affecting this contribution. Proving that link age might adversely influence the prediction power of a link might lead to the addition of the notion of 'vertex subtraction' to network theory.

In this study, our hypothesis is that younger links are usually more effective than the older ones in predicting future connections. We expect that as a link ages, it does not provide much information to predict possible link formations. This does not mean that all of the old links in a network gets weaker as they age. There are certainly old connections that are still powerful and informative on link prediction. However, many of the older links gradually become inactive in network evolution, and it is possible to delete these old and uninformative links from the network.

In order to validate the hypothesis, we treat this problem as a supervised learning task. This provides us the opportunity to solve this link prediction problem by using features and labels generated from the network. To generate the network, we use DBLP computer science dataset that includes research papers published in the field of computer science. Using the dataset, we create a co-authorship network by including authors who have at least 5 published papers. Each node in the network correspond to an author. Each link between two nodes means that two authors co-author a paper. We sort the co-authorship network with respect to years such that each year has its own network information. [1] demonstrates that author's collaboration period between one another radically decreases after 3 years. Hence, we define our source network as the combination of three consecutive years, and target network as the following year's co-authorship information. Figure (1.2) shows that source network has the years $y, y+1$ and $y+2$, while target network consists of the year $y+3$'s information.

We use source and target networks to generate features and labels, respectively, to predict the following year's link formations, and to observe how link age influence the link prediction. For

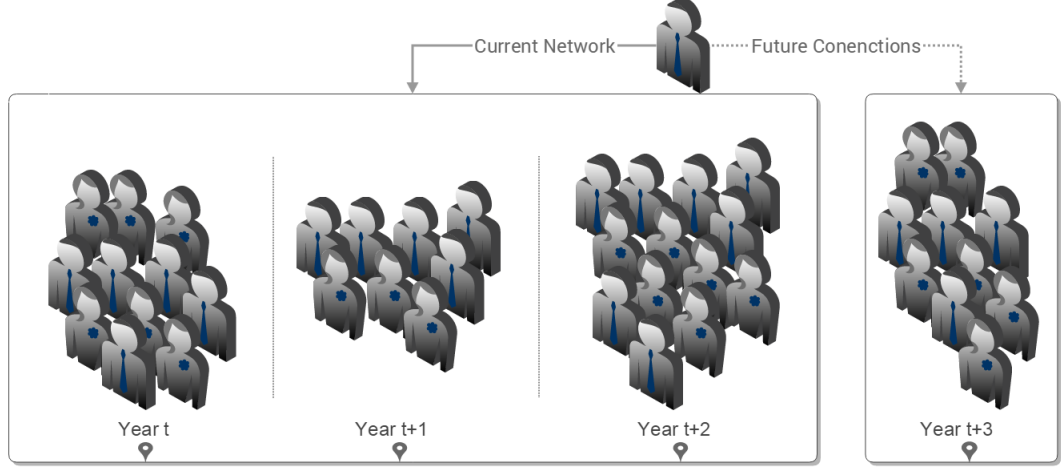


Figure 1.2: Visualization of Source and Target Networks

feature generation, we compare each links in the network, say $i - j$ pairs, and use the following:

$$x_1 = \sum_{\forall k \in m_{f_{ij}}} n_{ca,y}(i, k)$$

$$x_2 = \sum_{\forall k \in m_{f_{ij}}} n_{ca,y}(j, k)$$

where $n_{ca,y}(i, k)$ is the number of co-authored papers between nodes i and k in year y . The node k corresponds to each mutual friend that the nodes i and j have in common. Same process is repeated to find the total number of co-authored papers between the nodes j and k . Since we have three consecutive years in the source network, and each year has 2 individual features, we have 6 features in total. To create the labels that show the presence or absence of the links in the network, we use the target network. If nodes i and j are connected in the target network, the label, y_{ij} , is 1 and 0 otherwise.

Logistic regression classifier with a mixture extension is the model that we use. The reason why logistic regression is used is that learned coefficients of it would quantitatively describe which feature is more dominant. We use mixture extension because some links may be influenced by either old or young links. Or, in some cases, old links could be more important. There may be multiple profiles that explains different subsets of the network links.

Our experimental results show that younger links are generally more helpful than older ones in predicting the following year's links. In addition to what is found in [1], we also observe that in some cases, older connections are still strong and provide more information than younger links in predicting the future year's links. In fact, the performance of either young or old links are even better when we consider multiple component. This means that although younger connections provide more information than older ones in most cases, we cannot pigeonhole the idea that all of the older links gets weaker when they age.

The overall outline of this work is as follows: Chapter 2 reviews the previous work related to our study. Chapter 3 introduces the model that we use. Chapter 4 describes the data and the experimental procedure. Chapter 5 provides the results based on the experiments. Last but not least, Chapter 6 includes the discussions and future research that can be conducted.

Related Work

Generation and evolution of a network has been widely studied in recent years. Researchers have proposed numerous methods visualizing the statistical properties that give the overall picture of a network. Since networks are dynamic, many models predicting future formations have been presented. Besides, the link prediction in networks has been even interpreted as a supervised link prediction problem. However, few studies have focused on the significance of link age in forming future links in a network.

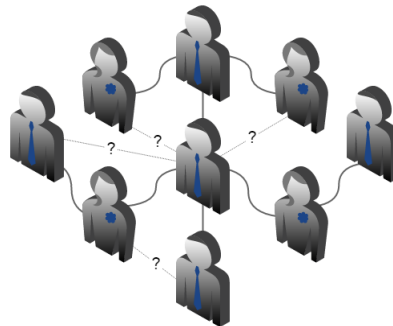


Figure 2.1: Network Generation and Evolution

Models approximating the statistical properties of networks have been thoroughly studied in literature. A network with high clustering coefficient and short average path length can be generated [2]. For larger networks, it has been shown that networks with scale-invariant power-law degree distribution can also be generated [3]. This is based on the fact that networks grow with new nodes; and these nodes usually connect to those whose connections are well-established (Preferential Attachment). Graphical distribution of a network is revealed to be inferred by power law [4]. It is also useful to generate complex networks [4]. Degree information of a graph is proven to be grown as the network evolves (Densification Power Law) [5]. Another model produces a network with heavy-tailed degree, the Densification Power Law, and a shrinking diameter [5].

Advances in structures of network topology are reviewed in [6, 7]. All these well known studies show that theory of network structure has already been developed.

Evolution of a network can be considered as adding more vertices to the network. To predict which nodes will be added to form new links in the network, vertex similarity measures can be used [8]. In addition to this, accuracy of link prediction is higher when local structured similarity measures are used [9, 10]. Al Hasan et.al [11] is one of the first studies considering the link prediction problem as a supervised learning task. This means that presence or absence of links in a network can be the labels, and global or local statistics of the network can be expressed as the features of this supervised classification [11]. Their approach is simply extracting features from network, and labeling the links such that they belong to particular classes according to their presence or absence in the network. Many of the supervised classification techniques are compared in [11] to find the prediction accuracy. Even though all of the techniques provide satisfactory results, it is pointed out in the paper that SVM is the best one among all of the methods. Papers [12], [13], [14], are the literature surveys that review most of the popular work in detail.

Even though network generation and evolution has been widely studied, influence of link age on formation of new links has yet to attract much attention [1]. This phenomenon has been studied after link prediction problem is approached as a supervised link prediction task [11]. Chen et al. [1] enlightens two unanswered questions in literature: First, how the interaction of nodes change over time; and second, how the age of a link has an impact on predicting the following year's link info. They first realized that the collaboration periods among nodes are no more than 3 years. What is observed in their experiments is that connection of nodes drastically decreases after 3 years. In order to answer the second question, logistic regression classifier is used [1]. Learned coefficients of the logistic regression classifier show [1] that younger links dominate the prediction of the following year's links. Hence, inactive and unimportant links are claimed to be deleted from the network [1].

In our work, we want to evaluate the link age as a factor when predicting future links. As in [11], we consider this as a supervised learning problem. What is different in our study is that we evaluate the importance of link age, while [11] concentrates on prediction accuracy of supervised learning methods such as decision tree, K-NN, MLP and SVM. As mentioned above, [1] also studies link age as a factor of link prediction. What makes our study unique is that we use logistic regression classifier and its mixture extension, which is a more powerful model than used in [1], allowing multiple link formation profile. Hence, our proposed model is able to cover more scenarios than the one given in [1].

In summary, in literature, there are countless well-known models that generates networks with shortest path length, high clustering coefficient, scale free power-law distribution etc. Besides,

some notable researchers in network evolution area have proposed particular properties such that network always expands by new vertices. Network evolution has also been studied in terms of predicting possible future connections of the nodes. Many models accurately predicting new links have been introduced. Even though these problems have been widely investigated, how active the links are in link prediction as they age is still an open area to study. In this work, we answer this question.

Methodology

This section introduces the methods to conduct this research. We first introduce the dataset we use to perform experiments. Then, we discuss how the features are generated from the dataset. Next, we introduce logistic function, derive its mixture extension for link prediction problems and propose it as our model for this work. We finally, describe our performance criteria showing how this work achieves.

3.1 DBLP Dataset

To create the co-authorship network and to find the influence of link age in new link prediction, we use the DBLP Computer Science Bibliography data set. The data is collected between the years 1905 and 2007, and includes the detailed list of research papers published in the field of computer science. It has the following information: author, paper and the year that the paper is published.

Table 3.1: Data Characteristics

Number of Nodes	<i>178,154</i>
Number of Edges	<i>3,621,265</i>

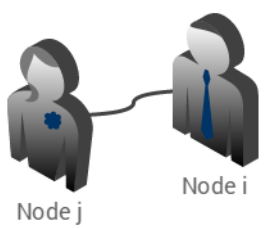
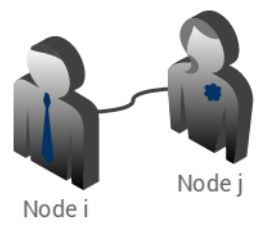
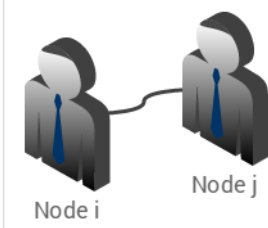
Table (3.1) shows that the number of nodes of the data set is 178,154. When predicting new links, we compare each node to the rest of the nodes in the network. In this case, the data set becomes tremendously large, which causes computational problems. In order to avoid such complications, we sample 20,000 nodes from the data.

In order to train the data we use the year 2003's network. Information of the years between 2004 and 2007, on the other hand, is used for testing.

3.2 Features

After the data is obtained and sampled, the next step is generating the features to perform the experiments. We use the following six features extracted from the years $t, t+1, t+2$;

Table 3.2: Features with respect to years

 <p>Node j Node i</p> <p>Year y</p>	 <p>Node i Node j</p> <p>Year y+1</p>	 <p>Node i Node j</p> <p>Year y+2</p>
$x_1 = \sum_{\forall k \subset m_{f_{ij}}} n_{ca,t}(i, k)$ $x_4 = \sum_{\forall k \subset m_{f_{ij}}} n_{ca,t}(j, k)$	$x_2 = \sum_{\forall k \subset m_{f_{ij}}} n_{ca,t+1}(i, k)$ $x_5 = \sum_{\forall k \subset m_{f_{ij}}} n_{ca,t+1}(j, k)$	$x_3 = \sum_{\forall k \subset m_{f_{ij}}} n_{ca,t+2}(i, k)$ $x_6 = \sum_{\forall k \subset m_{f_{ij}}} n_{ca,t+2}(j, k)$

Where the features $x_1 - x_4$, $x_2 - x_5$ and $x_3 - x_6$ belong to the years $t, t+1, t+2$, respectively. Describing $x_1 - x_4$ will be sufficient to understand $x_2 - x_5$ and $x_3 - x_6$ since the only distinction between these is that they are generated using different year's data. $n_{ca,t}(i, k)$ is the number of co-authored papers between the nodes i and k where k denotes each mutual friend that the nodes i and j have in common. Hence, the notation $\forall k \subset m_{f_{ij}}$ shows the mutual neighbors of i and j .

As described before, x_1 and x_4 are the features extracted from the year t . The difference between x_1 and x_4 is that the former generates the number of co-authored paper between the nodes i and k , while the latter finds the number of co-authored papers between j and k .¹

After generating the feature matrix, we re-scale it such that each feature has the value between 0 and 1. By re-scaling the features, values of the learned coefficients become the most significant factor to evaluate the performance of the model.

Assuming that the dimension of the feature matrix is d , we have the following row for each $i - j$ pair;

$$\widetilde{X}_{ij} = [x_1, x_2, x_3, \dots, x_d] \quad (3.1)$$

After the normalization process, we have;

$$\tilde{X} = \left[\frac{x_1 - \min(X_{ij})}{\max(X_{ij}) - \min(X_{ij})}, \frac{x_2 - \min(X_{ij})}{\max(X_{ij}) - \min(X_{ij})}, \dots, \frac{x_d - \min(X_{ij})}{\max(X_{ij}) - \min(X_{ij})} \right] \quad (3.2)$$

¹Generating a co-authorship network, mutual friends and feature matrix is thoroughly described in Appendix (A).

Table (3.2) shows that we use 6 features in this study. In this case, the dimension of our feature matrix is $d = 6$, and we have 6 data points to normalize for each realization.

The network of the years $t, t+1$ and $t+2$ are defined as the source network, while $t+3$ is the target network. We use the features of the source network given in table (3.2) to predict the links in the target network. When doing so, we only use the nodes that are in both source and target networks. This means that we ignore the nodes that are only in the source or target networks. We also do not consider the links connected in both networks since we want to concentrate only on new connections. Hence, we pay no attention to edges that are in both of these networks.

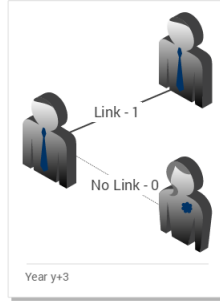


Figure 3.1: Generating labels

Since this is a supervised link prediction problem, we need class labels. To get these labels, we use the links in target network that is either 1 or 0. The class label 1 means that there is a link between $i - j$ pair, while 0 means there is not.

Table 3.3: Classifiers

	C1	C2	C3	C4
t	✓			✓
t+1	✓		✓	
t+2	✓	✓		

In order to compare the performance of the links with respect to particular years, we use 4 different classifiers. Classifier C_1 includes all of the features, while C_2 , C_3 and C_4 have only the youngest, mid-age and the oldest links, respectively.

3.3 Model: Logistic Regression and a Mixture Extension for Link Prediction

Before introducing the logistic regression concept, let us recall the problem we are concentrating on. We investigate the effect of link age on new link predictions. In order to find an answer to this, the problem should be divided into two separate parts: first, we have networks of consecutive years, and second, we want to observe how these networks influence the following

year's network formation.

To observe how the networks of the consecutive years impact the following year, we need to combine their features so that we have meaningful information to predict the following year's connection. This means that we are to find the prediction power of each feature in order to observe their importance.

The second part of the problem is whether these networks have an impact on presence or absence of the links in the network of the following year. Link existence implies binary outcomes such that the outcome is 1 and 0 in case of link existence and non-existence, respectively. To solve the second part of the problem, we need to predict the outcome of the class labels.

Since we want to know how features of the input affect output and to predict the outcomes of it, logistic regression model is a perfect match for our purpose. The reason is that one can infer the strength of each feature by the coefficients of the logistic regression function. This advantage of the logistic regression model helps us to solve the first part of our problem. Logistic regression is also widely used to predict the outcomes of dependent variables such as class labels, which is another reason why it is a good fit as a model.

We have described the problem that we want to solve and the model we want to use. We now introduce the logistic function. Then we will derive the conditional likelihood and the log likelihood to maximize the probabilities.

The logistic regression function is the following;

$$P(Y = 1|\tilde{\mathbf{X}}; \underline{\theta}) = \frac{1}{1 + e^{(\theta_0 + \sum_{k=1}^d x_k \theta_k)}} \quad (3.3)$$

The term $(\theta_0 + \sum_{k=1}^d x_k \theta_k)$ is nothing but the expression of a linear regression function whose range is between $[-\infty, \infty]$. Since this term is used inside an exponential function as given in equation (3.3), $P(Y = 1|X; \theta)$ tends to 1 and 0 as $(\theta_0 + \sum_{k=1}^d x_k \theta_k)$ goes to ∞ and $-\infty$, respectively. $\theta_0, \dots, \theta_d$ and x_1, \dots, x_d are the parameters of the logistic function and the features, respectively. d denotes the dimension of the feature vector. In order to avoid notational complexity, let us define $x_0 = 1$, and write the following:

$$\begin{aligned} P(Y = 1|\mathbf{X}; \underline{\theta}) &= \frac{1}{1 + e^{(\sum_{k=0}^d x_k \theta_k)}} \\ &= \frac{1}{1 + e^{-(\mathbf{X}\underline{\theta}^T)}} = p \end{aligned} \quad (3.4)$$

where the parameters are,

$$\begin{aligned} \underline{\theta} &= [\theta_0, \theta_1, \dots, \theta_d] \\ \tilde{\mathbf{X}} &= [x_1, x_2, \dots, x_d] \\ \mathbf{X} &= [1 \ \tilde{\mathbf{X}}] \end{aligned} \quad (3.5)$$

One can think of computing the probability of logistic regression as parameterized Bernoulli

distribution. What this means is that if the probability of link presence, $P(Y = 1|X; \theta)$, is defined as p , then the probability of link absence is $1 - p$. Assuming that y denotes the link presence or absence, this can be shown as the following:

$$P(Y = y|X; \theta) = \begin{cases} p, & y = 1 \\ 1 - p, & y = 0 \end{cases} \quad (3.6)$$

The above equation is given separately. Merging these two separate probabilistic equations will notationally be convenient for future derivations. Since the probabilities are p and $1-p$ for $y=1$ and $y=0$, respectively, we can write the equation as follows:

$$P(Y = y|X; \theta) = p^y(1 - p)^{(1-y)} \quad (3.7)$$

Equation (3.7) given above give us the opportunity to derive an equation such that the probability can be found for each y . For example, when $y = 1$, the equation becomes $P(Y = 1|X; \theta) = p$, and similarly, $P(Y = 0|X; \theta) = (1 - p)$ when $y = 0$.

3.3.1 Mixture of Logistic Regression

We have introduced the notion of logistic regression, and given its probability mass function. In our study, however, we want to use a mixture of logistic functions. The reason for using mixture is that some link formations may be strongly influenced by new links, while others may be influenced by reliable old links. More generally, there may be multiple predictive profiles that are each helpful in predicting part of the network in a future year. To mix the model, we use individual distributions and mixing coefficients that sum to one.

$$P[Y = y|X; \Lambda] = \sum_{k=1}^M \alpha_k P_k[Y = y|X; \theta_k] \quad (3.8)$$

where Λ includes the parameters $\alpha_1, \dots, \alpha_M$ and $\{\theta_{k0}, \dots, \theta_{kd}\}$ such that $k = [1, 2, \dots, M]$, and d is the dimension of the feature vector, respectively. Recall that α_k 's are the mixing coefficients such that $\sum_{k=1}^M \alpha_k = 1$ where M is the number of components. $P_k[Y = y|X; \theta_k]$ is the individual distribution of the component k for a given feature and parameter vectors X and θ_k , respectively.

$$\Lambda = (\{\alpha_k\}, \{\theta_{k,j}\}) \quad (3.9)$$

We are to optimize the parameters in Λ to maximize the likelihood of $P(Y = y|X; \theta_k)$. Before discussing the optimization part, we will introduce the notion of maximum likelihood in the following section.

3.3.2 Maximum Likelihood

As discussed in section (3.3.1), we have mixture of logistic regression, $P(Y = y|\mathbf{X}; \theta_k)$, whose likelihood is to be maximized. We already discussed that coefficients in equation (3.9) are the ones to be optimized so that we have the maximum likelihood. In order to optimize these parameters, we have an objective function, which is a likelihood function in this case. Assuming the data has N nodes to compare, and all of these realizations are independent of each other, the likelihood function can be written as the following;

$$L = \prod_{i=1}^N \prod_{j=i+1}^N P[Y_{ij} = y_{ij} | \tilde{\mathbf{X}}_{ij}; \Lambda] \quad (3.10)$$

where i and j are individual nodes, and $i - j$ is the edge showing the link information in a network. It is common in machine learning applications to take the logarithm of the likelihood function for mathematical and notational convenience. In so doing, we have the following;

$$\ell = \sum_{i=1}^N \sum_{j=i+1}^N \log P[Y_{ij} = y_{ij} | \tilde{\mathbf{X}}_{ij}; \Lambda] \quad (3.11)$$

$$= \sum_{i=1}^N \sum_{j=i+1}^N \log \sum_{k=1}^M \alpha_k P_k[Y_{ij} = y_{ij} | \mathbf{X}_{ij}; \theta_k] \quad (3.12)$$

Equation (3.12) shows that the log likelihood of the mixture of logistic regression is found by simply summing the logarithm of the mixture densities of each realization. Having found an equation for the log-likelihood function, we now write the complete data log-likelihood.

3.3.3 EM Algorithm

In order to estimate the parameters that maximizes the likelihood, we use Expectation-Maximization (EM) algorithm [15] that is widely used in pattern recognition community. EM algorithm is an iterative process that estimates the parameters that maximizes the likelihood. It is useful especially when the data has latent variables.

EM algorithm has two main steps: Expectation and Maximization steps. In the expectation step, the algorithm computes the expectation of the log-likelihood function by assuming all of the latent variables are observed. This is done for each iteration. Then, in the maximization step, also known as the M-Step, the maximum likelihood function is maximized by using the quantities found in the E-Step. Since this is an iterative algorithm, the parameters that are maximized in the M-Step of the i_{th} iteration are used in the E-Step of the $(i + 1)_{th}$ iteration. This process continues until the parameters and the log-likelihood function converges, which means that there is not a significant change in parameters anymore.

[15] proves that the likelihood function used in the EM Algorithm never decreases during EM iterations. Based upon this fact, we expect in theory that objective function monotonically converges after some point.

3.3.3.1 Expectation Step

The data that we use has obviously latent (hidden) variables [16]. In this section, however, we assume that we observe the latent variables and know the following binary random variable such that it is 1 if a particular component is responsible for generating the information for a link i - j , and 0, otherwise. This binary random variable can be shown as;

$$V_{ij,k} = \begin{cases} 1, & \text{if } k^{th} \text{ component generated } (i,j) \text{ link info} \\ 0, & \text{otherwise} \end{cases} \quad (3.13)$$

If $V_{ij,k}$ is defined as above, and assuming the data is complete, the log-likelihood function, ℓ_c can be written as;

$$\ell_c = \sum_{i=1}^N \sum_{j=i+1}^N \log \sum_{k=1}^M V_{ij,k} P_k[Y_{ij} = y_{ij} | \mathbf{X}_{ij}; \theta_k] \quad (3.14)$$

$$= \sum_{i=1}^N \sum_{j=i+1}^N \sum_{k=1}^M V_{ij,k} P_k[Y_{ij} = y_{ij} | \mathbf{X}_{ij}; \theta_k] \quad (3.15)$$

where, $(i - j)$ is the pair that is linked in network. N and M are the total number of nodes and the number of components, respectively. As mentioned, $V_{ij,k}$ is the binary random variable that shows whether or not k_{th} component generated the link $(i - j)$. $P_k[Y = 1 | X; \Theta]$ is the probability that there is a link between $(i - j)$ pairs.

If there is a link between pair $(i - j)$, then we get $P_k[Y_{ij} = 1 | \tilde{\mathbf{X}}_{ij}; \Theta]$ or p . By the same token, if there is no link between $(i - j)$, then we get $P_k[Y_{ij} = 0 | \tilde{\mathbf{X}}_{ij}; \Theta]$ or $(1 - p)$.

Taking the expectation of the complete data log-likelihood, ℓ_c , we get;

$$E[\ell_c | X; \Theta] = E \left[\sum_{i=1}^N \sum_{j=i+1}^N \log \sum_{k=1}^M V_{ij,k} P_k[Y_{ij} = y_{ij} | \mathbf{X}_{ij}; \theta_k] \right] \quad (3.16)$$

$$\begin{aligned} &= \sum_{i=1}^N \sum_{j=i+1}^N \sum_{k=1}^M E[V_{ij,k} | \mathbf{X}_{ij}] \log(\alpha_k P_k[Y_{ij} = y_{ij} | \mathbf{X}_{ij}; \theta_k]) \\ &= \sum_{i=1}^N \sum_{j=i+1}^N \sum_{k=1}^M E[V_{ij,k} | \mathbf{X}_{ij}] (\log \alpha_k + \log P_k[Y_{ij} = y_{ij} | \mathbf{X}_{ij}; \theta_k]) \\ &= \sum_{i=1}^N \sum_{j=i+1}^N \sum_{k=1}^M E[V_{ij,k} | \mathbf{X}_{ij}] \log \alpha_k \\ &\quad + \sum_{i=1}^N \sum_{j=i+1}^N \sum_{k=1}^M E[V_{ij,k} | \mathbf{X}_{ij}] \log P_k[Y_{ij} = y_{ij} | \mathbf{X}_{ij}; \theta_k] \end{aligned} \quad (3.17)$$

From equation (3.17), it is possible to recognize $E[V_{ij,k} | \mathbf{X}_{ij}]$ as the posterior probability of component k , i.e $P[\mu_{ij} = k | \mathbf{X}_{ij}]$. Besides, using Bayes's rule, an equation can be derived for

$P[V_{ij,k} = k | \mathbf{X}_{ij}; \theta_k]$. In so doing, we have the following;

$$E[V_{ij,k} | \mathbf{X}_{ij}; \theta_k] = P[V_{ij,k} = 1 | \mathbf{X}_{ij}; \theta_k] \quad (3.18)$$

$$= \frac{P[Y_{ij} = y_{ij} | V_{ij,k}, \mathbf{X}_{ij}; \theta_k] P[V_{ij,k} = 1 | \mathbf{X}_{ij}; \theta_k]}{P[Y_{ij} = y_{ij} | \mathbf{X}_{ij}; \Theta]} \quad (3.19)$$

$$= \frac{\alpha_k P_k[Y_{ij} = y_{ij} | \mathbf{X}_{ij}; \theta_k]}{\sum_{l=1}^M \alpha_l P_l[Y_{ij} = y_{ij} | \mathbf{X}_{ij}; \theta_l]} \quad (3.20)$$

We found an equation for the expectation of the binary random variable; and it is given in equation (3.20).

3.3.3.2 Maximization Step

Equations (3.16) and (3.17) show how the expectation of the complete data log-likelihood can be expressed. We now maximize the parameters α and θ using the expectation equation derived in (3.17). Since the parameters α and β are independent of each other, we first optimize α , then θ , respectively. We will then use these maximized parameters in the expectation step of the next iteration.

Maximizing α

In order to maximize α , we are to take the derivative of equation (3.17), and set it to zero.

$$\frac{\partial}{\partial \alpha_k} (E[\ell_c | \mathbf{X}; \Theta]) = 0 \quad (3.21)$$

Since the second term of equation (3.17) does not include the parameter α , its derivative will be zero. Hence, we only concentrate on the first part of the equation.

$$\frac{\partial}{\partial \alpha_k} (E[\ell_c | \mathbf{X}; \Theta]) = \frac{\partial}{\partial \alpha_k} \left(\sum_{i=1}^N \sum_{j=i+1}^N \sum_{k=1}^M E[V_{ij,k} | X_{ij}] \log \alpha_k \right) \quad (3.22)$$

Knowing from equation (3.18) that $E[V_{ij,k} | X_{ij}]$ can be expressed as $P[V_{ij,k} = 1 | \mathbf{X}_{ij}; \theta_k]$, we can write the following:

$$\frac{\partial}{\partial \alpha_k} (E[\ell_c | \mathbf{X}; \Theta]) = \frac{\partial}{\partial \alpha_k} \left(\sum_{i=1}^N \sum_{j=i+1}^N \sum_{k=1}^M P[V_{ij,k} = 1 | X_{ij}; \theta] \log \alpha_k \right) \quad (3.23)$$

We use Lagrangian multiplier to maximize α_k , and set the derivative to zero. Since the mixing coefficients sum to 1, lagrangian multiplier λ has the constraint such that $\sum_{k=1}^M \alpha_k = 1$. Thus,

the equation will be as,

$$\frac{\partial}{\partial \alpha_k} \left[\sum_{i=1}^N \sum_{j=i+1}^N \sum_{k=1}^M P[V_{ij,k} = 1 | \mathbf{X}_{ij}; \theta_k] \log \alpha_k + \lambda \left(\sum_{k=1}^M \alpha_k - 1 \right) \right] = 0 \quad (3.24)$$

To take the derivative of equation (3.24), we are to consider the terms $\log \alpha_k$ and $\lambda \left(\sum_{k=1}^M \alpha_k - 1 \right)$ since these are the only ones with the parameter α . After taking the derivative, we can write the following;

$$\sum_{i=1}^N \sum_{j=i+1}^N \frac{1}{\alpha_k} P[V_{ij,k} = 1 | \mathbf{X}_{ij}; \theta_k] + \lambda = 0 \quad (3.25)$$

Solving equation (3.25) yields the equation that maximizes α .

$$\alpha_k^{(n+1)} = \frac{1}{\binom{N}{2}} \sum_{i=1}^N \sum_{j=i+1}^N P[V_{ij,k} = 1 | \mathbf{X}_{ij}; \theta_k] \quad (3.26)$$

where N is the number of nodes. $\sum_{i=1}^N \sum_{j=i+1}^N$ gives us all of the $(i - j)$ pairs in the network. $P[V_{ij,k} = 1 | \mathbf{X}_{ij}; \theta_k]$ is the expectation of the binary random variable $V_{ij,k}$ as given in equation (3.18).

Maximizing θ

Once the mixing coefficients, $\alpha_k^{(n+1)}$, are maximized, the next, and the final step in this iteration is to find an equation for the parameters θ for the optimization process. To do so, we need the gradient ascent algorithm that is given below;

$$\theta^{(n+1)} = \theta^{(n)} + \mu \nabla_{\theta} \ell_{ic} \quad (3.27)$$

where ℓ_{ic} is the incomplete data log likelihood and shown as,

$$\ell_{ic} = \sum_{i=1}^N \sum_{j=i+1}^N \log \sum_{k=1}^M \alpha_k P_k[Y_{ij} = y_{ij} | \mathbf{X}_{ij}; \theta_k] \quad (3.28)$$

$$\nabla_{\theta} \ell_{ic} = \frac{\partial}{\partial \theta_{k,t}} \left(\sum_{i=1}^N \sum_{j=i+1}^N \log \sum_{k=1}^M \alpha_k P_k[Y_{ij} = y_{ij} | \mathbf{X}_{ij}; \theta_k] \right) \quad (3.29)$$

$$= \sum_{i=1}^N \sum_{j=i+1}^N \frac{\partial}{\partial \theta_{k,t}} \left(\log \sum_{k=1}^M \alpha_k P_k[Y_{ij} = y_{ij} | \mathbf{X}_{ij}; \theta_k] \right) \quad (3.30)$$

where the indices k and t correspond to the components and features, respectively.

Let $A = \sum_k \alpha_k P_k[Y_{ij} = y_{ij} | \mathbf{X}_{ij}; \theta_k]$. Then (3.30) can be shown as,

$$\nabla_{\theta} \ell_{ic} = \sum_{i=1}^N \sum_{j=i+1}^N \frac{\partial}{\partial \theta_{k,t}} (\log A) \quad (3.31)$$

$$= \sum_{i=1}^N \sum_{j=i+1}^N \frac{1}{A} \frac{\partial}{\partial \theta_{k,t}} (A) \quad (3.32)$$

We find $\frac{\partial}{\partial \theta_{k,t}} (A)$ in order to solve (3.32).

$$\frac{\partial}{\partial \theta_{k,t}} (A) = \frac{\partial}{\partial \theta_{k,t}} \left(\sum_k \alpha_k P_k[Y_{ij} = y_{ij} | \mathbf{X}_{ij}; \theta_k] \right) \quad (3.33)$$

$$= \alpha_k \frac{\partial}{\partial \theta_{k,t}} (P_k[Y_{ij} = y_{ij} | \mathbf{X}_{ij}; \theta_k]) \quad (3.34)$$

Equation (3.32) has been simplified into the equation (3.34). To go further, $\frac{\partial}{\partial \theta_{k,t}} (P_k[Y_{ij} = y_{ij} | \mathbf{X}_{ij}; \theta_k])$ is found. From (3.7),

$$\frac{\partial}{\partial \theta_{k,t}} (P_k[Y_{ij} = y_{ij} | \mathbf{X}_{ij}; \theta_k]) = \frac{\partial}{\partial \theta_{k,t}} \left(p_{ij,k}^{y_{ij}} (1 - p_{ij,k})^{(1-y_{ij})} \right) \quad (3.35)$$

$$\begin{aligned} &= y_{ij} p_{ij,k}^{(y_{ij}-1)} \frac{\partial}{\partial \theta_{k,t}} (p_{ij,k}) (1 - p_{ij,k})^{(1-y_{ij})} \\ &+ p_{ij,k}^{y_{ij}} (1 - y_{ij}) (1 - p_{ij,k})^{(-y_{ij})} \frac{\partial}{\partial \theta_{k,t}} (-p_{ij,k}) \end{aligned} \quad (3.36)$$

$$= \frac{\partial}{\partial \theta_{k,t}} (p_{ij,k}) \frac{p_{ij,k}^{y_{ij}}}{(1 - p_{ij,k})^{y_{ij}}} \left(\frac{(1 - p_{ij,k})}{p_{ij,k}} y_{ij} - (1 - y_{ij}) \right) \quad (3.37)$$

In order to find an answer to (3.37), $\frac{\partial}{\partial \theta_{k,t}} p_{ij,k}$ is derived. Let us define e_p as $e^{-(\mathbf{X}\theta^T)}$, then, using (3.4), we can write the derivative as,

$$\frac{\partial p_{ij,k}}{\partial \theta_{k,t}} = \frac{\partial}{\partial \theta_{k,t}} \frac{1}{1 + e_p} \quad (3.38)$$

Then,

$$\frac{\partial p_{ij,k}}{\partial \theta_{k,t}} = (-1)(1 + e_p)^{-2} \frac{\partial}{\partial \theta_{k,t}} e_p$$

$$\begin{aligned}
&= (-1)(1 + e_p)^{-2} e_p \frac{\partial}{\partial \theta_{k,t}} \left(- \sum_k^M x_{ij,k} \theta_{k,t} \right) \\
&= (-1)(1 + e_p)^{-2} e_p (-x_{ij,t}) \\
&= (-1) \frac{1}{1 + e_p} \frac{e_p}{1 + e_p} (-x_{ij,t}) \\
&= p_{ij,k} (1 - p_{ij,k}) x_{ij,t}
\end{aligned} \tag{3.39}$$

If (3.39) is plugged into (3.37),

$$\frac{\partial}{\partial \theta_{k,t}} (P_k[Y_{ij} = y_{ij} | \mathbf{X}_{ij}; \theta_k]) = p_{ij,k} (1 - p_{ij,k}) x_{ij,t} \frac{p_{ij,k}^{y_{ij}}}{(1 - p_{ij,k})^{y_{ij}}} \left(\frac{(1 - p_{ij,k})}{p_{ij,k}} y_{ij} - (1 - y_{ij}) \right) \tag{3.40}$$

$$\begin{aligned}
&= \begin{cases} +x_{ij,t} p_{ij,k} (1 - p_{ij,k}), & y_{ij} = 1 \\ -x_{ij,t} p_{ij,k} (1 - p_{ij,k}), & y_{ij} = 0 \end{cases} \\
&= (2y_{ij} - 1) x_{ij,t} p_{ij,k} (1 - p_{ij,k})
\end{aligned} \tag{3.41}$$

By using (3.41) in (3.34),

$$\frac{\partial}{\partial \theta_{k,t}} (A) = (2y_{ij} - 1) \alpha_k p_{ij,k} (1 - p_{ij,k}) x_{ij,t} \tag{3.42}$$

Recall that gradient of the incomplete data log likelihood is the following:

$$\nabla_{\theta} \ell_{ic} = \sum_{i=1}^N \sum_{j=i+1}^N \frac{1}{A} \frac{\partial}{\partial \theta_{k,t}} (A)$$

Hence the simplified version of the gradient of the incomplete data log likelihood is,

$$\nabla_{\theta} \ell_{ic} = \sum_{i=1}^N \sum_{j=i+1}^N \frac{1}{\sum_k \alpha_k P_k[Y_{ij} = y_{ij} | \mathbf{X}_{ij}; \theta_k]} (2y_{ij} - 1) \alpha_k p_{ij,k} (1 - p_{ij,k}) x_{ij,t} \tag{3.43}$$

Finally, $\theta^{(n+1)}$ becomes,

$$\theta_{k,t}^{(n+1)} = \theta_{k,t}^n + \mu \nabla_{\theta} \ell_{ic} \tag{3.44}$$

3.4 Evaluation Methods

Previous sections of Chapter (3) have introduced the dataset, features and the proposed model. This section covers how the performance of the model is measured. Specifically, the way links are predicted is explained. Since true positive rate is the main evaluation metric that we use in this work, a brief background of it will also be given.

As mentioned, we use mixture of logistic regression model to approach our model. Since there is no close form solution for the coefficients of the model, we use EM algorithm, and iteratively learn the coefficients. These coefficients are randomly initialized to observe how the model behaves for different initial parameters. The parameter initialization process is as follows: we first initialize the mixing coefficients, α , such that they sum to 1. Then, we randomly initialize θ such that each one has the value in the range of $[10^{-7}, 10^{-5}]$.

After finding the parameters, we compute the value of the mixing logistic regression using equation (3.8). In order to evaluate the performance, we use n predictions. In this study the range of n varies between 1 and 5000. To get these predictions, we rank the probabilities in descending order, and choose the the first n of these.

Having chosen the predictions, the next step is to label these probabilities. In logistic regression applications, the outcome is 1 if the probability is greater than or equal to 0.5, and 0 otherwise. Since the probabilities in our case are small, however, we define our own threshold. We set the proportion of 1s in the original class label as the threshold. If we define the class label 1s and 0s as y_1 and y_0 , the threshold, thr , can be expressed as follows:

$$thr = \frac{y_1}{y_1 + y_0} \quad (3.45)$$

In this work, the label 1s are predicted if the mixture probability is greater than the threshold defined above. By the same token, if the probability is less than the threshold, the model labels it as 0. By letting y' is the predicted label, we can write the following:

$$y' = \begin{cases} 1, & P[Y = y|\mathbf{X}; \Lambda] \geq thr \\ 0, & P[Y = y|\mathbf{X}; \Lambda] < thr \end{cases} \quad (3.46)$$

As a final step, we evaluate the relative performance of the first n predictions. This is defined as the true positive rate of the first n predictions divided by the true positive rate of random guessing of these predictions. Letting RP and TPR as the relative performance and the true positive rate, we can elaborate this definition as follows:

$$RP_n = \frac{TPR_n}{TPR_{rand_n}} \quad (3.47)$$

where TPR_n is the true positive rate of the first n predictions, and TPR_{rand_n} is the true positive rate of the random guessing of the first n predictions of the class label data. Since our main evaluation metric is the true positive rate [17], let us introduce the fundamentals of this concept.

Contingency table is given in Table (3.4). True positive rate, TPR , is computed as number of

Table 3.4: Contingency Table

		Original Labels	
		1	0
Predicted Labels	1	True Positives	False Positives
	0	False Negatives	True Negatives

true positive divided by total number of original positives. As can be seen from Table (3.4), total number of the original positives is simply the summation of the true positives and false negatives. Hence, TPR is shown as:

$$\text{True Positive Rate (TPR)} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \quad (3.48)$$

We compute the relative performance of the classifiers by using equations (3.47) and (3.48). TPR_n is the number of true positives in the first n predictions divided by the the total number of positives in the n predictions. TPR_{rand_n} is simply the proportion of ones of the first n predictions of the class label data. We find the relative performance of the classifiers with varying components as described above. Then, we observe the performances and evaluate which one performs best.

In this chapter, we discussed the methods that we use to approach our problem. We introduced our data, features, proposed model and performance evaluation metrics, and explained in detail. In the next chapter, we will provide and evaluate the experimental results.

Experimental Results

In this section, we give the results of the experiments we performed. We first give the figures showing how the objective function vs EM iteration varies against different number of components and different initialization types. Then, the learned logistic regression and mixing coefficients will be provided. We finally give the performances of the different number of components.

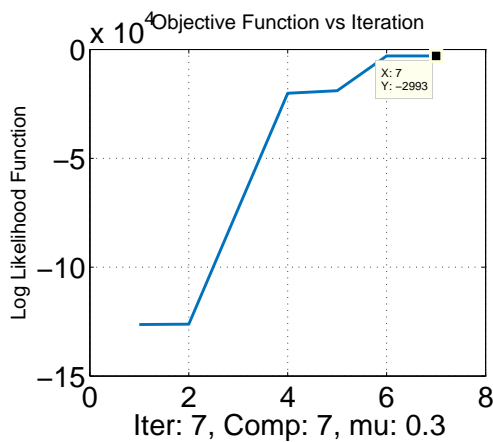
4.1 Optimization Results

We observed in our experiments that the way we initialize the parameters have an impact on convergence of the objective function. Figure (4.1) shows how the objective function behaves when we use distinct methods to initialize the parameters. Figure (4.1a) depicts that the objective function converges only after 6 iterations when we randomly initialize the parameters for the certain data. For the same data, in figure (4.1b), we see that the objective function does not significantly change after 16 iterations. Hence, the way that we initialize the parameters as described in section (3.4) provides faster convergence.

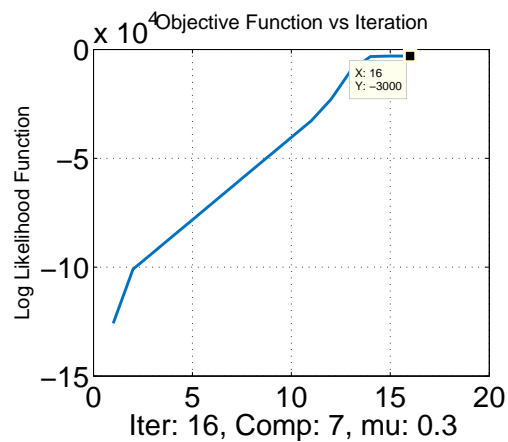
Figures (4.2) demonstrates how the classifier C1 performs for different number of components. (4.2a),(4.2b),(4.2c) and (4.2d) are the optimization results of the log-likelihood function when the number of components are 1,2,3 and 7, respectively. In theory, we expect the log likelihood function to improve as more component is used. As for the experiments, it is clear to see that the log likelihood in figures (4.2a), (4.2b) and is increasing as more component is mixed. After some point, however, this rise stops as given in figure (4.2d). Hence, we can reach the conclusion that using mixture model increases the likelihood.

Hence the following conclusions can be reached from above figures: First, log-likelihood improves with EM iterations. Second, values of the converged log-likelihood is higher when more mixture components are used.

Having maximized the objective function, the learned coefficients of the logistic regression are given in Table (4.1). As introduced before, parameter θ is the coefficients of the logistic regression.

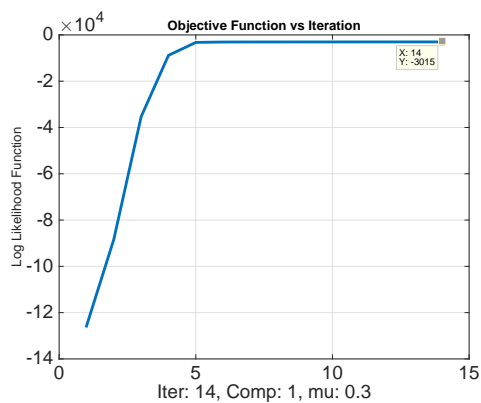


(a) Objective Function when Parameters are Randomly Initialized

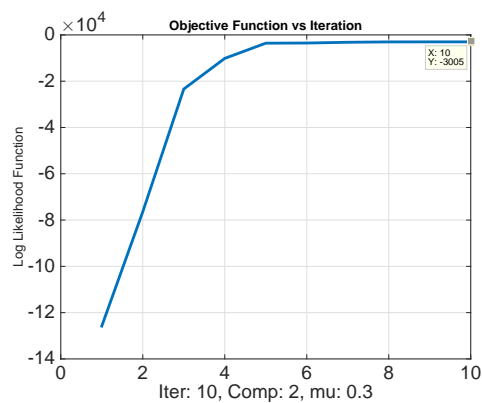


(b) Objective Function when Parameters are set to zero.

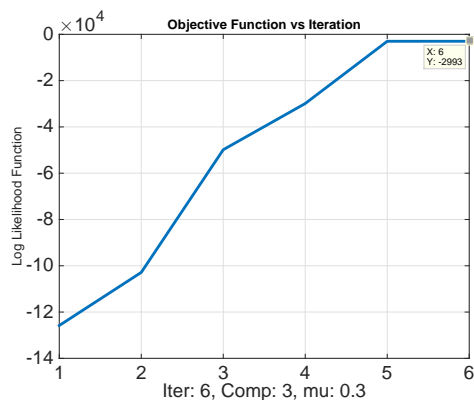
Figure 4.1: Objective Function with Different Initial Values



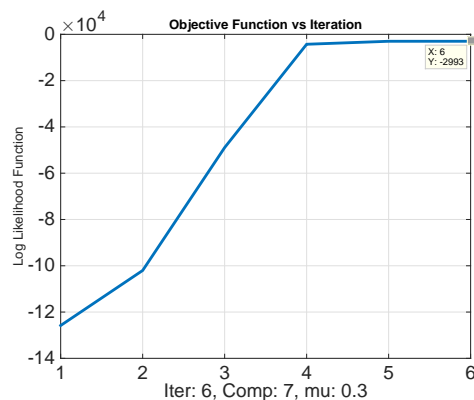
(a) $M=1$



(b) $M=2$



(c) $M=3$



(d) $M=7$

Figure 4.2: Objective Function of C1 with different number of components.

Table 4.1: Learned Coefficients of the Classifier C1 in the year 2003

Year = 2003	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6
M=1	0.05	0.08	0.28	0.05	0.10	0.30
M=2	0.25	0.46	0.93	0.22	0.46	0.95
	0.06	0.12	0.53	0.05	0.12	0.53
M=3	0.06	0.11	0.64	0.09	0.12	0.66
	2.53	4.75	9.54	2.53	4.70	8.62
	2.29	4.31	8.40	2.59	4.39	8.72
M=7	0.92	1.73	3.30	0.82	1.75	3.51
	1.16	2.18	4.27	1.26	2.08	4.40
	1.09	2.04	4.09	1.09	2.05	4.23
	0.53	1.00	2.00	0.53	1.10	2.42
	0.27	0.51	1.01	0.22	0.51	1.22
	0.92	1.72	3.45	0.95	1.72	3.48
	0.01	0.05	0.28	0.01	0.04	0.38

In Table (4.1), $\theta_1 - \theta_4$, $\theta_2 - \theta_5$ and $\theta_3 - \theta_6$ are the learned coefficients of the years 2000, 2001 and 2002, respectively. Recall that we re-scaled the feature matrix so that the learned coefficients would quantitatively tell which feature is more significant. By evaluating the coefficients in Table (4.1), thus, we can clearly say that magnitude of $\theta_3 - \theta_6$ are the greatest among any of the thetas. This means that features belong to the year 2002 contributed most when predicting the new links in the year 2003.

Table 4.2: Mixing Coefficients of the Classifier C1 in the year 2003

Year : 2003	α_1	α_2	α_3	α_4	α_5	α_6	α_7
M=1	1.00	-	-	-	-	-	-
M=2	0.80	0.20	-	-	-	-	-
M=3	0.00	0.52	0.47	-	-	-	-
M=7	0.19	0.24	0.22	0.11	0.05	0.19	0.00

Table (4.2) shows the updated mixing coefficients of the classifier C1 in the year 2003. These mixing coefficients can be considered as weights. α_7 when $M = 7$, for instance, is 0, which tells us that seventh component does not effect on the performance of the model.

Similar to C1, the learned coefficients of the classifiers C2, C3 and C4 are given in Table (4.3).

As for the values given in table (4.1), learned coefficients of the classifier C2, $\theta_3 - \theta_6$, are the ones that are the largest. This indicates that the parameters that belong to the year 2002 are more informative than those belong to 2000 and 2001.

A question can be asked at this point that the parameters have a symmetrical pattern. For example, in table (4.3), $\theta_3 - \theta_6$, $\theta_2 - \theta_5$ and $\theta_1 - \theta_4$ themselves are similar or sometimes even equal to each other. This stems from the fact that we use features $x_3 - x_6$, $x_2 - x_5$ and $x_1 - x_4$ to find the coefficients. From Table (A.13), features of each year is generated by simply summing

Table 4.3: Learned coefficients of classifiers C2,C3 and C4 in the year of 2003

Year: 2003	C2		C3		C4	
	θ_3	θ_6	θ_2	θ_5	θ_1	θ_4
M=1	3.63	3.93	1.07	1.07	1.07	1.02
M=2	2.21	2.44	1.08	1.08	0.99	0.91
	1.76	1.95	0.25	0.25	0.73	0.71
M=3	1.22	1.35	1.05	1.05	0.27	0.26
	1.58	1.76	0.27	0.27	1.83	1.78
	1.85	2.06	2.08	2.08	1.66	1.50
M=7	2.00	2.35	0.27	0.27	1.57	1.51
	2.48	2.99	0.47	0.47	1.09	1.09
	2.50	3.02	0.16	0.15	1.48	1.47
	0.42	0.45	0.56	0.54	1.20	1.16
	1.93	2.25	0.19	0.12	1.62	1.45
	1.86	2.16	0.67	0.66	1.62	1.44
	2.47	2.98	0.22	0.29	0.19	0.39

Table 4.4: Mixing coefficients of the classifiers C2,C3 and C4 in the year 2003.

Year: 2003		α_1	α_2	α_3	α_4	α_5	α_6	α_7
M=1	C_2	1.00	-	-	-	-	-	-
	C_3	1.00	-	-	-	-	-	-
	C_4	1.00	-	-	-	-	-	-
M=2	C_2	0.66	0.34	-	-	-	-	-
	C_3	1.00	0.00	-	-	-	-	-
	C_4	0.78	0.22	-	-	-	-	-
M=3	C_2	0.35	0.17	0.48	-	-	-	-
	C_3	0.40	0.43	0.17	-	-	-	-
	C_4	0.49	0.18	0.33	-	-	-	-
M=7	C_2	0.24	0.07	0.27	0.10	0.07	0.07	0.18
	C_3	0.07	0.28	0.11	0.19	0.06	0.21	0.09
	C_4	0.09	0.28	0.15	0.32	0.06	0.09	0.00

the number of co-authored papers between $i - k$ and $j - k$ where k denotes the mutual nodes of i and j pairs. Since the features generated using $i - k$ and $j - k$ pairs have similar characteristics, learned coefficients are also proportional to each other.

4.2 Relative Performance of the Classifiers

For testing, we use the data that belongs to the years between 2004 and 2007. Figures (4.3), (4.4), (4.5) and (4.6) demonstrate how the classifiers C_1, C_2, C_3, C_4 behave for different number of components and data sets. In this section, we will discuss the performance of each classifier

for varying component numbers and years. Then, we will compare the relative performance of the classifiers to each other to find out which one achieves best.

4.2.1 Relative Performance of C_1

In 2004s data, we see that the performance increases as the number of components increase. C_1 performs worst for the single component. When we mix 4 components together, the performance is better than the one for single component. When we use 7 components, moreover, the performance is even better. In this dataset, hence, we observe that the relative performance improves when more components are mixed.

C_1 performs almost the same for different component numbers we tried using 2005s data. For the first 950 predictions the performance is the same, while $C_{1M=7}$ goes down after this point.

For the network of the year 2006, the performance does not always improve when the number of components increases. For the first 350 predictions, single component gives us a better result. Mixing more components in the range between 350 and 600 predictions provides more desired results. After the first 600 predictions, the performance of $C_{1M=7}$ is getting worse than the one for single and 4 components.

When it comes to 2007s data, $C_{1M=1}$ and $C_{1M=4}$ have the similar performance, while $C_{1M=7}$ provides the best result.

4.2.2 Relative Performance of C_2

We observe in our experiments that performance of C_2 does not quite change with different number of components. In 2004, 2006 and 2007 data, $C_{2M=1}$, $C_{2M=4}$ and $C_{2M=7}$ have the similar performance results. In 2007, the performance of $C_{2M=7}$ improves after the first 900 predictions, while it performs worse than the others in 2005. Apart from these two statements, the relative performances have similar patterns.

4.2.3 Relative Performance of C_3

The performance of C_3 has some similar patterns for different number of components. In 2004, for instance, we observe similar results for different component number for the first 800 predictions. After this point, however, mixing 7 components outperformed the others.

In 2005, $C_{3M=7}$ and $C_{3M=4}$ perform similarly, while $C_{3M=7}$ provides the best performance. In 2006, we observe similar patterns for different components. In 2007, we also have almost identical performances for the first 550 predictions. Using single component after this point gives better result than using mixture of the components.

4.2.4 Relative Performance of C_4

We observe similar performances in years 2004, 2005 and 2006. In 2004, the performance of C_4 when 7 components are mixed is slightly better than the other. In 2007, we clearly see that

$C_{4M=7}$ outperforms the others.

As a conclusion, we can point out that when using mixture of components, the results are at least same as or better than using single component. This is expected since we know from theory that mixing components should improve the performance.

4.2.5 Comparison of the Relative Performance of the Classifiers

In section (4.2), we mainly discuss how each classifier behaves when different number of components and data sets are used. We now turn our attention to compare the performance of each classifier to others. Again, we do so by using the networks of years 2004, 2005, 2006 and 2007.

Performance of each classifier for a single component is given in figure (4.7). Figures (4.7b), (4.7c) and (4.7d) show that the classifiers including younger connections perform better than those with older links in years 2005,2006 and 2007. In 2004, on the other hand, we see that old and mid-aged links dominate the link prediction process (Figure (4.7a)).

When we mix more components to allow multiple link formation profiles, we observe in figures (4.8) and (4.9) that the performance of each classifier does improve. In 2005,2006 and 2007, for instance, the performance of young and old links have better performance, but younger connections are still better than older ones in predicting the following year's link formations. In 2004's data that is given in figures (4.8a) and (4.8a), on the other hand, oldest links are still strong, especially for the first 200 predictions.

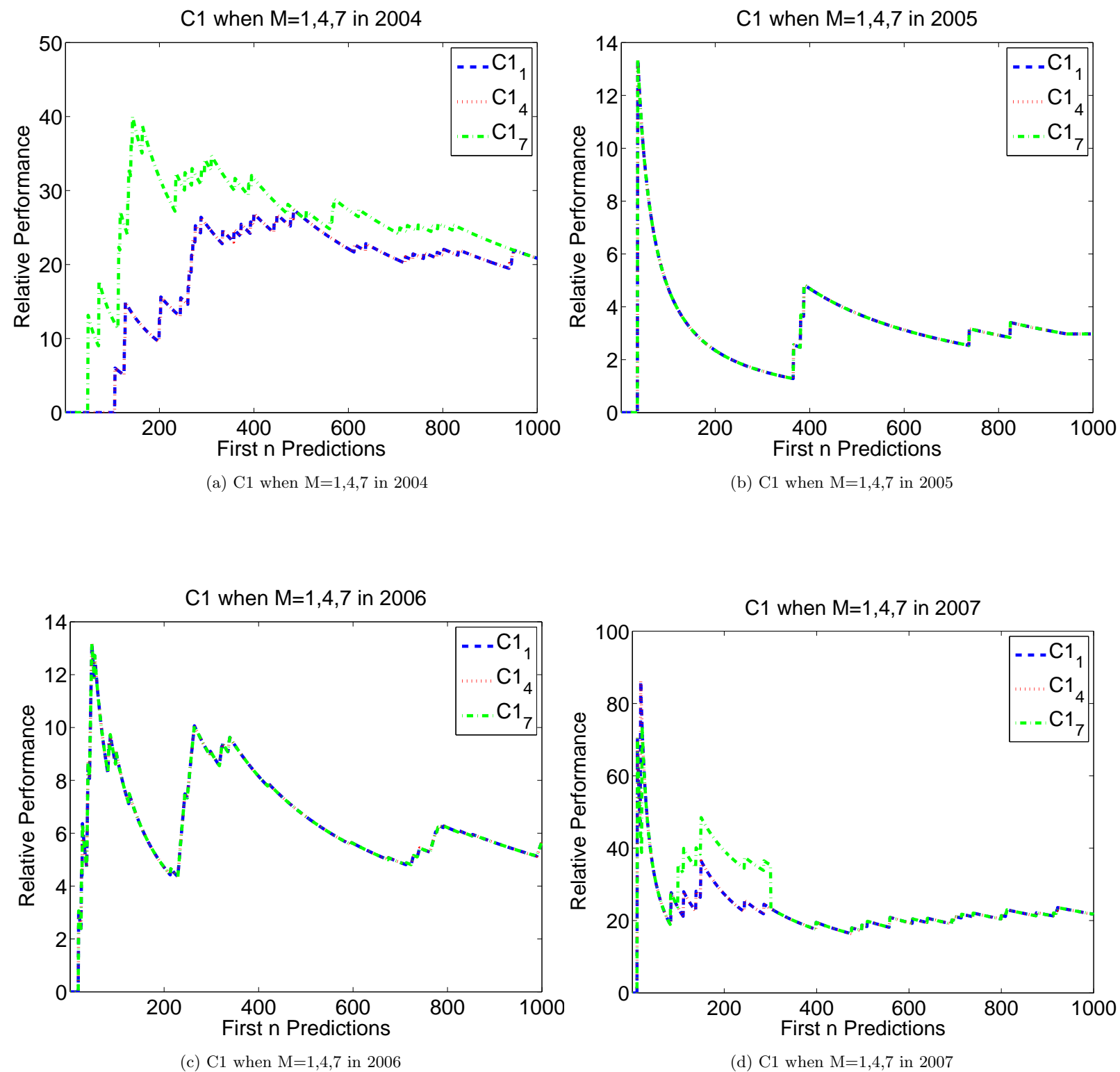


Figure 4.3: Relative Performance of C1 with different number of components in the range between 2004 and 2007

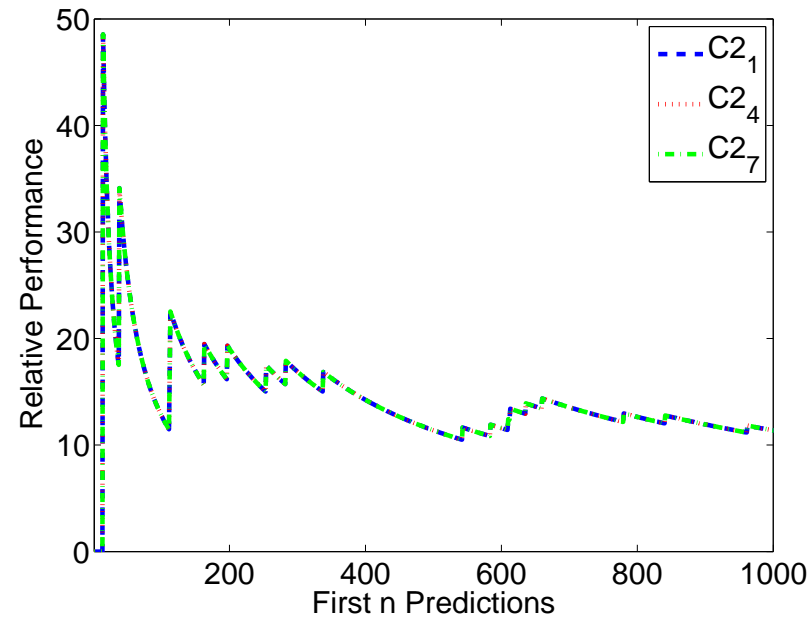
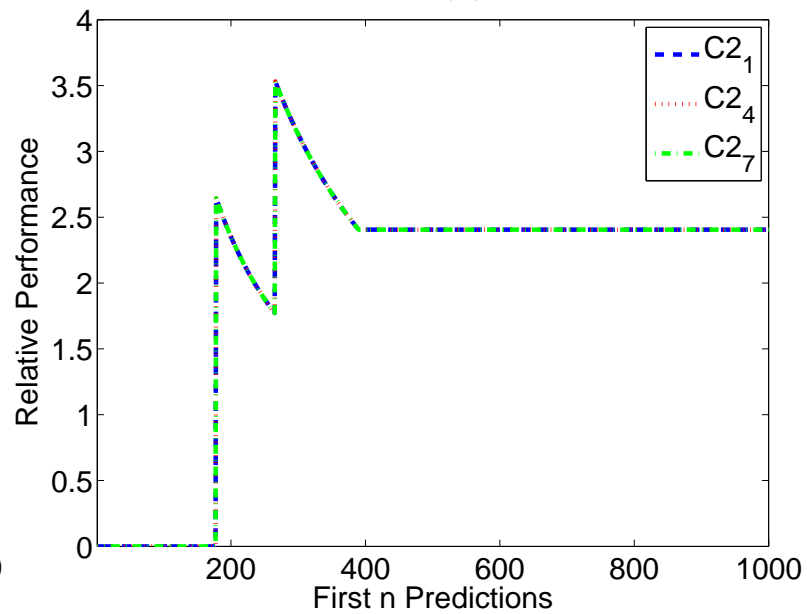
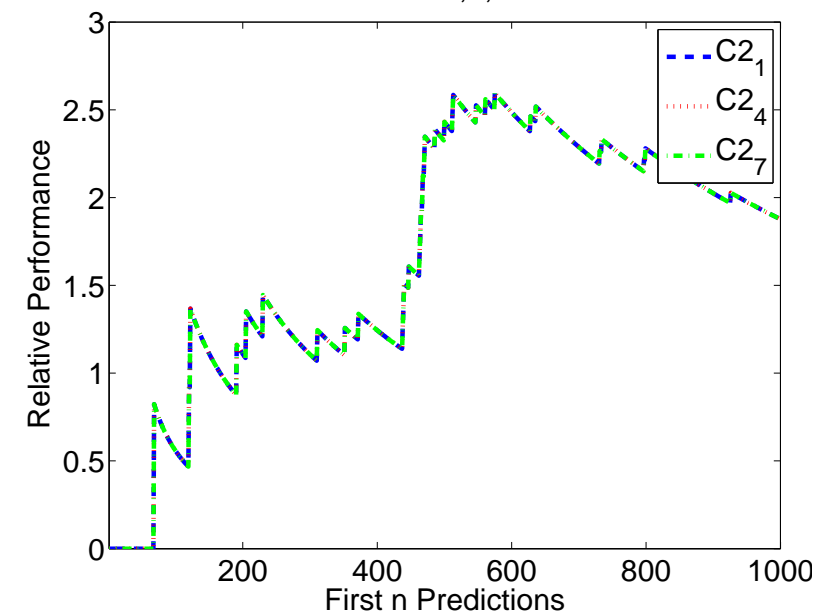
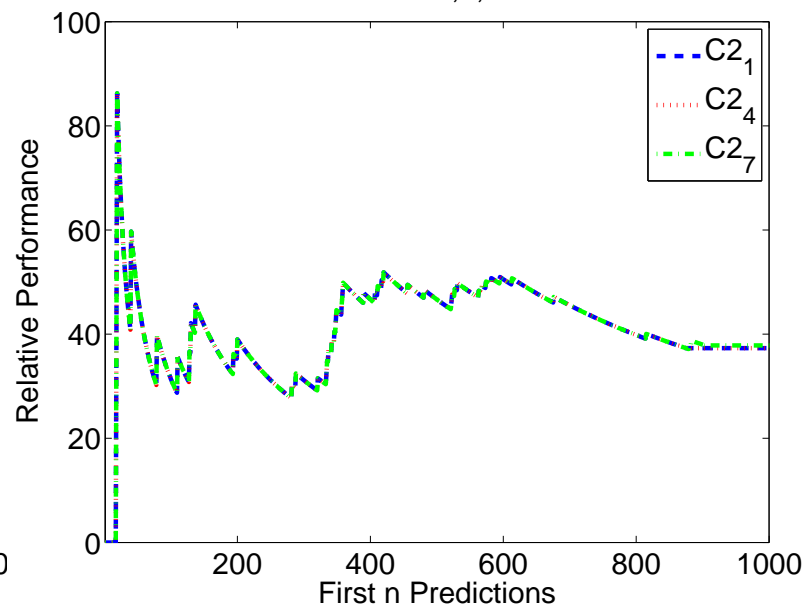
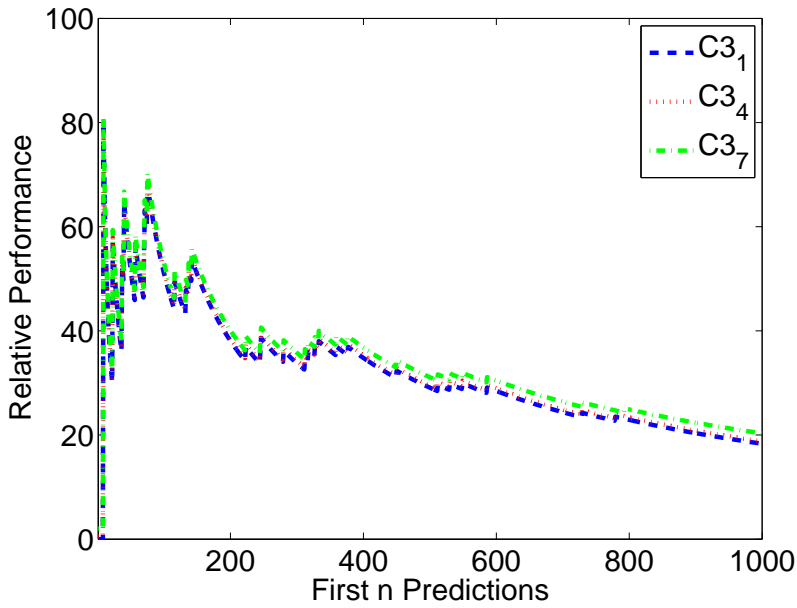
C2 when $M=1,4,7$ in 2004(a) C2 when $M=1,4,7$ in 2004C2 when $M=1,4,7$ in 2005(b) C2 when $M=1,4,7$ in 2005C2 when $M=1,4,7$ in 2006(c) C2 when $M=1,4,7$ in 2006C2 when $M=1,4,7$ in 2007(d) C2 when $M=1,4,7$ in 2007

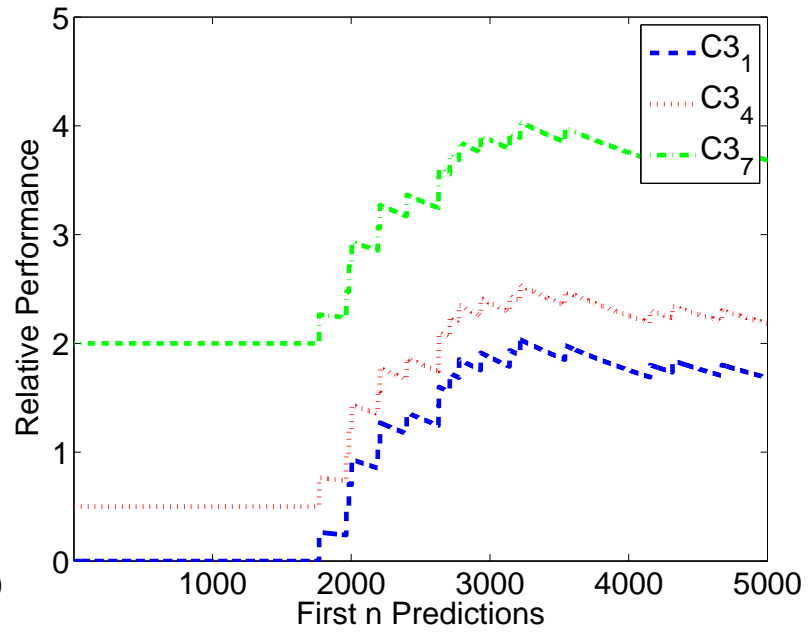
Figure 4.4: Relative Performance of C2 with different number of components in the range between 2004 and 2007

C3 when M=1,4,7 in 2004



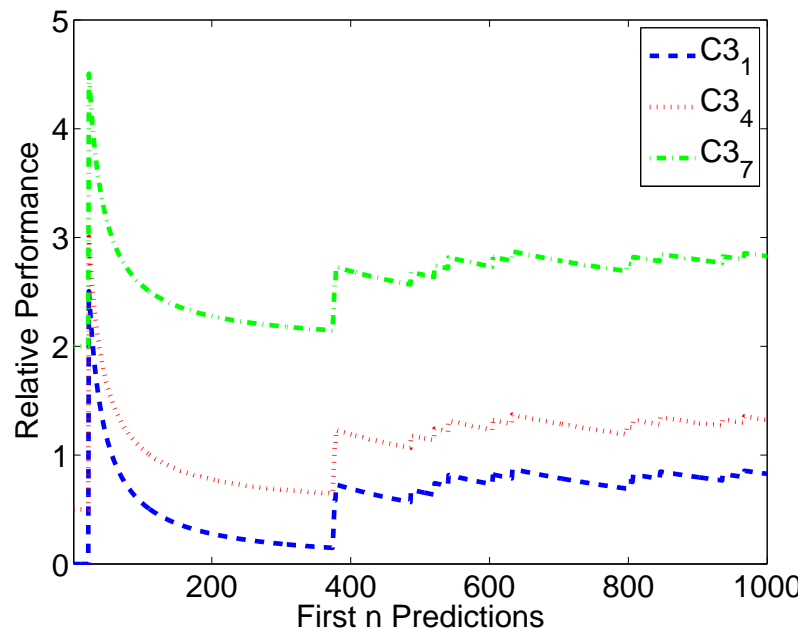
(a) C3 when M=1,4,7 in 2004

C3 when M=1,4,7 in 2005



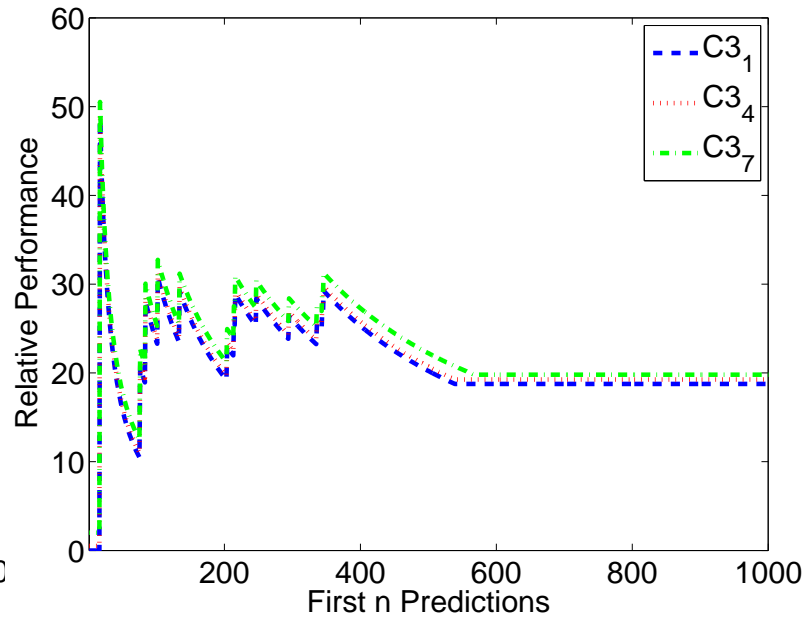
(b) C3 when M=1,4,7 in 2005

C3 when M=1,4,7 in 2006



(c) C3 when M=1,4,7 in 2006

C3 when M=1,4,7 in 2007



(d) C3 when M=1,4,7 in 2007

Figure 4.5: Relative Performance of C3 with different number of components in the range between 2004 and 2007

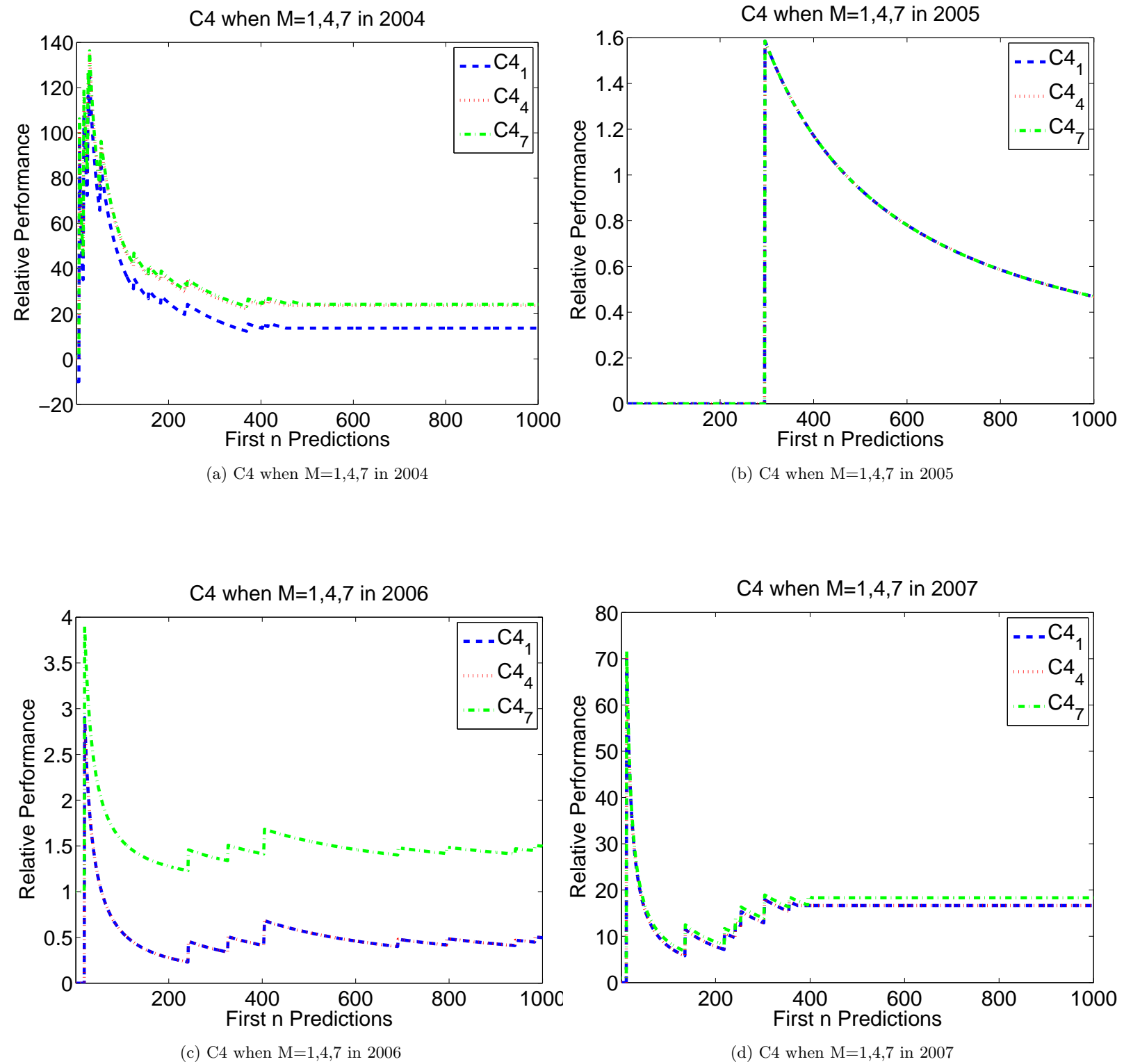
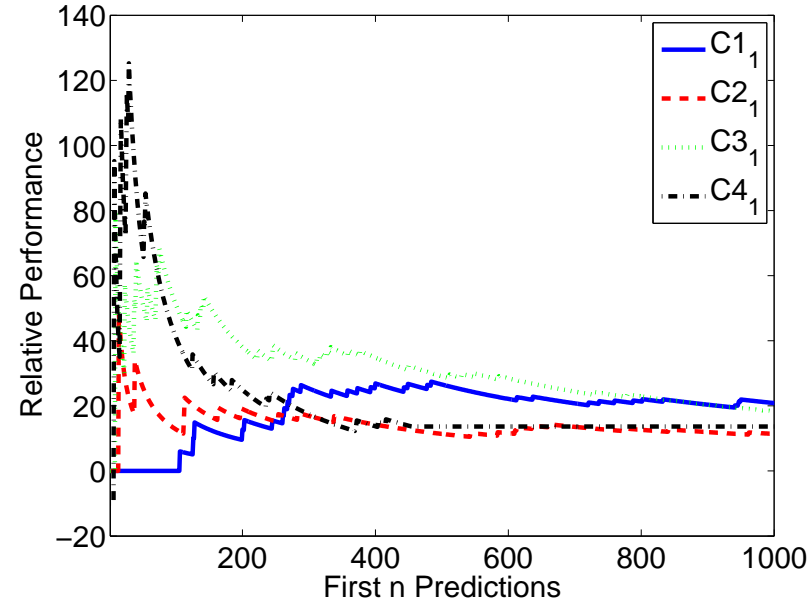


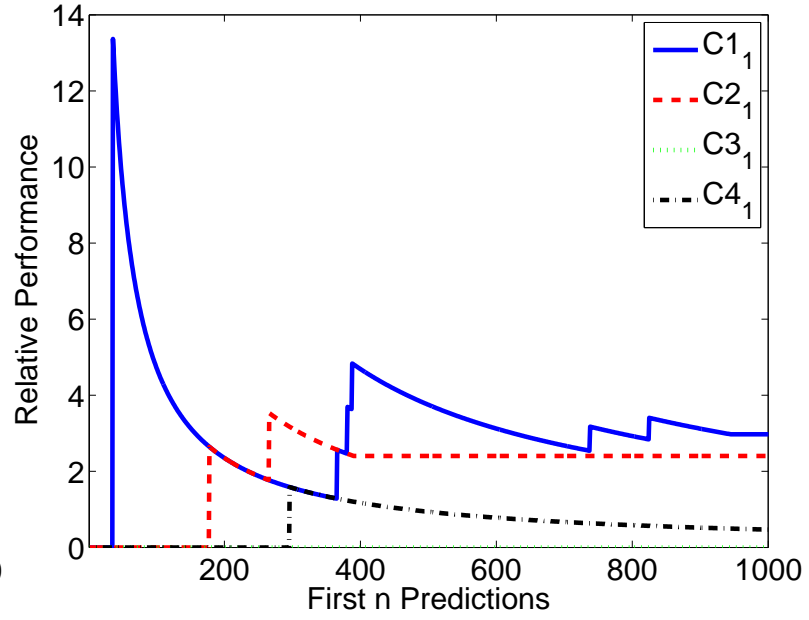
Figure 4.6: Relative Performance of C4 with different number of components in the range between 2004 and 2007

C1,C2,C3,C4 when M=1 in 2004



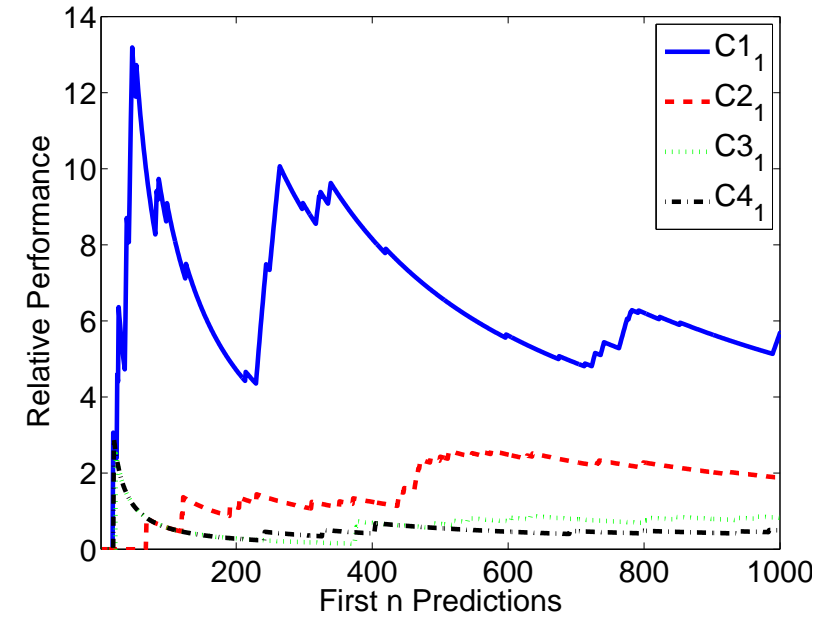
(a) C1,C2,C3,C4 M=1 in 2004

C1,C2,C3,C4 when M=1 in 2005



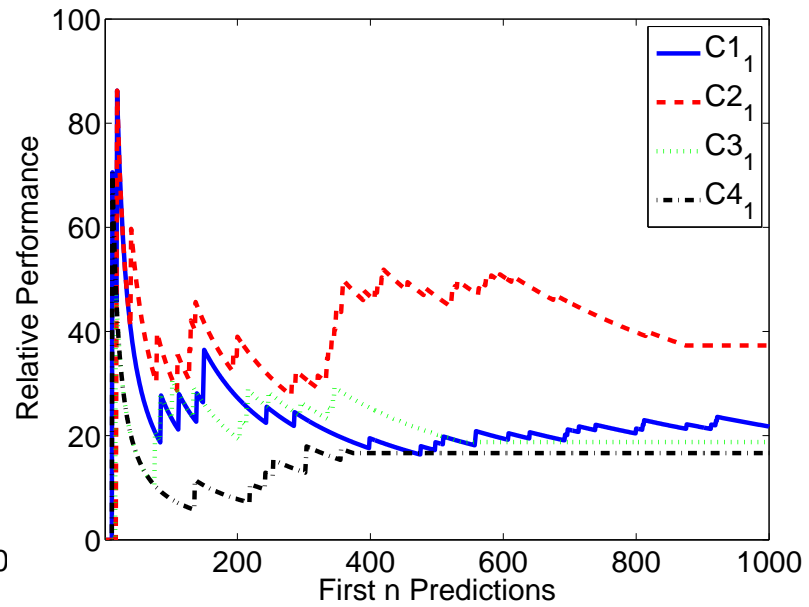
(b) C1,C2,C3,C4 M=1 in 2005

C1,C2,C3,C4 when M=1 in 2006



(c) C1,C2,C3,C4 M=1 in 2006

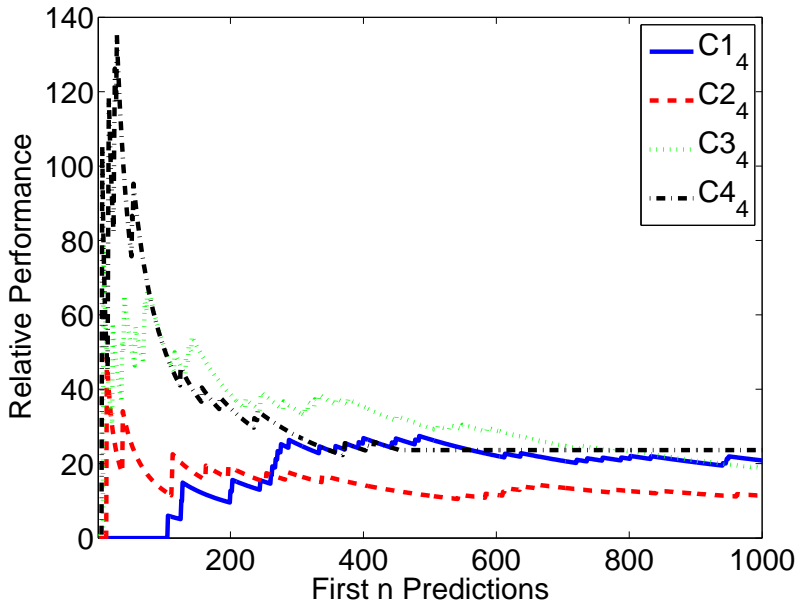
C1,C2,C3,C4 when M=1 in 2007



(d) C1,C2,C3,C4 M=1 in 2007

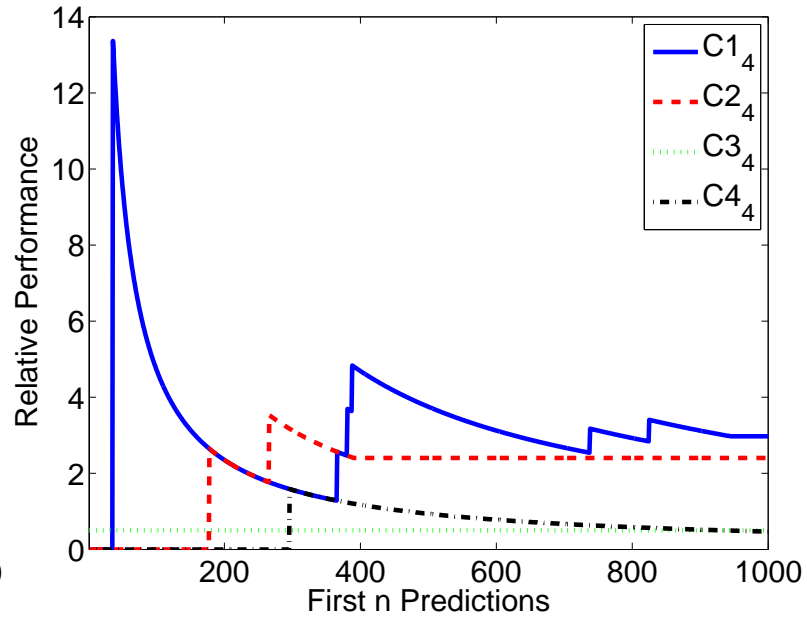
Figure 4.7: Relative Performance of C1,C2,C3,C4 when M=1 in the range between 2004 and 2007

C1,C2,C3,C4 when M=4 in 2004



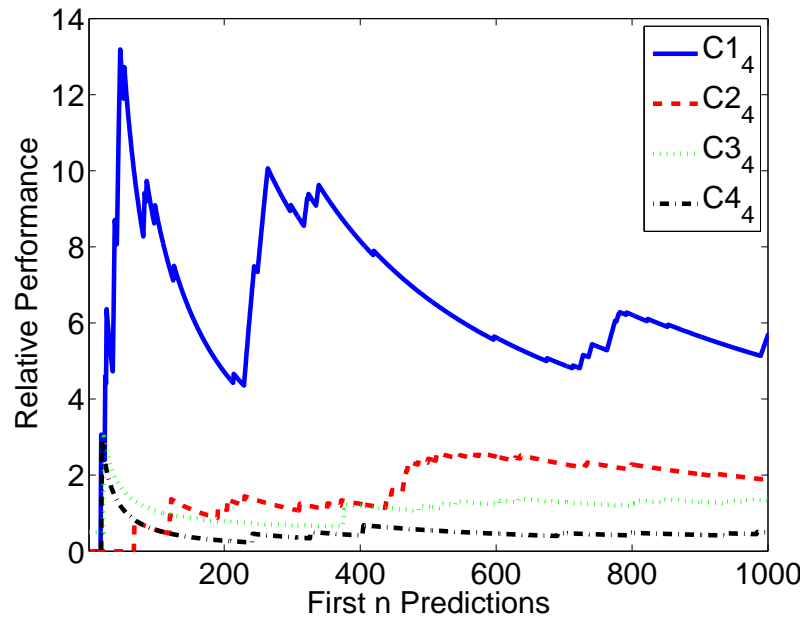
(a) C1,C2,C3,C4 M=4 in 2004

C1,C2,C3,C4 when M=4 in 2005



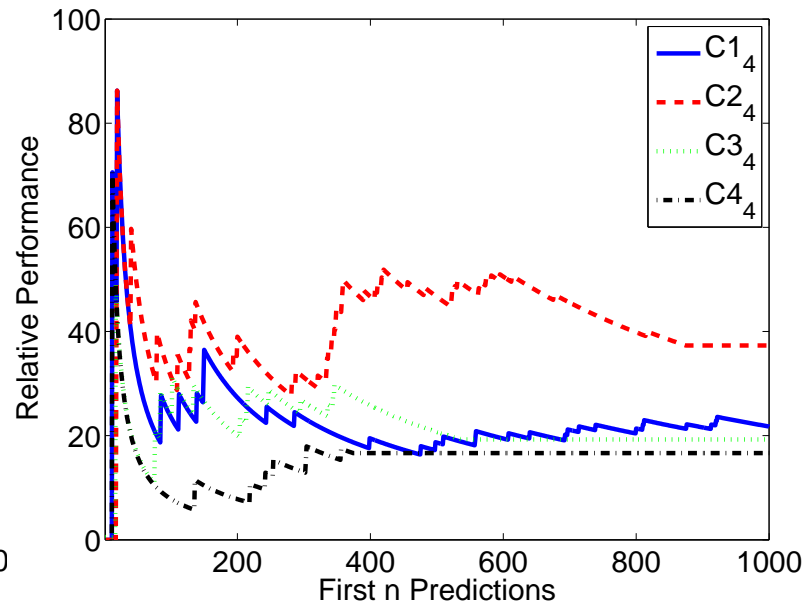
(b) C1,C2,C3,C4 M=4 in 2005

C1,C2,C3,C4 when M=4 in 2006



(c) C1,C2,C3,C4 M=4 in 2006

C1,C2,C3,C4 when M=4 in 2007



(d) C1,C2,C3,C4 M=4 in 2007

Figure 4.8: Relative Performance of C1,C2,C3,C4 when M=4 in the range between 2004 and 2007

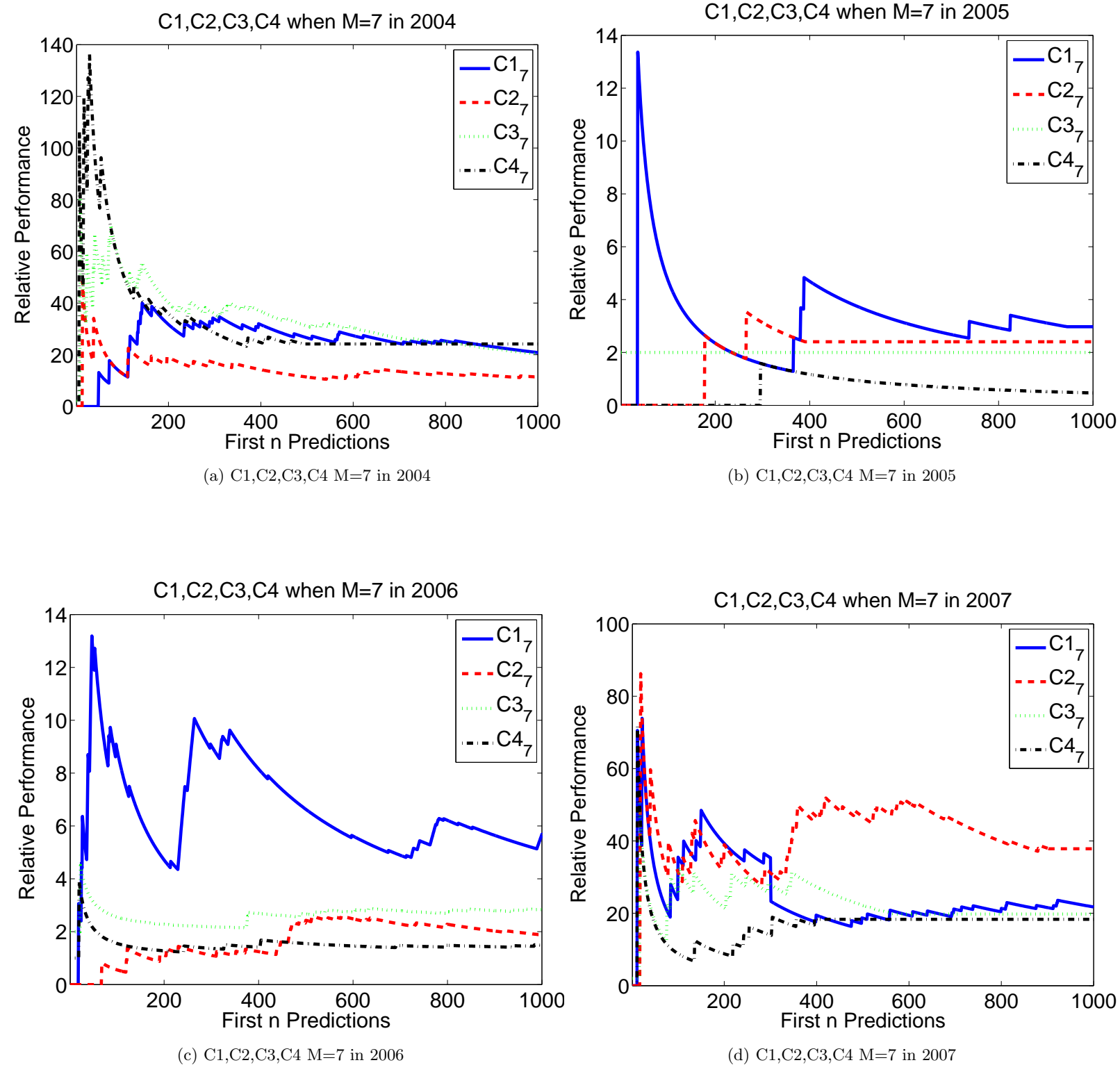


Figure 4.9: Relative Performance of C1,C2,C3,C4 when M=7 in the range between 2004 and 2007

Conclusion, Discussion and Future Work

In this chapter, we summarize the main points highlighted in previous parts of this thesis, and discuss the main findings, contributions and future work. Section (5.1) gives a summary of the thesis. Section (5.2) discusses the main findings and how these contribute to the network evolution theory. Then, section (5.3) introduces the possible future work extending what has been done in this work. Finally, section (5.4) concludes the thesis.

5.1 Summary of the Thesis

This thesis has introduced a mixture of logistic regression model to observe how aging could be an important factor that might adversely affects the prediction power of the links in a network.

Chapter 2 reviewed some of the existing work that is widely known in network theory. Much of the current work is based on network generation models that capture some of the common popular statistical properties such as high clustering coefficient, scale-free power law distribution. Network evolution is also an attractive field. Scientists have observed many common properties such as densification power law, heavy tailed degree. Since networks evolve over time, many link prediction algorithms have also been proposed. For instance, the link prediction problem has been approached using supervised learning methods, and highly accurate results have been obtained. Even though above fields have been hot topic for years, link's age evaluation in network evolution has not gained sufficient attention.

Chapter 3 describes the experimental setup of this work. More specifically, DBLP data-set is used to create the co-authorship network. Using similarity measures in this network, features are generated. Class labels are derived from the network according to the presence or absence of the links. Chapter 3 also introduces the model used. Logistic regression model is introduced, and its mixture extension is derived. Each part of this derivation process is thoroughly described.

Since the estimation of the coefficients of the logistic regression model do not have a closed form solution, expectation maximization algorithm, a widely known iterative method, is introduced. The parameter optimization procedure is also explained.

Chapter 4 provides the experimental results. Training results are given first. In training results section, performance of EM algorithm is discussed. Since initialization of the parameters does have an impact on the performance, many experiments with different initialized parameters are performed, and the results are provided. Then, the way log-likelihood function behaves against each iteration is observed, and the results are given. Finally, results of the learned coefficients for different number of components are provided. For the test set, relative performance of each classifiers with varying components are given. Then performance of each classifier is shown in the same figure for comparison.

5.2 Discussion

This work evaluates the age of the links in link prediction, and considers aging as a factor that influences network evolution. In our experiments, we observe that relative performance of young links are generally higher than old links. This means that young links' contribution to the link prediction process is higher than old connections in many cases. Thinking that there could be reliable old links that are significant, we use mixture components and performed the experiments. And, we observe in our experiments that in some cases, old connections are still informative. Apart from these old but significant links, younger connections are more powerful in predicting following year's link formations. Another observation is that performance of the connections do improve when more components are mixed. These lead to a conclusion that we could disregard some of the old and uninformative connections when predicting new links since they do not provide much information. Hence, we could also state that in network evolution theory, node-link removal should also be taken into account as [1] also proposed.

5.3 Future Work

Our link prediction process in this study is as follows: we generate features of the co-authorship network using three consecutive years' information, and predict the following years' link connections. We restrict ourselves to three consecutive years because [1] reveals that collaboration period of authors decreases after the third year. Even though [1] observes that even in starring, co-starring behavior also diminishes after 3 years, further research could be conducted to find out whether this period differs in any other network.

The second question mark is that we do not know how exactly to remove the nodes or links from the network. A proper procedure is still needed. After finding a way to handle this, the networks with/out old links could be analyzed and compared in terms of their statistical properties.

5.4 Conclusion

This work proposes a model to observe age of links as an effect in link prediction. Our experiments demonstrate that as a link in a network becomes mature, its effect on link prediction generally worsens. This means that some of the links could become inactive over time. A conclusion can be reached at this point that link removal from network might be a possible action to maintain the stability of a network.

Feature Extraction

In this section, we discuss the data we use, how we process it to generate the co-authorship network, and the methods to extract the feature matrix. We will first start by introducing the fundamentals of the data-set.

1 Fundamentals of the Data

To create the co-authorship network, we use DBLP Computer Science Bibliography data set, which provides us with detailed list of research papers published in the field of computer science. Table (A.1) shows some of the properties of the data.

Table A.1: Data Characteristics

Number of Nodes	<i>178154</i>
Number of Edges	<i>3,621,265</i>

Table (A.2) represents a sample of the data. As seen from the table, it has three columns: *from node*, *year*, *to node*. These columns represent author id, year of the publication and the coauthor id, respectively. The main advantage of this data set is that we can generate a co-authorship network, sort it with respect to years, and investigate how the influence of links in predicting new formations change over time. In order to find this effect, we first create the co-authorship network, then find the mutual friends of author pairs, and finally extract the features.

2 Generating the Co-Author Network

In this section, we explain how we generate the co-authorship network. More specifically, we use the sample data, and show how to generate the network step by step. From the definition, we know that a co-author is the one who collaborates and publishes a paper with an author. To

From Node (Author ID)	Year	To Node (Paper ID)
000001	2000	55
000001	2000	56
000001	2000	58
000001	2004	65
000001	2005	66
000002	2004	99
000003	2000	55
000003	2000	56
000003	2000	57
000003	2000	58
000003	2006	78
000003	2004	79
000004	1958	1
000005	2000	55
000005	2000	56
000005	2000	57
000005	2000	58
000005	2000	59
000005	2005	78
\vdots	\vdots	\vdots
881512	2005	279197
881514	1981	008540

Table A.2: A Sample data

find the co-authors in our case, hence, we simply compare the paper IDs given in the data. Those whose paper IDs are the same are the co-authors of each other. Using this notion, we generate the co-authorship network. In order to make the network even stronger, we delete the authors and co-authors with less than 5 papers [1]. For the sample data we give above, for instance, we delete the following authors since they do not have at least 5 papers.

000002	2004	99
000004	1958	1
881512	2005	279197
881514	1981	008540

Table A.3: Sample data: Authors who do not have at least 5 papers.

In so doing, we get rid of the authors 2, 4, 881512 and 881514 in table (A.2). Having deleted these nodes, we get table (A.4).

We now find the co-authorship network of the year 2000. To do so, we filter the sample data with respect to year 2000, and get the table (A.5) and figure (A.1).

Recall that we find the co-authors by comparing paper IDs. Authors with same paper id mean they publish a paper together. To make this more meaningful, let us simplify the network given in figure (A.1) so that we only focus on the papers that author 1 publishes.

From Node (Author ID)	Year	To Node (Paper ID)
000001	2000	55
000001	2000	56
000001	2000	58
000001	2004	65
000001	2005	66
000003	2000	55
000003	2000	56
000003	2000	57
000003	2000	58
000003	2006	78
000003	2004	79
000005	2000	55
000005	2000	56
000005	2000	57
000005	2000	58
000005	2000	59
000005	2005	78
\vdots	\vdots	\vdots

Table A.4: Sample data: Authors with at least 5 papers.

Figure A.1: Data visualization of table (A.5)

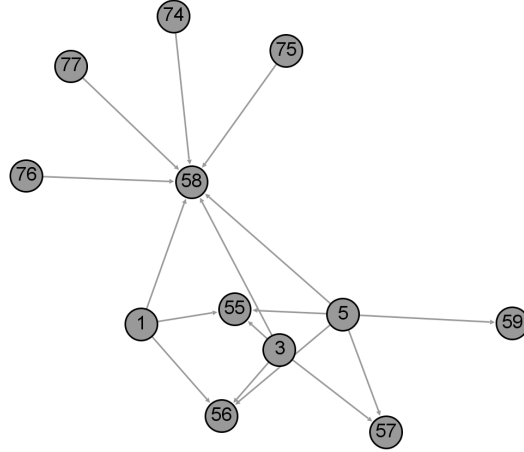


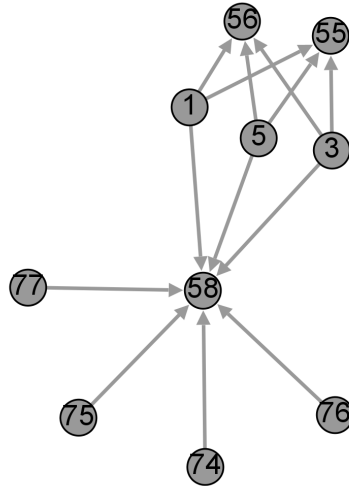
Figure (A.2) depicts that author 1 published papers 55,56 and 58. For the paper 58, we see that authors 3,5,74,75,76 and 77 contributes the paper. By the same token, 1,3,5 write papers 55 and 56. Hence, we reach the conclusion that 3, 5, 74, 75, 76, 77 are the co-authors of 1 in year 2000. Using this notion, we find the complete network of year 2000.

Recall that the main aim of this study is to observe how link age impacts the prediction

From Node (Author ID)	Year	To Node (Paper ID)
000001	2000	55
000001	2000	56
000001	2000	58
000003	2000	55
000003	2000	56
000003	2000	57
000003	2000	58
000005	2000	55
000005	2000	56
000005	2000	57
000005	2000	58
000005	2000	59
000074	2000	58
000075	2000	58
000076	2000	58
000077	2000	58
\vdots	\vdots	\vdots

Table A.5: Sample data of the year 2000.

Figure A.2: Author #1's network in year 2000.



power. In order to find the relationship, we need to compare the networks of different years. [1] proves that the average co-authoring period is no longer than 3 years. Hence, we will consider this time span when comparing the networks.

We have already discussed how to generate a co-authorship network and given the network of year 2000 as an example. Since we restrict the co-authoring period to 3 years, we also need to find the networks of years 2001 and 2002. We will then compare these networks to the year 2003 to observe how they influence the new connections in 2003. Since we already discuss network

Table A.6: Co-Authors of the sample data of year 2000

Author ID	Co-Authors					
1	3	5	74	75	76	77
3	1	5	74	75	76	77
5	1	3	74	75	76	77
74	1	3	5	75	76	77
75	1	3	5	74	76	77
76	1	3	5	74	75	77
77	1	3	5	74	75	76

generation process, we only give the results for 2001,2002 and 2003 networks.

Table A.7: Co-Authors of the sample data of year 2001

Author ID	Co-Authors		
3	5	75	76
4	6	0	0
5	3	75	76
6	4	0	0
75	3	5	76
76	3	5	75

Table A.8: Co-Authors of the sample data of year 2002

Author ID	Co-Authors					
2	3	4	6	0	0	
3	2	4	6	72	76	
4	2	3	6	80	0	
6	2	3	4	0	0	
72	3	76	0	0	0	
76	3	76	0	0	0	
80	4	0	0	0	0	

2.1 Updating Co-Authorship Networks

Co-Author networks of years 2000,2001,2002 and 2003 are given in tables (A.6),(A.7),(A.8),(A.9), respectively. Let us define a source network N_1 that has the three consecutive years 2000,2001,2002 and a target network N_2 that has the co-authorship information of the year 2003. We use the connections in [2000,2002] to predict the new links in 2003. Note that there are distinct nodes either in N_1 and N_2 . To predict new links in N_2 , we need to use the same nodes to compare the links. To do so, we use the common nodes that both N_1 and N_2 have in common.

$$Node = Node_{N_1} \cap Node_{N_2} \quad (A.1)$$

Table A.9: Co-Authors of the sample data of year 2003

Author ID	Co-Authors				
2	4	76	83	0	0
4	2	5	76	77	83
5	4	76	77	0	0
76	2	4	5	77	0
77	4	5	76	79	0
79	77	0	0	0	0
83	2	4	0	0	0

In the above example, the unique nodes in N_1 and N_2 and the common nodes to train N_2 are;

$$Node_{N_1} = \{1, 2, 3, 4, 5, 6, 72, 73, 75, 76, 77, 80\}$$

$$Node_{N_2} = \{2, 4, 5, 76, 77, 79, 83\}$$

$$Node = \{2, 4, 5, 76, 77\}$$

The next step is that we ignore the edges that are connected in both N_1 and N_2 since we want to focus only on the formation of the new links. In so doing, the edges in N_2 become unique, which means that they are the new links in target network. After these two steps, we have the following networks for the sample data that we use as an example;

Table A.10: Co-Author Network of 2000 and 2001.

Co-Author Network of 2000						Co-Author Network of 2001					
Author ID	Co-Authors					Author ID	Co-Author				
2	0	0	0	0	0	2	0	0	0	0	0
4	0	0	0	0	0	4	6	0	0	0	0
5	1	3	75	76	77	5	3	75	76	0	0
76	1	3	5	75	77	76	3	5	75	0	0
77	1	3	5	75	76	77	0	0	0	0	0

Table A.11: Co-Author Network of 2002 and 2003.

Co-Author Network of 2002						Co-Author Network of 2003					
Author ID	Co-Authors					Author ID	Co-Author				
2	3	4	6	0	0	2	76	0	0	0	0
4	2	3	6	80	0	4	5	76	77	0	0
5	0	0	0	0	0	5	4	0	0	0	0
76	3	72	0	0	0	76	2	4	0	0	0
77	0	0	0	0	0	77	4	0	0	0	0

Tables (A.10) and (A.11) shows the authors and co-authors of years [2000,2003]. The steps explained in section (2) shows the way how co-authorship network is created and updated to

train the data. As a next step we will describe how to find the mutual friends in the following section.

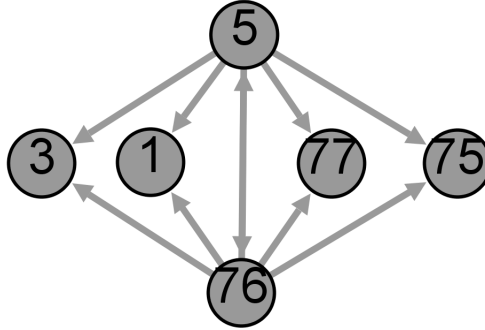
3 Generating Mutual Friends

The definition of a mutual friend is intuitively the common authors of a given pair of authors in a network. This can be formulated as the following;

$$mf_{ij} = cn_i \cap cn_j \quad (\text{A.2})$$

where mf_{ij} is mutual friend of ij pair, cn_i and cn_j are the connected neighbors (co-authors) of nodes i and j , respectively. To illustrate this notion clearer, let us use the connected components of 5 and 76 in year 2000. From table (A.6),

Figure A.3: Mutual Friends of 5-76 in 2003. Table (A.10)



$$cn_5 = \{1, 3, 75, 76, 77\} \quad (\text{A.3})$$

$$cn_{76} = \{1, 3, 5, 75, 77\} \quad (\text{A.4})$$

$$mf_{5-75} = \{1, 3, 75, 77\} \quad (\text{A.5})$$

Using this idea, we find the mutual friends of the co-author network of year 2000 and show in table (A.12).

As seen in table (A.12), we find the mutual friends by pairing the authors in network, and comparing their co-authors. It is important to note that we pair the node i by matching the rest of the nodes in the network with it. An algorithm computing the mutual friends is given below;

Table A.12: Mutual Friends of co-author network in 2000.

Author		Mutual Friends			
i	j	(i-j)			
2	4	0	0	0	0
2	5	0	0	0	0
2	76	0	0	0	0
2	77	0	0	0	0
4	5	0	0	0	0
4	76	0	0	0	0
4	77	0	0	0	0
5	76	1	3	75	77
5	77	1	3	75	76
76	77	1	3	5	75

Algorithm 1 Computing mutual friends

```

1: Declaring the Variables:
2:  $N$  : ▷ Number of authors in the co-author network
3:  $mf_{ij}$  : ▷ Mutual Friend of ij pair
4:  $ca_i$  : ▷ Co-Authors of the node i
5:  $ca_j$  : ▷ Co-Authors of the node j
6:
7: Initialize:
8:  $id \leftarrow$  Row Size of Co-Authorship Network
9:  $N \leftarrow \binom{id}{2}$ 
10: Loop:
11: for  $i := 1 \rightarrow N$  do
12:   for  $j := i + 1 \rightarrow N$  do
13:      $mf_{ij} \leftarrow ca_i \cap ca_j$ 

```

4 Generating Feature Matrix

We create the feature matrix by using the following equations:

$$x_1 = \sum_{\forall k \in mf_{ij}} n_{ca,t}(i, k) \quad (\text{A.6})$$

$$x_2 = \sum_{\forall k \in mf_{ij}} n_{ca,t}(j, k) \quad (\text{A.7})$$

where mf_{ij} is the mutual friends of pair i-j and $n_{ca,t}$ is the number of co-authored papers between author i and k in year t. This means that for x_1 , we sum the number of co-authored papers between i and k in year t where k corresponds to each mutual friends of pair i and j. Likewise, we follow the same steps for x_2 with an only one distinction. For x_2 , we find the number of co-authored papers between j and k where k refers to each mutual friend of pair i - j.

Since we want to compare how link age influences new connections, we generate the features

of each year. To do so, we use equations (A.6) and (A.7) for $t = \{y, y+1, y+2\}$. Since having 3 consecutive years, and 2 features for each year, we have 6 features in total.

Table A.13: Features with respect to years

Year	Features	
y	$x_1 = \sum_{\forall k \in mf_{ij}} n_{ca,y}(i, k)$	$x_4 = \sum_{\forall k \in mf_{ij}} n_{ca,y}(j, k)$
$y+1$	$x_2 = \sum_{\forall k \in mf_{ij}} n_{ca,y+1}(i, k)$	$x_5 = \sum_{\forall k \in mf_{ij}} n_{ca,y+1}(j, k)$
$y+2$	$x_3 = \sum_{\forall k \in mf_{ij}} n_{ca,y+2}(i, k)$	$x_6 = \sum_{\forall k \in mf_{ij}} n_{ca,y+2}(j, k)$

We already declared the notations given in feature equations, and thoroughly explained the mutual friend term in section (3). We now focus how to compute the number of co-authored papers in the following section.

Computing Number of Co-Authored Papers between Two Nodes

Number of co-authored papers between two nodes is computed by the following: We first find the papers that two authors published. Then, we find the papers that these two authors have in common. Finally, the number of the common papers will give us number of co-authored papers between two nodes.

$$n_{ca}(i, k) = |pap_i \cap pap_k| \quad (\text{A.8})$$

where $n_{ca}(i, k)$ is the number of co-authored papers between i and k, pap_i and pap_k are the papers of i and k in year t, respectively.

So far, we have covered all the concepts to extract the feature matrix. To illustrate how to generate the feature matrix, we use the sample data given in the previous sections. For example, mutual friends of authors 5 and 76 in 2000 in sample data given in table (A.12) are;

$$mf_{(5-76)} = \{1, 3, 75, 77\}$$

Recall that we set $y=2000$ in the examples we give in this chapter, so we need to find equations (A.13) and (A.13) to compute the features of (5-76).

$$\begin{aligned} x_1 &= \sum_{\forall k \in mf_{(5-76)}} n_{ca,2000}(i, k) \\ &= n_{ca,2000}(5, 1) + n_{ca,2000}(5, 3) + n_{ca,2000}(5, 75) + n_{ca,2000}(5, 77) \end{aligned}$$

We need to compute number of co-authored papers $n_{ca,2000}$ to find the features. Using equation (A.8),

$$n_{ca,2000}(5, 1) = |pap_5 \cap pap_{76}|$$

From (A.5),

$$\begin{aligned}
 pap_5 &= \{55, 56, 57, 58, 59\} \\
 pap_{76} &= \{55, 56, 58\} \\
 n_{ca,2000}(5, 1) &= |\{55, 56, 57, 58, 59\} \cap \{55, 56, 58\}| \\
 n_{ca,2000}(5, 1) &= 3
 \end{aligned}$$

By the same token, we compute $n_{ca,2000}(5, 3) = 4$, $n_{ca,2000}(5, 75) = 1$ and $n_{ca,2000}(5, 77) = 1$.

$$x_1 = 3 + 4 + 1 + 1 = 9$$

By following the same steps, we find the feature values of 5 and 76 for years 2001 and 2000.

Table A.14: Feature Matrix of years 2000,2001,2002

i	j	x_1	x_2	x_3	x_4	x_5	x_6
2	4	0	0	2	0	0	2
2	5	0	0	0	0	0	0
2	76	0	0	1	0	0	1
2	77	0	0	0	0	0	0
4	5	0	0	0	0	0	0
4	76	0	0	1	0	0	1
4	77	0	0	0	0	0	0
5	76	9	2	0	4	2	0
5	77	9	0	0	4	0	0
76	77	4	0	0	4	0	0

Bibliography

- [1] CHEN, H.-H., D. J. MILLER, and C. L. GILES (2013) “The Predictive Value of Young and Old Links in a Social Network,” in *Proceedings of the ACM SIGMOD Workshop on Databases and Social Networks*, DBSocial '13, ACM, New York, NY, USA, pp. 43–48.
URL <http://doi.acm.org/10.1145/2484702.2484711>
- [2] WATTS, D. J. and S. H. STROGATZ (1998) “Collective dynamics of small-world networks,” *nature*, **393**(6684), pp. 440–442.
- [3] BARABÁSI, A.-L. and R. ALBERT (1999) “Emergence of scaling in random networks,” *science*, **286**(5439), pp. 509–512.
- [4] FALOUTSOS, M., P. FALOUTSOS, and C. FALOUTSOS (1999) “On power-law relationships of the internet topology,” *ACM SIGCOMM Computer Communication Review*, **29**(4), pp. 251–262.
- [5] LESKOVEC, J., J. KLEINBERG, and C. FALOUTSOS (2005) “Graphs over time: densification laws, shrinking diameters and possible explanations,” in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, ACM, pp. 177–187.
- [6] ALBERT, R. and A.-L. BARABÁSI (2002) “Statistical mechanics of complex networks,” *Reviews of modern physics*, **74**(1), p. 47.
- [7] NEWMAN, M. E. (2003) “The structure and function of complex networks,” *SIAM review*, **45**(2), pp. 167–256.
- [8] LIBEN-NOWELL, D. and J. KLEINBERG (2007) “The link-prediction problem for social networks,” *Journal of the American society for information science and technology*, **58**(7), pp. 1019–1031.
- [9] CHEN, H.-H., L. GOU, X. L. ZHANG, and C. L. GILES (2012) “Discovering missing links in networks using vertex similarity measures,” in *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, ACM, pp. 138–143.
- [10] JEHL, G. and J. WIDOM (2002) “SimRank: a measure of structural-context similarity,” in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, pp. 538–543.
- [11] AL HASAN, M., V. CHAOJI, S. SALEM, and M. ZAKI (2006) “Link prediction using supervised learning,” in *SDM06: Workshop on Link Analysis, Counter-terrorism and Security*.

- [12] LÜ, L. and T. ZHOU (2011) “Link prediction in complex networks: A survey,” *Physica A: Statistical Mechanics and its Applications*, **390**(6), pp. 1150–1170.
- [13] GETOOR, L. and C. P. DIEHL (2005) “Link mining: a survey,” *ACM SIGKDD Explorations Newsletter*, **7**(2), pp. 3–12.
- [14] AL HASAN, M. and M. J. ZAKI (2011) “A survey of link prediction in social networks,” *Social network data analytics*, pp. 243–275.
- [15] DEMPSTER, A. P., N. M. LAIRD, and D. B. RUBIN (1977) “Maximum Likelihood from Incomplete Data via the EM Algorithm,” *Journal of the Royal Statistical Society. Series B (Methodological)*, **39**(1), pp. pp. 1–38.
URL <http://www.jstor.org/stable/2984875>
- [16] BISHOP, C. M. (2006) *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag New York, Inc., Secaucus, NJ, USA.
- [17] FAWCETT, T. (2006) “An introduction to ROC analysis,” *Pattern recognition letters*, **27**(8), pp. 861–874.