SUPERVISED AND FUZZY RULE BASED LINK PREDICTION IN WEIGHTED CO-AUTHORSHIP NETWORKS

by

Aziz Asil

B.S., Computer Engineering, Doğuş University, 2003

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering
Boğaziçi University

2017

ACKNOWLEDGEMENTS

I would like to thank my advisor Mr. Fikret Gürgen and Mr. Albert Ali Salah for their genuine encouragement, patience and guidance. Furthermore, I would like to thank my family for their support in order to finish this study.

ABSTRACT

SUPERVISED AND FUZZY RULE BASED LINK PREDICTION IN WEIGHTED CO-AUTHORSHIP NETWORKS

Link Prediction is a fundamental problem in the social networks analysis. In order to solve this problem supervised learning algorithms, which include one fuzzy rule based algorithm were applied in this study. Besides supervised learning algorithms, an unsupervised strategy is also applied to compare the supervised and unsupervised results. Two different networks are chosen for the experiments: a computer science co-authorship network and an eye disease co-authorship network. Those networks were used in both weighted and unweighted versions in the experiments. In a network, a weight refers the strength of a relationship between nodes. Our Experiments' results proved that weighted networks had better results in comparison to unweighted networks. In the link prediction, the task is to predict the new connections in future for unconnected pair of nodes in present. In the link prediction process with supervised algorithms, metric values were employed as predictor attributes and existence of links was used as class labels. On the other hand, in link prediction process with the unsupervised strategy, a ranking method was employed. In the light of the experiments, it was seen that supervised algorithms had better performance than the unsupervised strategy. Furthermore, among the supervised algorithms, a decision tree and a probabilistic algorithm provided the best performances in comparison with fuzzy rule based algorithm.

ÖZET

AĞIRLIKLANDIRILMIŞ ORTAK YAZARLIK AĞLARINDA EĞİTİCİLİ VE BULANIK MANTIK KURAL TABANLI LİNK TAHMİNİ

Link tahmini, sosyal ağ analizinde temel bir problemdir. Halihazırda aralarında bağlantı olmayan nodların gelecekte aralarında bağlantı kurma tahminini yapmak bir sınıflandırma problemidir. Bu problemin çözümü için bu çalışmada eğiticili öğrenme algoritmaları ve eğiticisiz bir metot kullanılmıştır. Bulanık mantığın link tahminindeki sonuçlarını da görebilmek için eğiticili öğrenme algoritmalarından biri olarak bulanık mantık temelli bir algoritma kullanılmıştır. Iki farklı ağ kullanılmıştır: bunlar, bilgisayar bilimleri yazarlarından oluşan ve göz hastalıkları yazarlarından oluşan ağlardır. Bu ağlar ayrıca kendi içlerinde ağırlıklı ve ağırlıksız olarak kullanılmıştır. Bir ağda ağırlık, iki nod arasındaki ilişkinin gücünü gösterir. Deney sonuçları göstermiştir ki ağırlıklı ağlarla elde edilen sonuçlar, ağırlıksız olarak kullanılan ağlarla elde edilen sonuçlardan daha iyi çıkmıştır. Link tahmini, şuan aralarında bağlantı olmayan iki nod için, gelecekteki yeni bağlantıyı tahmin etme görevidir. Eğiticili öğrenme algoritmaları ile yapılan link tahmini deneylerinde, nodların metrik değerleri input olarak kullanılmış, bu nodların gelecekte aralarındaki link varlığı ise sınıf etiketi olarak kullanılmıştır. Eğer aralarında link varsa sınıf etiketi 1 olarak, eğer link yoksa sınıf etiketi 0 olarak atanmıştır. Eğiticisiz metotta ise her bir nod çifti için metric değerleri en yüksekten en düşüğe kadar sıralandırılmıştır. Deney sonuçları göstermiştirki eğiticili öğrenme algoritmaları ile elde edilen performans, eğiticisiz metot ile elde edilen performansdan daha iyi çıkmıştır. Eğiticili öğrenme algoritmaları ile yapılan deneylerin sonuçlarında ise olasılık teoremini ve karar ağacını kullanan algoritmalar, bulanık mantık kullanan algoritmadan daha iyi sonuç vermiştir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

I^F	Fuzzy Interval
$\phi^{c,L}$	Lower bound of the core
$\phi^{c,U}$	Upper bound of the core
$\phi^{s,L}$	Lowerbound of the support
$\phi^{s,U}$	Upper bound of the support

LIST OF ACRONYMS/ABBREVIATIONS

AA	Adamic Adar Coefficient
CC	Clustering Coefficient
CN	Common Neighbors
CSCN	Computer Science Co-authorship Network
EDCN	Eye Disease Co-authorship Network
fn	False negatives
fp	False positives
FURIA	Fuzzy Unordered Rule Induction Algorithm
IBK	Instance-Based method based on k
J48	Is an open source Java implementation of the C4.5 algorithm in WEKA
JC	Jaccard Coefficient
LibLinear	It's a library for linear support vector machines
LibSVM	An algorithm, which uses SVM algorithm
LP	Local Path
Network1	Network, which regards the contribution of authors
Network2	Network, which regards the number of papers of authors
PA	Preferential Attachment
PD	Path Distance
RIPPER	Repeated Incremental Pruning to Produce Error Reduction
SNA	Social Network Analysis
tp	True positives
WEKA	Waikato Environment for Knowledge Analysis

1. INTRODUCTION

A social network is a social structure, which consists of social actors and relationships between these actors. These relationships may be friendship, like, dislike [1], knowledge etc. A social network may have online structure, which consists of people who interact with each other at a moment whenever they want. Online dating networks and online social medias for example Twitter and Facebook are kinds of online social networks. Internet has a great contribution to create and improve the social networks. Availability of internet connections and smart mobile devices make this improvement easy.

A social network can be represented as a graph. In this graph, people can be considered as vertices and their relationships can be considered as edges. In Figure 1.1., there is a sample of a node and an edge. In this figure colored circles are nodes and they can be assumed that authors and lines are edges and they can be assumed that papers whom authors study together.

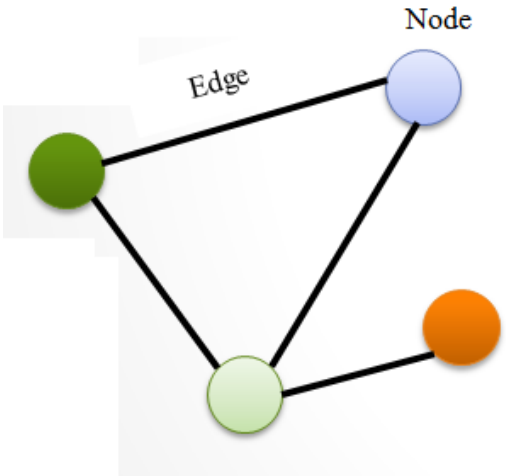

Figure 1.1. A sample of small social network.

Networks can be dived into two main categories according to their kind of relationships. If there is only one type relation in a graph it is called one-dimensional network. If there are many types of relationships (e.g. friends, relatives, colleagues) between nodes it is called multidimensional network [2]. When Facebook network is

considered, two nodes may have relations in terms of being friends, relatives and colleagues at the same time. For coauthors network, there are different types of relations such as attending the same conference, citing the same papers and collaborating a paper etc.

Social networks have a dynamic structure. It comes from their nature. Existing connections can be dissolved or new connections can be established in time. Link prediction is a task of predicting those new connections (links) between two disjoint nodes in the near future. In a social network, Link prediction can help to uncover the relationships, which are probably exist but not observed yet. In the real world it can be used to anticipate the future behavior of a network, which is difficult to observe directly. Many studies made the link prediction process in different kind of networks, such as co-authorship networks [3-7], web-page networks [1, 4], protein networks [8-10] and paper citation networks [2, 11].

Link Prediction is one of the methods, which are used in Social Network Analysis (SNA). Social Network Analysis (SNA) uses some techniques to understand the behavior of a network [12, 13]. SNA may help in a wide range of fields such as health, economics and safety.

There are three mainly different approaches to treat the link prediction problem:

- Topological/ structural measures
- Similarity between nodes
- Probabilistic models.

These approaches will be explained in detail under “Link Prediction” section.

Evolving the social networks is still fundamental question and this forms the motivation of this study. In this study, it is focused on exploiting the structural information of given networks and both supervised learning strategy and a fuzzy rule based method are applied to solve the link prediction problem. In the supervised learning

strategy, link prediction is treated as a binary classification problem about two nodes whether there is a link between them or not. Every pair of nodes is an instance of the classification problem and metric values of the nodes, which are obtained from network are predictor attributes. On the other hand, target attributes indicate a presence or an absence of the relationship in the future. Presence of the relationship indicates a positive label and absence of the relationship indicates a negative label. In the fuzzy rule based method, which is Fuzzy Unordered Rule Induction Algorithm (FURIA), using of the predictor and target attributes are same as they are used in supervised learning but it uses “soft” decision boundaries instead of using “sharp” decision boundaries. In this study, NaivesBayes, J48, IBK, LibLinear, LibSVM and FURIA algorithms are applied for supervised learning and only FURIA uses a fuzzy rule based algorithm among them. Some of these methods such as decision tree, K-NN, multilayer perceptron, SVM, RBF network are also used in [14]. At the end of this study, result of the fuzzy ruled based and the other supervised learning algorithms are compared. Link prediction process is shown In Figure 4.4.

Many unsupervised methods were used in previous studies by using topological structure of the network. These methods are explained in detail in [3]. One of the unsupervised method the preferential attachment [15, 16] is calculated by product of their degrees. Another method is counting of the common neighbors [17] of two separated nodes, which have a length of path that is 2. Jaccard’s Coefficient [18] is another method that uses the number of common neighbors, which is divided into total number of neighbors of those two nodes. Another variation of common neighbor calculation is used in Ademic / Adar measure [19], which will be explained under metric section.

Metric values of the nodes, which are obtained from the networks and used as the predictor attributes in supervised learning and fuzzy rule based method give important information about the nodes. For example, a high Number of Common Neighbors for two separated nodes indicates a high probability of a connection in future. In contrast to a high number of common neighbors, if there are no common neighbors between those separated nodes, there is a low probability of a connection between those two nodes in future. For example, it is assumed that there are two situations about number of common neighbors between two people who do not know each other. In situation 1 they have many common friends and in situation 2 they have no common friends. For situation 1, in future those two

people may meet in a way by helping of common neighbors with a high probability compared to people in situation 2. Other metric values are Jaccard Coefficient, Preferential Attachment, Adamic Adar, Path distance, Local Path and Local Clustering Coefficient, which will be explained in detail under Metric section.

In a network, weight describes the strength between two nodes. To include the strength in a link prediction, supplies more information in terms of the degree of a relationship in comparison to unweighted network. For example, in a co-authorship network, if a couple contributes only one paper, their contribution (weight) to that paper is 1 for each authors but, if they have many contributions for more than one paper, their contribution (weight) will be greater than 1 and it means that they have a strong relationship in comparison to previous couple who publishes only one paper in a collaboration. In an unsupervised link prediction study [20], it is claimed that weighted calculation improves the performance of the link prediction. In our study, when weighted networks are used, performance of the supervised link prediction (including fuzzy rule based algorithm) improves. Weights will be explained in more detailed in social network data section.

In this study two different networks are studied on: a computer science co-authorship network (CSCN) and an eye disease co-authorship network (EDCN). A co-authorship network consists of authors, which are willing to collaborate with each other to reach a common aim. For this reason a co-authorship network is selected. [21] explained the nature of co-authorship in detail. Computer science co-authorship network data is available at DBLP(Digital Bibliography & Library Project), which consist of many different research publications about Computer Science [22]. Some studies [4, 5, 13, 23] made their link prediction processes over DBLP. Data, which are used for implementing the supervised link prediction and fuzzy rule based method is extracted from DBLP and the eye disease network [24]. Some studies preferred a medical co-authorship network for the link prediction task. One of them is [25]. It contains authors who have papers about coronary artery disease and it is a similar network with our eye-disease network in terms of the disease fields. To make classification, WEKA [26] tool, which consists of many different learning algorithms was used. These studies [4, 5, 25] made the link prediction by using WEKA, as well. Our experiments demonstrate that performance of the link prediction in

the weighted networks is better than performance of the link prediction in the unweighted networks for fuzzy rule based algorithm and supervised learning algorithms. Furthermore, most of the performance results of the link prediction in the supervised learning algorithms (NaivesBayes, J48, IBK, LibLinear and LibSVM) is better than performance of the link prediction in fuzzy rule based algorithm (FURIA).

2. RELATED WORKS

There are many studies about the link prediction task. These studies vary according to their kind of networks, methods and algorithms.

Gimenes *et al.* [4] considered the link prediction task as a binary classification problem and run the experiments over the Digital Bibliography & Library Project (DBLP), which contains Computer Science publications that represent a co-authorship graph. They had 400 positive and 400 negative instances in order to provide an equal class distribution. If a pair of nodes had a connection in future, they called this pair "a positive instance" and it was labelled as 1. On the other hand, if a pair of nodes had not a connection in future, they called this pair "a negative instance" and it was labelled as 0. In order to classify, pairs must be represented as vectors of numbers, so each dimension of the vectors is a metric. For this reason they used edge-oriented metrics, which are Number of common neighbors (CN), Jaccard's coefficient (JC), Preferential attachment (PA), Adamic-Adar coefficient (AA), Path distance (PD), Local path (LP), Local clustering coefficient (CC). These metrics are calculated from the topological information of the network. By using five algorithms, which are J48, Naive Bayes, Multilayer Perceptron, Bagging, and Random Forest, they tried to solve classification problem with classical 10-fold cross validation. They applied those algorithms in the Weka framework. In order to measure the effect of the number of coauthoring, they considered the authors that had at least k coauthoring during past and present intervals. At the end of their experiment, they claimed that k must be between 2 and 4. In that k interval they got better performance in most of the algorithms for all time intervals. They used three different time intervals in the same co-authorship network to see the effects of the length of the time. They claimed that time parameters can alter the results of the recommendations. According to their experiment, they believed that short past and short present configuration had a bigger potential for link prediction because of the memory of the system was more recent. When their experiment results are considered, their recommendation accuracy rates was around the 90 % and they found that decision trees worked better than neural networks and Naive Bayes classification. They did not consider contribution of authors (weight) value in prediction process. Furthermore there was no evidence in the results about their supervised method was better than an

unsupervised method because they did not use any unsupervised methods to compare the results. Finally they used only one co-authorship network, which contained computer science publications. When this supervised method was applied to another kind of co-authorship network, it was not discussed whether results would be same or not. They continued their link prediction study [27] as extend of this study with similar methods.

Another link prediction study [5] used both supervised and unsupervised methods. This study is very similar to [4] in terms of the definition of the problem, network type, chosen metrics, framework of supervised learning algorithms (WEKA) and used supervised methods. There are two fundamental differences between Sa *et al.* [5] and Gimenes *et al.* [4]. The First difference is that Sa *et al.* used unsupervised methods besides supervised methods in order to compare the results of two different methods. The second difference is that Sa *et al.* used the weighted version of the network. In the study of Sa *et al*, number of co-authoring did not considered in contrast to study of the Gimenes *et al*. According to both supervised and unsupervised results of the study, they found that weighted networks had better performance than unweighted networks at most cases. Among the supervised algorithms, LibLinear algorithm had highest accuracy, followed by J48. Their unsupervised method was based on the ranking system and 8 different metrics (CN, JC, PA, AA, PD, RA, LP and CC) were used together with their weighted versions for ranking. First of all for each metric value, their minimum and maximum values were calculated. Then according to these minimum and maximum values, a rank value was assigned for each metric value. They stated that supervised results (except IBK) were better than the unsupervised strategy in terms of F-measure values. This study considered only one co-authorship network like study of Gimenes *et al.*. Hence, it was not certain that their results would be same with another kind of co-authorship network. In our study, we applied a link prediction method to the additional co-authorship network and we got same results, which were most of the supervised algorithms showed better performance than the unsupervised strategy. Furthermore, we proved that (in most cases) all two weighted networks had better performances in both supervised method and the unsupervised strategy in comparison with two unweighted networks. [11, 25] also used different kind of supervised learning algorithms.

Liben-Nowell and Kelinberg's study [3] is one of the works in terms of the link prediction via an unsupervised strategy. They tried to discover whether a link can be established between two nodes in future by using past and present information of the network. They used five co-authorship networks. Their method was based on the node similarity measures. After extracting the network's data and calculating the similarity measures, they made a ranking by descending order of similarity. This ranking was then used to predict the links between nodes according to the ranking value. The higher the rank, the more likely the interaction was to establish in the future. The figures showed that predictors that had good results in one network did not necessarily have good results in all networks. It is clear in our results that the performance of the unsupervised strategy is unstable from one network to the other network. Lichtenwalter *et al.* [28] concluded same results, which were about the instability of the unsupervised strategy with us. At the end of their study, Liben-Nowell and Kelinberg admitted that this unsupervised method did not have satisfactory results because of such ranking. In our study we used an unsupervised strategy in order to compare the results, which we got with supervised learning algorithms, as well. In our study, most of the supervised algorithms showed better performance than the unsupervised strategy.

Link prediction can be applied in heterogeneous networks, which have different types of relationships. Sun *et al.* [2] considered not only author type but also venues, topics and papers as types of objects in their study. Similarly, they included in more than one type of links. They considered "publishing", "citing" and "containing" in addition to the type of "writing" as types of links in DBLP network. They used a supervised model by using the extracted topological features of the network. They made the class distribution equally in the training dataset. Their dataset consisted of four different sets according to author's productivities and number of edges (they called "hops"), which were needed to reach from a source node to a target node. In terms of the productivity they created two sets, which consisted of authors who had at least 16 papers and authors who had papers between 5 and 15. For the first set, they called it as "high productive" and for the second set they called it "less productive". They separated authors like this because they tried to avoid the excessive computing between authors, which were unrelated. Every sets had two subsets according the number of hops, which were "2 hop" and "3 hop".

According to their result, they stated that link prediction with heterogeneous topological features had slightly better results than link prediction with homogenous topological features in most of the datasets. On the other hand, this study was lack of any comparisons with another network because they handled only one network in that study.

Using negative edges (indicating relations such as opposition or antagonism) in addition to positive edges (indicating relations such as friendship) can be useful for link prediction problem. Leskovec *et al.* employed negative edges for three different networks (epinions, slashdot and wikipedia) with the logistic regression classifier in their study [1] and they stated that using the negative edges with positive edges improved the link prediction results around 3 %.

3. LINK PREDICTION

There are plenty of works about the link prediction. Liben-Nowell and Kelinberg's study [3] is one of the important works. They tried to discover whether a link can be established between two nodes in future by using past and present information of the network. Their method was based on node similarity measures. After extracting the network's data and calculating the similarity measures, they made a ranking by descending order of similarity. This ranking is then used to predict the links between nodes according to the ranking value. The higher the rank, the more likely the connection is to establish in the future. The figures showed that predictors that have good results in one network do not necessarily have good results in all networks. It is clear in our results that the performance of the unsupervised strategy is unstable from one network to the other network. At the end of their study, they admitted that this unsupervised method did not have satisfactory results because of such ranking.

Definition of the link prediction is that: In a snapshot of a social network at time t, prediction the new links between time interval t and future time t' for disjointed nodes. In Figure 3.1., there is no connection between Node A and Node B in time interval 0 and t. However, in time interval t and t' a connection appears. The "Link prediction" problem aims to find this connection in that time interval t and t'.

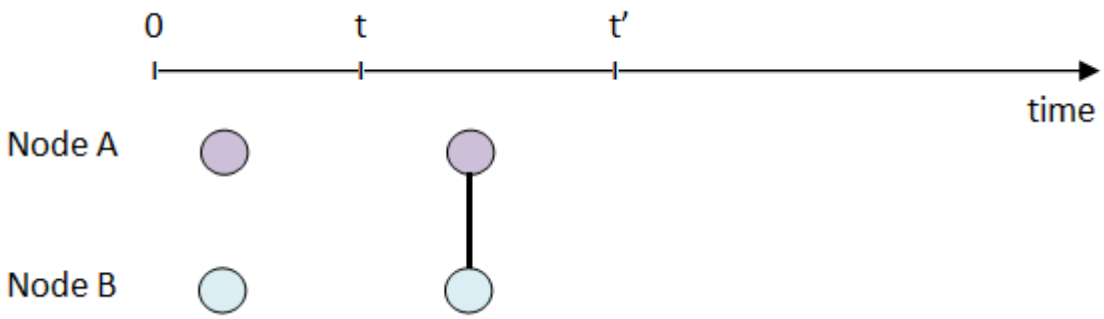

Figure 3.1 Showing the link between nodes in future.

To solve this problem three main models are used. These are [29]:

- Topological/Structural Model
- Node-wise similarity model
- Probabilistic model.

In the Topological/Structural model, different metrics, which show specific features of nodes are used. Those metric values will be explained under metrics section in detail. According to data, which are obtained from metrics, a model is built to predict the hidden links. Some studies [15, 20, 30] used topological model to predict the links.

On the other hand, in node-wise similarity model, similarity of two nodes is taken into account according to the content or semantics that they represent. Every node has a content, which is defined as a feature vector. Similarity between two nodes is defined by a similarity measure such as information content, mutual information, cosine coefficient etc. By using the feature vector and a similarity measure, node-wise similarity is calculated [31].

Finally, probabilistic model tries to produce a compact model that abstracts the social network best. In this model, the aim is to learn a model, which consists of the set of parameters in a given observed social network. Probabilistic Relational Models (Relational Bayesian Networks, Relational Markov Networks etc.), Bayesian Relational Models, Stochastic Relational Models are some of Probabilistic Model Based Approaches [31].

In this study Topological model is used. Supervised learning methods and an unsupervised strategy may be applied to this model. Supervised learning models including fuzzy rule based method are applied because of the better performance compared to unsupervised strategies. It is also studied in an unsupervised strategy to compare the results with supervised learning method's results. In this study, the unsupervised strategy refers to a ranking solution and it is not a kind of a learning model. Some studies [3, 19, 20] applied unsupervised methods to Topological Model. If there is no training set, which consists of network data, which are used for any prediction model, this method is called unsupervised

strategy. Unsupervised strategies are not learning models and include some defects. For example, if the link prediction will be made by only Common Neighbors unsupervised method, firstly CN (Common Neighbors) values are calculated for all unconnected nodes. Then CN values are ranked from the top to the bottom. The first problem arises here. In this list what top ranked number should be taken into account in order to presume existence of the links? The top 10 or 20 or 50 ranked pairs in the list? If this value is T (let's call treshold), how should be decided the value of T? The second problem is about the usage of more than one different unsupervised strategy together with. For CN value, the more CN value is high, the more tendency of creating link is great. Hence, aspect from CN, tendency of establishing the link is higher at the top ranked pairs not at the bottom ranked pairs. This may not be true if more than one metric will be used together in the link prediction. For example higher value is good for CN metric, but Adamic-Adar gives more importance to the common neighbors, which have fewer neighbors. Hence, if CN and AA (Adamic-Adar) will be used together for the link prediction, how will be decided to T threshold value?

For these limitations in the unsupervised strategy, supervised machine learning including fuzzy rule based strategy is applied in this study. Weighted network calculation also applied to this study to compare the result with unweighted network calculation. The contribution of this thesis to the link prediction study is that using the fuzzy rule based method along with supervised learning strategy and applying this to two different networks.

4. METRICS

Metric values of a node give important information about the node. By using this information, some predictions can be made about nodes and networks, which they are involved in. The role of the metric values in this study is that they are used as predictor attributes in supervised learning and fuzzy rule based method. Metric values are calculated by developing of java codes.

In this section, Common Neighbor, Jaccard Coefficient, Preferential Attachment, Adamic Adar, Path distance, Local Path and Local Clustering Coefficient will be explained in detail.

4.1. Common Neighbors

Number of common neighbors [16, 17] is the most widespread metric because of its simplicity. It refers the total common neighbors of the node x and node y. Usually if a couple of node has more common neighbors, it is a high probability to have a connection in future.

The CN measure for unweighted networks:

$$CN(x,y) = |\Gamma(x) \cap \Gamma(y)| \tag{4.1}$$

The CN measure for weighted networks:

$$CN(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} w(x,z) + w(y,z) \tag{4.2}$$

In Figure 4.1., there is a small network, which consists of X, A, B, C, D and Y nodes and arbitrary weights between nodes. While determining the weights, it's assumed that:

- X and A has a collaboration in 2 papers
- X and B has a collaboration in 1 paper
- X and C has a collaboration in 2 papers
- X and D has a collaboration in 1 paper
- Y and A has a collaboration in 3 papers
- Y and B has a collaboration in 1 paper
- Y and C has a collaboration in 1 paper
- Y and D has a collaboration in 2 papers

The common neighbors of X and Y is A, B, C and D.

- Unweighted CN(X,Y) = 4
- Weighted CN(X,Y) = (w(XA) + w(AY)) + (w(XB) + w(BY)) + (w(XC) + w(CY))
  + (w(XD) + w(DY))
  = (2+3)+(1+1)+(2+1)+(1+2)
  = 13

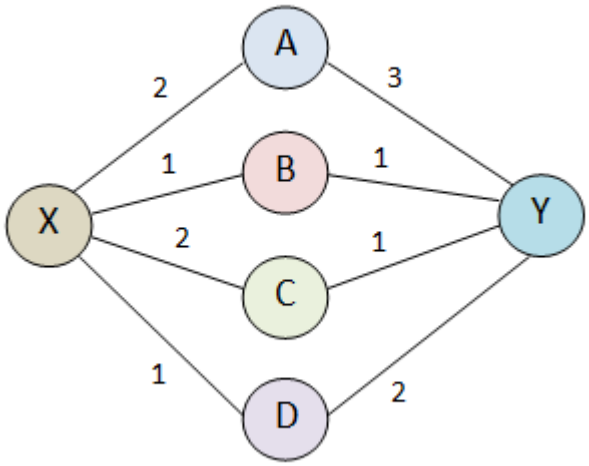

Figure 4.1 Showing common neighbors of node X and node Y in a weighted network.

4.2. Jaccard Coefficient

Jaccard coefficient [18] refers a proportion of intersection of the nodes and union of the nodes. As intersection increases, tendency of the future connection increases as well.

The JC measure for unweighted networks:

$$JC(x,y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|} \tag{4.3}$$

The JC measure for weighted networks:

$$JC(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{w(x,z) + w(y,z)}{\sum_{a \in \Gamma(x)} w(a,x) + \sum_{b \in \Gamma(y)} w(b,y)} \tag{4.4}$$

In Figure 4.2., there is a small network, which consists of X,A,B,C,D,E,F,G and Y nodes and arbitrary weights between nodes for Jaccard Coefficient calculation.

- Unweighted JC(X,Y) = 4/11
  = 0,36
- Weighted JC(X,Y) = (w(XA) + w(AY)) + (w(XB) + w(BY)) + (w(XC) + w(CY)) + (w(XD) + w(DY)) / ((w(EX) + w(XA) + w(XB) + w(XC) + w(XD)) + (w(YA) + w(YB) + w(YC) + w(YD) + w(YF) + w(YG)))
  = ((2+3)+(1+1)+(2+1)+(1+2)) / (9+10)
  = 13/19
  = 0,68

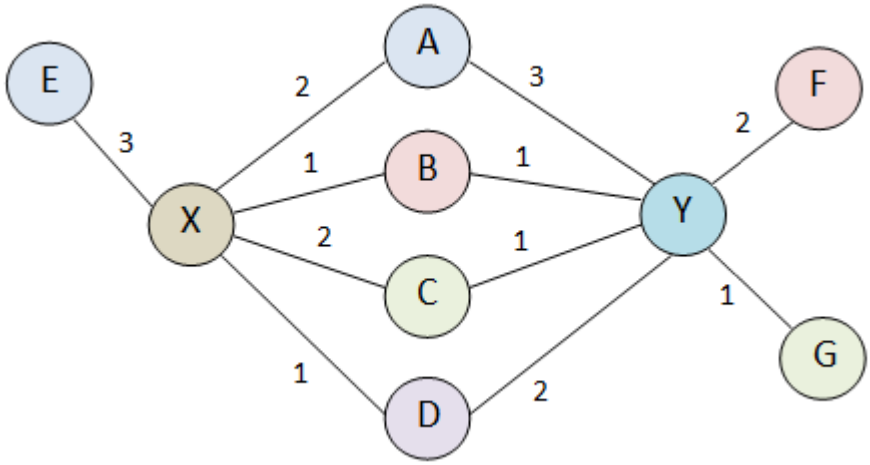

Figure 4.2. A small weighted network for Jaccard Coefficient calculation.

4.3. Preferential Attachment

Preferential attachment [15, 16] is calculated by production of the number of the neighbors of the node x and node y. According to this metric value, probability of the future connection between nodes increases as their number of total neighbors increase.

The PA measure for unweighted networks:

$$PA(x,y) = |\Gamma(x)| * |\Gamma(y)| \tag{4.5}$$

The PA measure for weighted networks:

$$PA(x,y) = \sum_{a \in \Gamma(x)} w(a,x) * \sum_{b \in \Gamma(y)} w(b,y) \tag{4.6}$$

According to Figure 4.2., PA values for node X and Y are as below:

- Unweighted PA(X,Y) = 5×6
  = 30

- Weighted PA (X,Y) = (w(EX) + w(XA) + w(XB) + w(XC) + w(XD)) × (w(YA) + w(YB) + w(YC) + w(YD) + w(YF) + w(YG))
  = (3+2+1+2+1) × (3+1+1+2+2+1)
  = 90

4.4. Adamic Adar Coefficient

Adamic Adar coefficient metric [19] gives more importance to a common neighbor who has fewer neighbors. Hence, it measures how strong is the relationship between a common neighbor and the evaluated pair of the nodes.
The AA measure for unweighted networks:

$$AA(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\log(|\Gamma(z)|)} \tag{4.7}$$

The AA measure for weighted networks:

$$AA(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{w(x,z) + w(y,z)}{\log\left(1 + \sum_{c \in \Gamma(z)} w(z,c)\right)} \tag{4.8}$$

According to Figure 4.2., AA values for node X and Y are as below:

- Unweighted AA(X,Y) = (1/log2) + (1/log2) + (1/log2) + (1/log2)
  = 13,2
- Weighted AA(X,Y) = ((w(XA) + w(AY)) / log(1 + (w(XA) + w(AY))) + ((w(XB) + w(BY)) / log(1 + (w(BX) + w(BY))) + ((w(XC) + w(CY)) / log(1 + (w(CX) + w(CY))) + ((w(XD) + w(DY)) / log(1 + (w(DX) + w(DY)))
  = (5/log6) + (2/log3) + (3/log4) + (3/log4)
  = 20,58

4.5. Path Distance

Path Distance gives the minimum number of nodes in order to reach from node x to node y. The lower is path distance value, the higher is chance to establish a link in future between nodes. If node x and y have a common value, PD(x,y) =1. For weighted networks PD is calculated by 1/w(x,y).

According to Figure 4.2., PD values for some nodes are as below:

- PD(X,Y) = 1
- PD(E,F) = 3
- PD(X,G) = 2
- PD(A,B) = 1
- PD(E,D) = 1

Weighted PD(E,D) = (1/w(E,X)) + (1/w(X,D))
= (1/3) + (1/1)
= 1,33

4.6. Local Path

For all paths of length exactly l from node x to node y is calculated by formulation of $paths_{x,y}^{(l)}$ .Local Path gives more information about the neighborhood of the nodes. Tao Zhou, Linyuan Lu, and Yi-Cheng Zhang [32] had a study about predicting the missing links on the basis of node similarity in six different networks. They used nine similarity measures and one of them is Local Path.

The LP counts the length exactly 2 and 3 between two unconnected nodes. $Paths_{x,y}^{2}$ (length 2) is equally to the number of common neighbors (CN) value clearly. When

closeness is considered, paths of length 2 is more relevant than paths of length 3. Because two unconnected nodes are closer to each other with length 2 than length 3. For example in Figure 4.3., there is only one node (Y) between unconnected two nodes (F and G) for length 2, but there are two nodes (A and Y) between another unconnected two nodes (E and G) for length 3. Because paths of length 2 are more relevant than paths of length 3, an adjustment factor e is applied to formulation for unweighted networks like this [5]:

$$LP(x,y) = \left|paths_{x,y}^{2}\right| + e * \left|paths_{x,y}^{3}\right| \tag{4.9}$$

Tao Zhou, Linyuan Lu, and Yi-Cheng Zhang [32] and took e value very small number (less than 1 but grater then 0) when they involved it in their LP calculation.

Considering x and y are unconnected nodes. In a weighted network, for each path of length 2 ($paths_{x,y}^{(2)}$) is calculated like this: w(x,z)+w(z,y) where z is the common neighbor of node x and y. For each path of length 3 ($paths_{x,y}^{(3)}$) is calculated as w(x, a)+w(a, b)+w(b,y) where a is the neighbor of node x and b is neighbor of node y and there is no connection between a and node y and there is no connection between b and node x.
So:

$$LP(x,y) = \left(\sum_{z\in\Gamma(x)\cap\Gamma(y)} w(x,z) + w(y,z)\right) + \left(e * \sum_{\{x,a,b,y\}\in paths_{x,y}^{(3)}} w(x,a) + w(a,b) + w(b,y)\right) \tag{4.10}$$

In Figure 4.3., there is a small network, which consists of X, A, B, C, D, E, F, G, Y nodes and arbitrary weights between nodes for Local Path calculation. There is a difference between Figure 4.2. and Figure 4.3. In Figure 4.3., there is an additional link, which is colored with red between node E and A.

- Unweighted LP(X,Y) = 4 + (0,7×1)
  = 4,7

- Weighted LP(X,Y) = (w(XA) + w(AY) + w(XB) + w(BY) + w(XC) + w(CY) + w(XD) + w(DY)) + (e × (w(XE) + w(EA) + w(AY)))
  = ((2+3) + (1+1) + (2+1) + (1+2)) + (0,7 × (3+2+3))
  = 13+5,6
  = 18,6

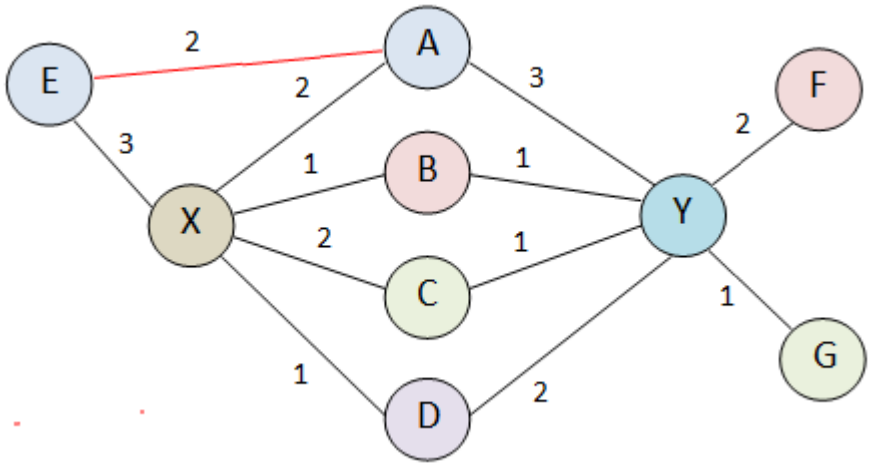

Figure 4.3. A small weighted network for Local Path calculation.

4.7. Local Clustering Coefficient

Cluster Coefficient metric [33] shows tendency to form links between neighboring nodes. According to Cluster Coefficient, creating a clique (complete graph) among neighbors of a node is important. Hence, if local densities of a couple nodes are high, there is a high possibility to establish a future connection between these couple of nodes. In Figure 4.4., there is an example of two graphs, which have a clique and do not have a clique. In this figure, nodes E, X and G have edges among each other, so graph A has a clique. On the other hand, node F and G does not have an edge and for this reason graph B has not a clique. This metric based on counting the triangles, which are formed around the selected nodes. One of the corners of this triangle is the one of the selected node i and other corners are node i's neighbors m and n. A closed triangle is created when m and n

directly connected. Below calculations, t_i is the number of closed triangles attached to node i.

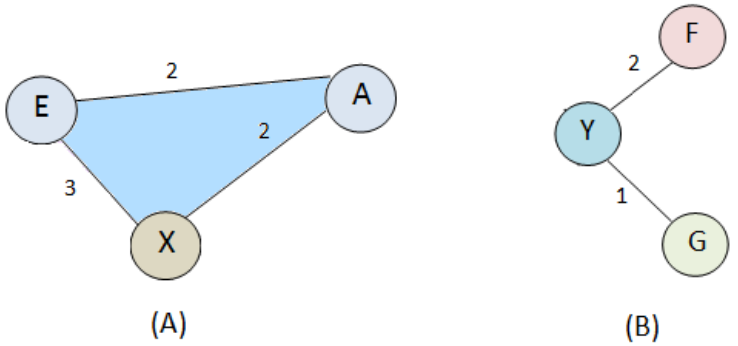

Figure 4.4. A clique (A) (complete graph) and no clique (B) example with weights.

The CC measure for unweighted networks:

$$CC(i) = \frac{2t_i}{|\Gamma(i)| * (|\Gamma(i)| - 1)} \tag{4.11}$$

The CC measure for weighted networks:

$$CC(i) = \frac{1}{|\Gamma(i)| * (|\Gamma(i)| - 1)} * \sum_{m,n \in \Gamma(i)} \frac{w(i,m) + w(i,n)}{2 * \sum_{z \in \Gamma(i)} \frac{w(i,z)}{|\Gamma(i)|}} \tag{4.12}$$

In this study, Cluster coefficient of node x and y is calculated by summing of each other's Cluster coefficient separately. Thus:

$$CC(x,y) = CC(x) + CC(y) \tag{4.13}$$

Figure 4.5. contains an additional node H and its connections and another connection between node F and G compared to Figure 4.2. In calculation of CC(X,Y), there are three different triangles in this network. Two triangles (HXE and EXA) will be included in Node X's CC calculation and one triangle (FYG) will be included in Node Y's CC calculation. Among the Node X's neighbors, only Node E, Node A and Node H are the part of a clique

(triangles), which are HXE and EXA. Besides this, among the Node Y's neighbors only Node F and G are the part of a triangle, which is FYG.

- Unweighted CC(X) = 2×2/(6×5)
  = 4/30
  = 0,13
- Unweighted CC(Y) = 2×1/(6×5)
  = 2/30
  = 0,06
- Unweighted CC(X,Y) = CC(X) + CC(Y)
  = 0,19
- Weighted CC(X) = (1/30) × ((w(XE) + w(XH) + w(XA) + w(XE)**)** / 2 × ((w(XH) + w(XE) + w(XA) + w(XB) + w(XC) + w(XD)) / 6))
  = 0,09
- Weighted CC(Y) = (1/30) × ((w(YF) + w(YG)) / 2 × ((w(YF) + w(YG) + w(YA) + w(YB) + w(YC) + w(YD)) / 6))
  = 0,03
- Weighted CC(X,Y) = CC(X) + CC(Y)
  = 0,09 + 0,03
  = 0,12

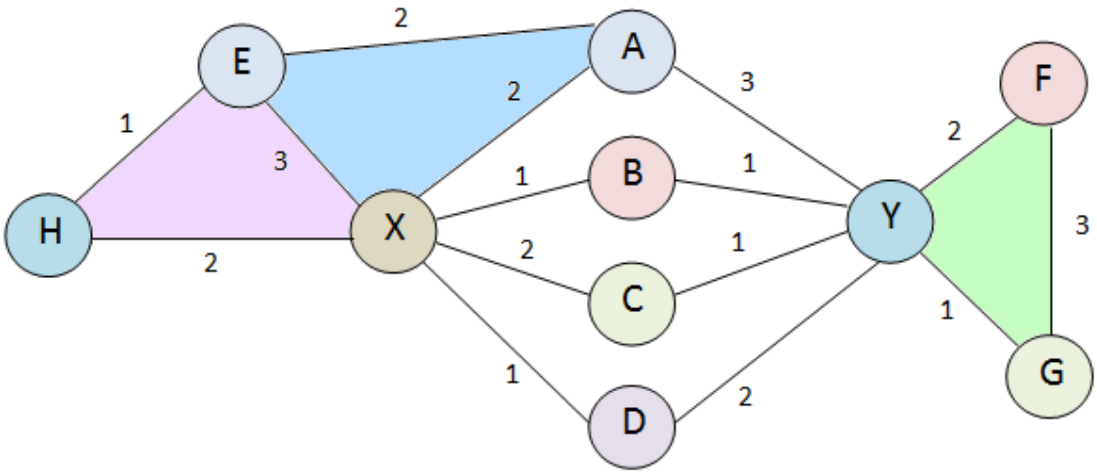

Figure 4.5. A small weighted network for Clustering Coefficient calculation.

5. EXPERIMENTS AND RESULTS

In this section, social network data, which are used in supervised learning and fuzzy rule based algorithm is explained and also preparing the dataset and how this dataset is used in the related algorithms is explained. Then supervised learning algorithms, which includes fuzzy rule based are presented before performance evaluation of experiments results. Finally the unsupervised strategy and its results are at the end of the section.

5.1. Social Network Data

One of the social networks, which is used in the experiment is the computer science co-authorship network and second one is the eye disease co-authorship network. In the computer science co-authorship network, time interval is between 2006 and 2010 for training the data. It consists of 7267 distinct authors and 32840 connections in that time interval. On the other hand, in the eye disease co-authorship network, time interval is between 2002 and 2004 for training the data. It consists of 8101 distinct authors and 88186 connections in that time interval.

Three versions of the networks are used in this study:

- A weighted version of the network: In this version each link between a pair of authors is weighted by the number of papers, which are co-operated by two authors.
- Another weighted version of the network: In this version, each link between a pair of authors is weighted by contribution of the authors who involve in the related paper. If a paper has n authors, the contribution of each author for the paper is calculated by 1/(n-1) [5]. The total link weight is calculated by summing of all contributions between each author. This type of link weight shows how exclusive is the relationship between the authors.
- An unweighted version of the social network: In this version, a link is created if two authors co-operate at least one paper.

5.2. Dataset for Link Prediction

In this study, below procedures are used to produce a labeled dataset for supervised learning algorithms. It is considered that evaluation of the networks on time is recorded. It means that from the beginning, each node and each link are recorded in the graph. Each social network is dived into two states. The first state is up to t and the second state is from t to t'. Using the information from state1 (from 0 to t), link prediction is made in satete2 (from t to t'). Nodes in the calculation are exist both state1 and state2. These states are shown In Figure 5.1.

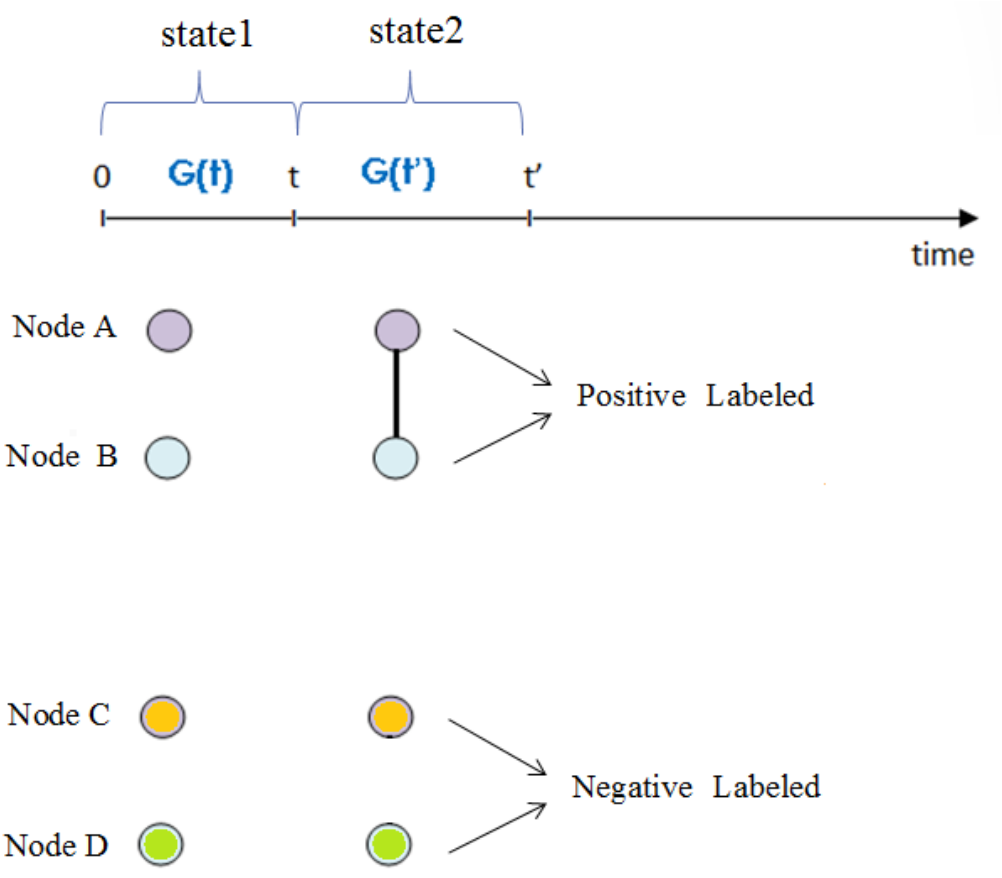

Figure 5.1. Subgraphs in time intervals for each networks.

In this study, an undirected graph G(V,E) is used for both networks where V is set of nodes (vertices) and E is a set of labeled edges. Nodes in the calculation are exist in both G(t) and G(t'). This graph G(V,E) consists of two sub graphs:

- G(t) contains edges and nodes, which are recorded until t and these nodes don't have connections in G(t) and
- G(t') contains edges and nodes, which are recorded until t' and some connected nodes exist in G(t') where t'>t.

In order to generate training examples, specific pair of nodes, which they don't have a connection in G(t) are included in training set with their predictor attributes and class labels (positive or negative). If an unconnected pair of node in G(t) has a connection in G(t'), this pair of node is labeled as positive but if an unconnected pair of node in G(t) still remains unconnected in G(t'), this node pair is labeled as negative. In Figure 5.1., Node A and B has a connection in G(t'), so they are labeled as positive but Node C and D has not a connection in G(t), so they are labeled as negative.

5.2.1. Computer Science Co-authorship Dataset

In predicting experiment, two sub graphs are used for the computer science co-authorship network (CSCN). The first sub graph is G(t), which holds information between 2006 and 2010 (2010 is not included). In G(t) all nodes and connections are observed and recorded. Besides, in G(t) all information is extracted by using metric values, which are define in metric section. The second sub graph is G(t'), which holds information from 2010 to 2014 (2014 is not included) as seen in Figure 5.2. The task is that, predicting the new connections occurred from 2010 to 2014 with 6supervised learning algorithms, which include one fuzzy rule based algorithm by using the extracted information in G(t). In G(t), 2048 unconnected pairs of authors are randomly defined and their features are extracted in order to use in training phase. Those 2048 nodes are also exist in G(t'). According to situation (connected or not) of those nodes in G(t'), label of 1024 pairs of nodes are signed as positive (connected in G(t')) and label of remain 1024 pairs of nodes are signed as negative (unconnected in G(t')). Hence, classes are distributed equally and the default accuracy of classification is 50 %. Three different datasets, which contains those 2048 authors are created and this class distribution is applied for all three datasets, which are described in social network data section. To summarize:

- G(t) contains information between 2006 and 2010. G(t') contains information from 2010 to 2014.
- 2048 unconnected pairs of authors are defined in G(t). Half of those authors are connected in G(t') and another half remains unconnected in G(t').
- Those authors are exist in both G(t) and G(t').
- 1024 connected authors' label, which have a connection in G(t') signed as 1 and remain is signed as 0.
- Then those 2048 authors are trained according to their extracted metric values with using supervised learning algorithms, which includes one fuzzy rule based algorithm.

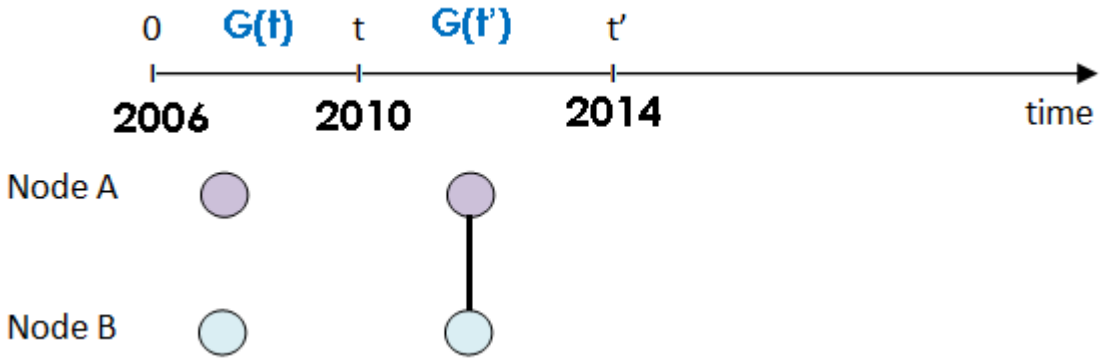

Figure 5.2. Subgraphs in time intervals for the CSCN.

5.2.2. The Eye Disease Co-authorship Dataset

Preparing the graphs in the eye disease co-authorship network (EDCN) is similar with the computer science co-authorship network. Only number of pairs and time intervals are different from the computer science co-authorship network. The eye disease co-authorship network contains two sub graphs G(t) and G(t') as seen in Figure 5.3. 400 pairs of nodes are selected. Half of this pairs is signed as negative because they don't have a connection in G(t'), remain is signed as positive because they have a connection in G(t'). Classes are also distributed equally. Finally 400 authors' class distribution is applied for all three datasets, which are described in social network data section.

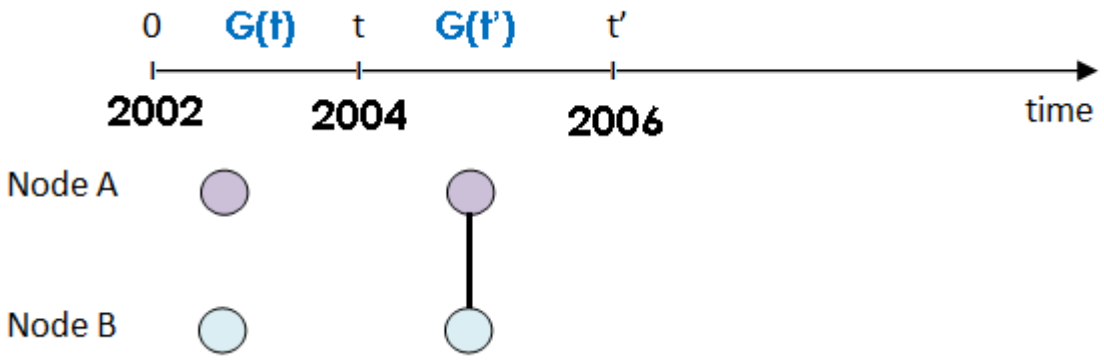

Figure 5.3. Subgraphs in time intervals for the EDCN.

In the link prediction task, WEKA tool is used. WEKA contains many different algorithms for classification problems. In this study six supervised learning algorithms, which include one fuzzy rule based algorithm are applied for the prediction in the datasets and in turn, those six algorithms are compared with each other according to their performances. During the training session with the algorithms, default parameters of WEKA experimenter module are used. In order to increase the reliability, every dataset run at ten times with 10-fold-cross validation. In WEKA, number (n) of cross validation can be set manually. During the cross validation process, dataset is reordered randomly and dived into n folds equally. In each iteration, one fold is used for testing and rest of folds (n-1) is used for training. At the end of the training, all test results are collected and averaged over all folds. After the training, a t-test is applied with the significance level of 95% in order to compare using algorithms.

5.3. Algorithms

In this study some supervised algorithms [34] , which are NaivesBayes, J48, IBK, LibLinear, LibSVM and FURIA were used. FURIA is a fuzzy rule based algorithm. Those algorithms will be explained in detail below sections. In supervised learning, an input set and its correspondent label (output) are required for every instance. Hence, metric values were used as input values and existence of a link (have a link or not) was used as an output in training phase. Metric values, which are obtained from every unconnected couple of nodes were signed as 1 or 0 according to their future relationship. If those unconnected

couples had a relationship in feature, they were marked as 1. Otherwise they were marked as 0. For all algorithms, input set was prepared like this. Then by using WEKA tool, training set was obtained for each algorithm. Then a t-test was applied with significance level of 95% for each one. Prediction and training processes are depicted in Figure 5.4.

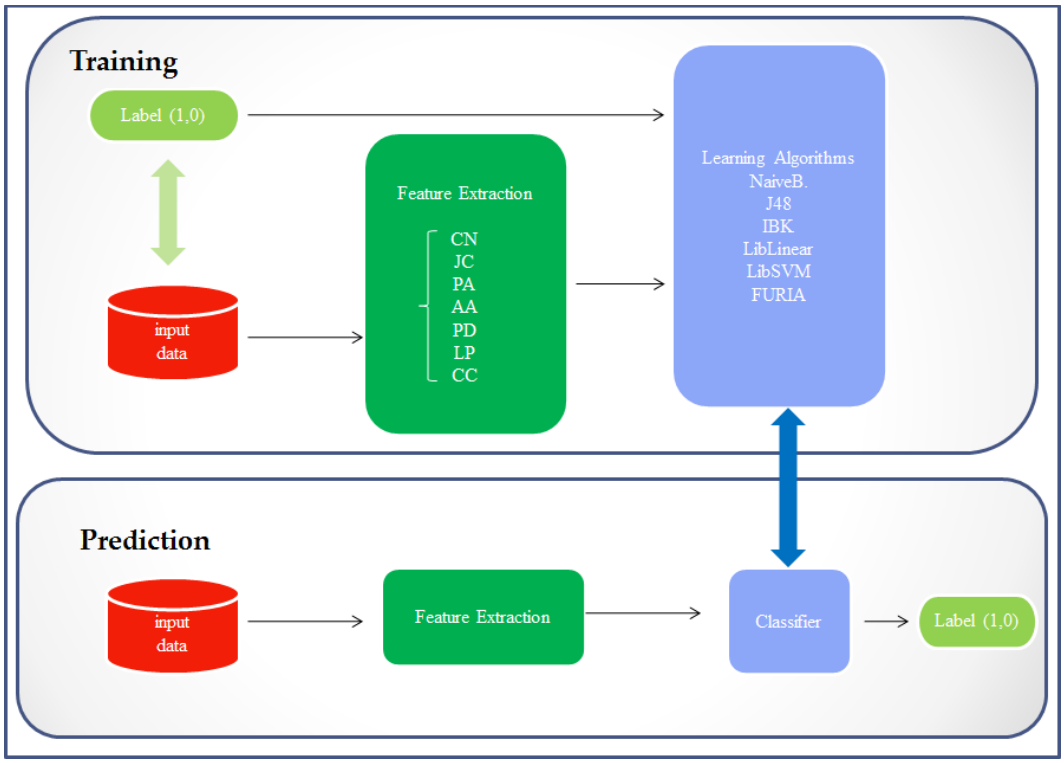

Figure 5.4. Prediction Process.

5.3.1. NaivesBayes

Implementation of the Naive Bayes classifier, which is a probabilistic classifier based on applying Bayes' theorem [35]. Given a hypothesis *h* and data *D*, which bears on the hypothesis:

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)} \tag{5.1}$$

Where *P(h)*: independent probability of *h*; *P(D)*: independent probability of *D*; *P(D|h)*: conditional probability of *D* given *h; P(h|D)*: conditional probability of *h* given *D.* This formula gives a probability by counting the frequency of values in the historical data.

In Naive Bayes classification, assumption is that all input attributes are conditionally independent. It means that there is no dependent relation between attributes. This algorithm has many advantages. First advantage is that preparing the training phase is simple. In training phase, a table of probability of each variable according to their classes is prepared. Besides, testing phase is straightforward also. It is enough to look up the table or calculating the conditional probabilities. Another advantage is that NaiveBayes classifier's performance is competitive to most of the classifier.

5.3.2. J48

J48 is an open source Java implementation of the C4.5 algorithm in the WEKA data mining tool. C4.5 builds decision trees from a set of training data and it is an extension of the ID3 algorithm [36]. It uses the concept of information entropy. While building the tree, at each node of the tree, C4.5 chooses the attribute of the data that most effectively splits its set of samples into subsets. The splitting criterion is the normalized information gain, which is difference in entropy. The attribute with the highest normalized information gain is chosen to make the decision. Algorithm then recurs on the smaller sub-lists. To solve the over-fitting problem, C4.5 goes back through the tree once it's been created and attempts to remove branches that do not help by replacing them with leaf nodes. One of the advantages of this algorithm is that it can handle training data with missing attribute values. Furthermore, C4.5 can handle attributes with discrete and continuous (numeric) values. On the other hand, ID3 does not apply any pruning procedures nor does it handle numeric attributes or missing values.

5.3.3. IBK

IBK is an instanced based k-nearest-neighbor classifier [37, 38]. A distance function used is a parameter of the search method (Euclidean, Chebyshev, etc.). Predictions from

more than one neighbor can be weighted according to their distance from the test instance. IBK does not learn anything from the training data and simply uses the training data itself for classification (so called lazy). The testing phase for a new instance 't', given a known set 'I' is as follows:

- Compute the distance between ’t’ and each instance in 'I‘ in training data.
- Sort the distances in increasing numerical order and pick the first 'k' elements
- Compute and return the most frequent class in the 'k' nearest neighbors, optionally weighting each instance's class.

The main advantage of K-Nearest Neighbor (KNN) Classifier is that: it is a very simple classifier that works well on basic recognition problems. The main disadvantage of this approach is that the algorithm must compute the distance and sort all the training data at each prediction, which can be slow if there are a large number of training examples. Another disadvantage of this approach is that the algorithm does not learn anything from the training data, which can result in the algorithm not generalizing well. Further, changing K can change the resulting predicted class label.

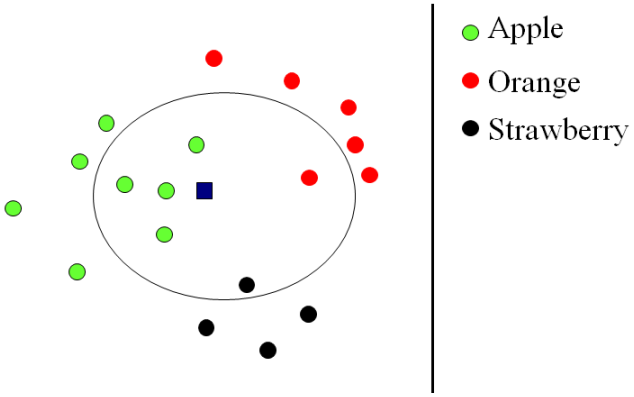

Figure 5.5. A KNN example (k=6). Blue rectangle belongs to Apple class.

5.3.4. LibLinear

It's a library for support vector machines and it exists in WEKA. LibLinear implements linear Support Vector Machines (SVMs) [39, 40]. In this algorithm, goal is to find the discriminator that maximizes the margins. The linear discriminator function (classifier) with the maximum margin is the best. Margin is defined as the width that the boundary could be increased by before hitting a data point. By using Lagrangian function, the linear discriminator function is obtained.

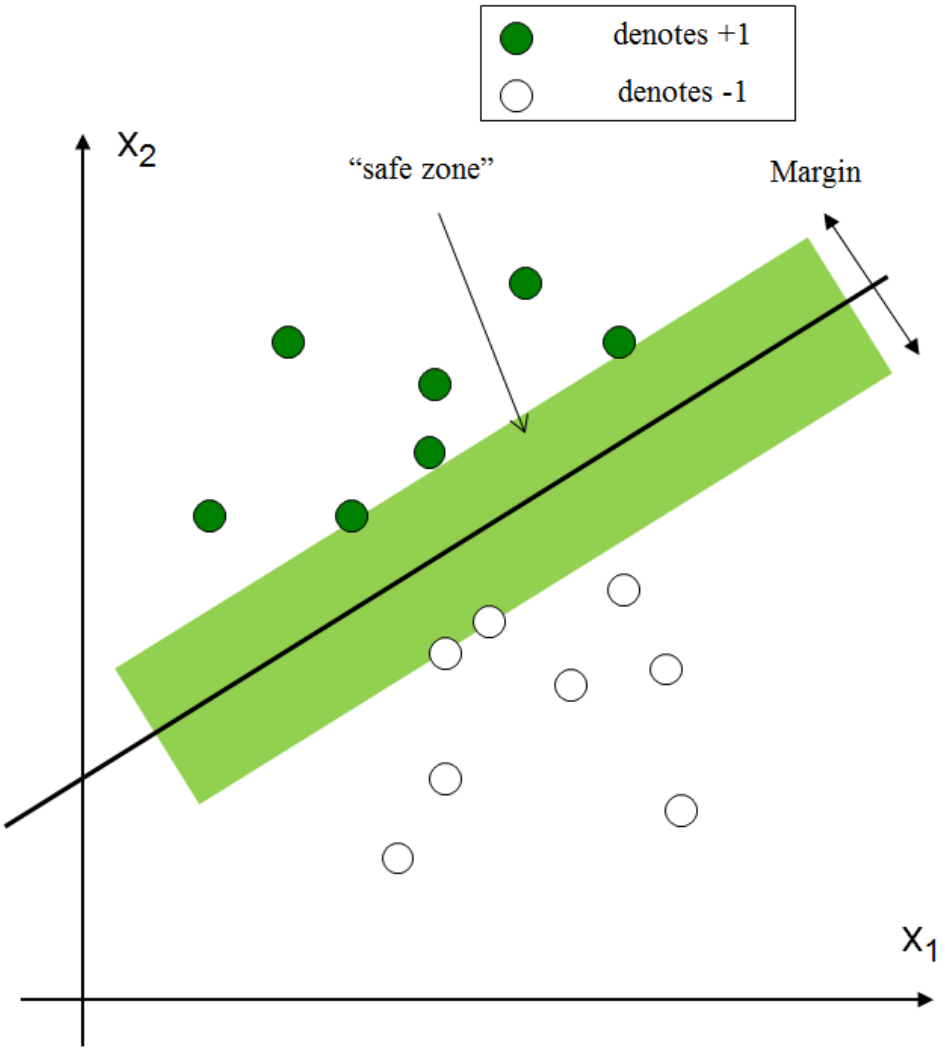

Figure 5.6. Example of Linear SVM.

5.3.5. LibSVM

LibSVM is an implementation of the support vector machines and one of the algorithms in WEKA. LIBSVM implements the SMO (Sequential minimal optimization) algorithm for kernelized support vector machines (SVMs) [41]. Sequential minimal optimization (SMO) is an algorithm for solving the quadratic programming (QP) problem that arises during the training of support vector machines. The general idea in this algorithm is that the original input space can always be mapped to some higher-dimensional feature space where the training set is separable. To obtain discriminator function, a kernal function (linear, polynomial of power, Gaussian, Sigmoid etc.) is used.

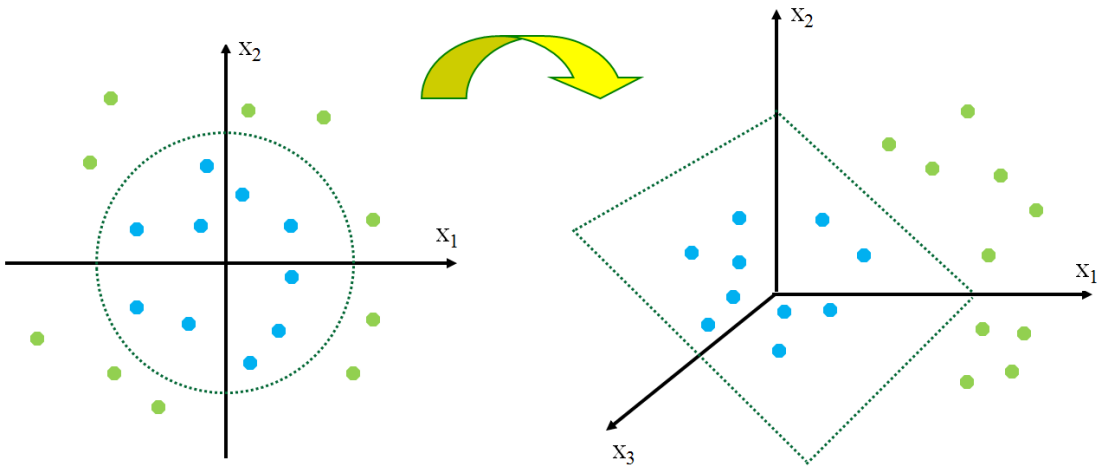

Figure 5.7. Conversion of original input space to high dimensional space.

5.3.6. RIPPER

RIPPER stands for Repeated Incremental Pruning to Produce Error Reduction. It is a classification algorithm designed to generate rules set directly from the training dataset, so it is a rule based algorithm [42]. It learns rules for positive class. On the other hand, negative class will be default class. Algorithm starts from empty rule and adds conjuncts that maximizes FOIL's (First Order Inductive Learner is a rule-based learning algorithm.) information gain measure and it stops when rule no longer covers negative examples. It means that the training set cannot be split any further. Furthermore, rule pruning process is applied by removing one of the conjuncts in the rule. Before and after pruning, error rate

on validation set is compared. If error improves, related conjunct is pruned. One of the advantages of Rule Based Classifiers is that it is as highly expressive as decision trees. Another advantage is that it can classify new instances rapidly. Finally it can easily handle missing values and numeric attributes.

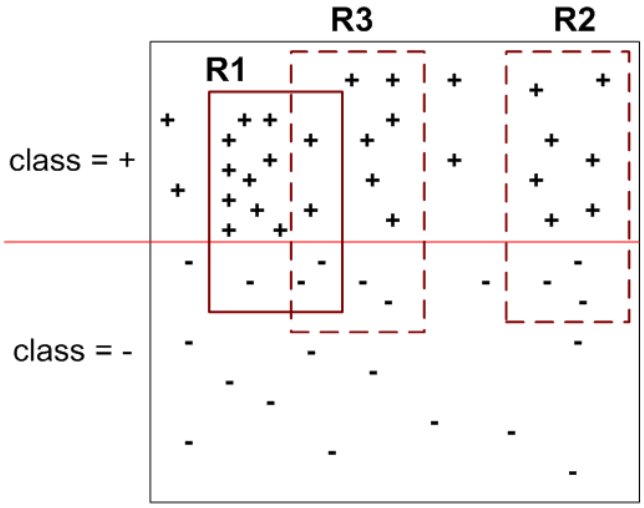

Figure 5.8. Rule examples of RIPPER.

5.3.7. FURIA

FURIA stands for Fuzzy Unordered Rule Induction Algorithm [43]. FURIA is a fuzzy rule learner based on the well-known RIPPER algorithm. Conventional (non-fuzzy) rules produce models with "sharp" decision boundaries and sudden transitions between different classes. On the other hand, fuzzy rules have "soft" boundaries, which is one of their main characteristics. In FURIA, fuzzy interval (I^F) specified by four parameters ($\phi^{s,L}$, $\phi^{c,L}$, $\phi^{c,U}$, $\phi^{s,U}$) and calculated as below:

$$I^F(v) = \begin{cases} 1 & \phi^{c,L} \le v \le \phi^{c,U} \\ \dfrac{v - \phi^{s,L}}{\phi^{c,L} - \phi^{s,L}} & \phi^{s,L} < v < \phi^{c,L} \\ \dfrac{\phi^{s,U} - v}{\phi^{s,U} - \phi^{c,U}} & \phi^{c,U} < v < \phi^{s,U} \\ 0 & else \end{cases} \tag{5.2}$$

$\phi^{c,L}$ and $\phi^{c,U}$, represent lower and upper bound of the core of the fuzzy set and element's membership is 1. On the other hand, $\phi^{s,L}$ and $\phi^{s,U}$ represent lower and upper bound of the support and element's membership is greater than 0. It is seen at Figure 5.9. [43].

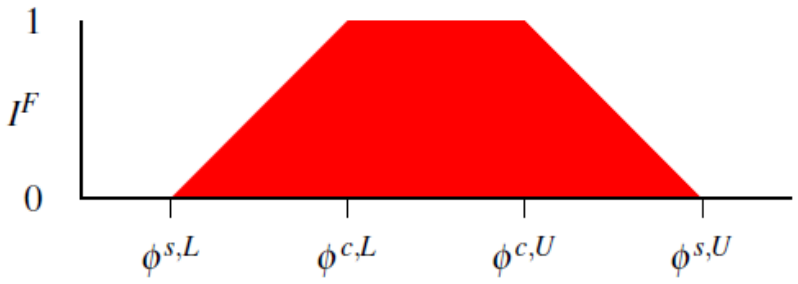

Figure 5.9. A fuzzy interval $\boldsymbol{I^F}$.

Algorithm in FURIA is similar with RIPPER. Essentially, the main difference with RIPPER is that sharp boundaries of a rule are replaced by "soft" boundaries. The main task is deciding the optimal soft boundaries for every rule, which is in RIPPER. The idea is fuzzify the final rules in RIPPER algorithm. Algorithm is trying to find the best fuzzy extension of each rule, which have same structure in RIPPER but intervals are replaced by fuzzy intervals. In Figure 5.10. [43], possible boundaries are shown.

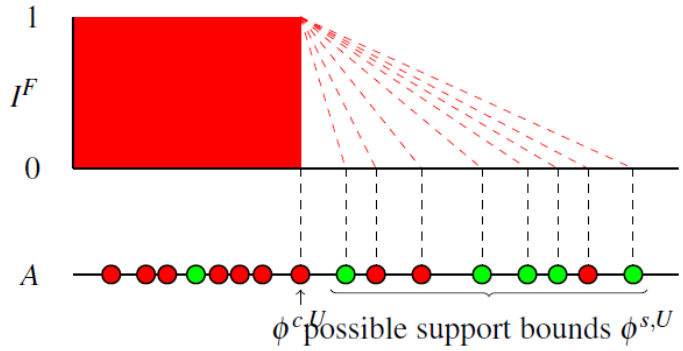

Figure 5.10. Possible support bounds.

5.4. Results and Performances of the Supervised Classifiers

There are different ways to measure the performance of a classifier. Accuracy, precision, recall, F-measure and Area Under ROC Curve performance measures were used in this study. Then these performance measures were compared with each other by using three type networks (unweighted network, weighted network1, weighted network2) for both the computer science and the eye disease co-authorship networks. Headings in tables were labeled as unweighted network, weighted network1 and weighted network2 in order to make easy to read and understand. Network1 means the network, which regards the contribution of authors. On the other hand, Network2 means the network, which regards the number of papers of authors.

As a result of the experiments, it is seen that: in almost all comparisons between the networks, at least one of the weighted networks has equal results with unweighted networks or better results than unweighted networks. Furthermore, in most cases, among the computer science and the eye disease co-authorship weighted networks, Network1 has same performance with Network2 or better performance than Network2.

5.4.1. Accuracy

The simplest way is counting the correctly predicted examples in an unseen test dataset to measure the performance. This value is known as *accuracy* and accuracy values, which are obtained by the evaluated algorithms was compared to default accuracy value (50%) in order to verify the result, which is provided by the supervised link prediction algorithms. All results were verified by t-test (at 95% of confidence) for all algorithms and datasets.

$$accuracy = \frac{number\ of\ true\ positives\ +\ number\ of\ true\ negatives}{number\ of\ true\ pos. + false\ pos. + false\ neg. + true\ neg.} \qquad (5.3)$$

In order to compare all algorithms' performances in terms of the accuracy, results are shown in Table 5.11. and 5.12. for the computer science and the eye disease co-authorship

network respectively. In Table 5.11., J48 has the best accuracy rate with 75,09%. On the other hand LibSVM has the highest rate with 63,05% in Table 5.12. As it can be seen, all results in both co-authorship networks are higher than default accuracy, which is 50%. It means that useful knowledge can be obtained from these datasets.

In order to understand of the effect of using weighted networks, accuracy results give important information for both two networks. As it is seen in Table 5.1. except NaiveBayes, all other algorithms give better performance in at least one weighted networks. This determination is clearer in the eye disease co-authorship network, which is shown in Table 5.2. There, all accuracy results in at least one weighted network are equal to or greater than unweighted networks. As a result, in almost all comparisons, weighted networks show better performance in comparison to unweighted networks for accuracy calculation. This result will be valid for the other performance measures in the rest of the experiments.

Table 5.1. Accuracy rates obtained by the algorithms for the CSCN.

Algorithm/Dataset	**Unweighted Network**	**Weighted Network1**	**Weighted Network2**
NaiveBayes	66,29%	65,64%	64,17%
LibLinear	70,52%	74,58%	69,59%
LibSVM	73,94%	74,79%	73,66%
Ibk	71,40%	70,82%	71,73%
Furia	74,72%	74,83%	74,73%
J48	74,59%	74,48%	75,09%

Table 5.2. Accuracy rates obtained by the algorithms for the EDCN.

Algorithm/Dataset	Unweighted Network	Weighted Network1	Weighted Network2
NaiveBayes	62,00%	62,73%	62,65%
LibLinear	57,70%	62,33%	56,55%
LibSVM	56,18%	63,05%	55,78%
Ibk	55,45%	60,50%	57,10%
Furia	60,73%	62,05%	59,68%
J48	62,75%	62,50%	62,75%

5.4.2. Precision and Recall

Precision and Recall values are another evaluation measures. Precision is the number of true positives divided by the total number of elements labeled as belonging to the positive class. Recall is the number of true positives divided by the total number of elements that actually belong to the positive class. They are formulated as below:

$$Precision = \frac{tp}{tp + fp} \tag{5.4}$$

$$Recall = \frac{tp}{tp + fn} \tag{5.5}$$

Precision rates obtained by the algorithms for the computer science and the eye disease co-authorship networks are shown in Table 5.3. and 5.4. respectively. In the computer science co-authorship network LibSVM has the best performance with 71,00 % in both unweighted and the weighted network2. Then FURIA has the second performance with 70,00 % after de LibSVM by using weighted network2. However, for the eye disease co-authorship network FURIA has the best performance with 60,00 % among other algorithms by using weighted network1. LibSVM and Ibk have same value with 59,00 % by using weighted network1 and come after FURIA.

Table 5.3. Precision rates obtained by the algorithms for the CSCN.

Algorithm/Dataset	Unweighted Network	Weighted Network1	Weighted Network2
NaiveBayes	61,00%	60,00%	59,00%
LibLinear	68,00%	69,00%	68,00%
LibSVM	71,00%	70,00%	71,00%
Ibk	69,00%	67,00%	69,00%
Furia	69,00%	69,00%	70,00%
J48	69,00%	69,00%	69,00%

Table 5.4. Precision rates obtained by the algorithms for the EDCN.

Algorithm/Dataset	Unweighted Network	Weighted Network1	Weighted Network2
NaiveBayes	57,00%	58,00%	57,00%
LibLinear	56,00%	58,00%	52,00%
LibSVM	56,00%	59,00%	55,00%
Ibk	55,00%	59,00%	57,00%
Furia	57,00%	60,00%	57,00%
J48	58,00%	57,00%	58,00%

Recall rates obtained by the algorithms for both the computer science and the eye disease co-authorship networks are shown in Table 5.5. and 5.6. respectively. NaiveBayes and J48 have the best results with 92,00 % in comparison to the other algorithms by using weighted networks for the computer science co-authorship network. These two algorithms have the best performances in the eye disease co-authorship network, as well. Among the algorithms in the eye disease co-authorship network, only FURIA gets better performance with 84,00 % in the unweighted network in comparison to FURIA's both weighted values.

Table 5.5. Recall rates obtained by the algorithms for the CSCN.

Algorithm/Dataset	Unweighted Network	Weighted Network1	Weighted Network2
NaiveBayes	91,00%	92,00%	92,00%
LibLinear	82,00%	90,00%	81,00%
LibSVM	82,00%	88,00%	81,00%
Ibk	78,00%	81,00%	79,00%
Furia	88,00%	89,00%	88,00%
J48	90,00%	89,00%	92,00%

Table 5.6. Recall rates obtained by the algorithms for the EDCN.

Algorithm/Dataset	Unweighted Network	Weighted Network1	Weighted Network2
NaiveBayes	99,00%	99,00%	99,00%
LibLinear	69,00%	95,00%	61,00%
LibSVM	64,00%	90,00%	66,00%
Ibk	57,00%	71,00%	61,00%
Furia	84,00%	79,00%	80,00%
J48	99,00%	98,00%	99,00%

5.4.3. F-Measure

F-measure is the harmonic mean of precision and recall and it balances two measures.

$$F = 2.\frac{precision.recall}{precision + recall} \tag{5.6}$$

According to Table 5.7., which shows F-measure rates obtained by the algorithms for the computer science co-authorship network, J48 has the best result with 0,79. It is clear to see that weighted results are equal to or better than unweighted results for all algorithms in this

table. On the other hand, NaiveBayes and J48 have the best performances with 0,73 by using the eye disease weighted network as seen in Table 5.8.

Table 5.7. F-measure rates obtained by the algorithms for the CSCN.

Algorithm/Dataset	Unweighted Network	Weighted Network1	Weighted Network2
NaiveBayes	0,73	0,73	0,72
LibLinear	0,74	0,78	0,73
LibSVM	0,76	0,78	0,76
Ibk	0,73	0,74	0,74
Furia	0,78	0,78	0,78
J48	0,78	0,78	0,79

Table 5.8. F-measure rates obtained by the algorithms for the EDCN.

Algorithm/Dataset	Unweighted Network	Weighted Network1	Weighted Network2
NaiveBayes	0,72	0,73	0,73
LibLinear	0,54	0,72	0,48
LibSVM	0,59	0,71	0,59
Ibk	0,56	0,64	0,58
Furia	0,68	0,67	0,66
J48	0,73	0,72	0,73

5.4.4. Area Under ROC Curve

Area Under ROC Curve (AUC) is another important performance measure, which is related with sensitivity (true positive rates) and specificity (true negative rate). AUC rates obtained by the algorithms for the computer science co-authorship network are shown in Table 5.9. NaiveBayes and J48 have the best results with 0,76 among all algorithms in this table. Except J48, in most cases at least one weighted network reaches better result in comparison with unweighted network for the computer science co-authorship network.

Similarly, Except J48 and NaiveBayes, in most cases at least one weighted network has better result in comparison with unweighted network for the eye disease co-authorship network, which is shown in Table 5.10.

Table 5.9. AUC rates obtained by the algorithms for the CSCN.

Algorithm/Dataset	Unweighted Network	Weighted Network1	Weighted Network2
NaiveBayes	0,75	0,76	0,75
LibLinear	0,71	0,75	0,70
LibSVM	0,74	0,75	0,74
Ibk	0,70	0,70	0,71
Furia	0,75	0,75	0,75
J48	0,76	0,75	0,75

Table 5.10. AUC rates obtained by the algorithms for the EDCN.

Algorithm/Dataset	Unweighted Network	Weighted Network1	Weighted Network2
NaiveBayes	0,63	0,62	0,62
LibLinear	0,58	0,62	0,57
LibSVM	0,56	0,63	0,56
Ibk	0,55	0,65	0,56
Furia	0,63	0,64	0,61
J48	0,63	0,63	0,63

As a result of experiments, when all algorithms are considered it is seen in Table 5.11. and 5.12., J48 has the best performance among all algorithms with six times at total in both the computer science and the eye disease co-authorship networks. Then NaiveBayes comes with four times. It is interesting that the linear support vector machine algorithm (LibLinear) cannot have a best performance even one time. Hence, it can be said that using a linear support vector in order to separate classes gets lower performance in comparison to other algorithms in these networks. On the other hand, using a kernelized support vector machine (LIBSVM) gives better results in comparison to linear support vector machine (LibLinear) in this study. LIBSVM has the best performance two times in

total. Similar to LibLinear, the k-nearest algorithm (Ibk) also has a lower performance. Classification process by using the k nearest cannot obtain the highest performance as J48 and NaiveBayes algorithms.

One of the aims of this study is to see that effects of a fuzzy logic in the link prediction. According to results, FURIA (fuzzy rule based algorithm) has the best performance only one time. Hence, it can be said that fuzzy logic does not increase the performance in this study as much as J48 and NaiveBayes. On the other hand, the probabilistic algorithm (J48) and the decision tree algorithm (J48) have better performances than the other algorithms to predict the links.

Table 5.11. Best performance measures according to algorithms for the CSCN.

Algorithm/Performance Measure	Accuracy	Precision	Recall	F-Measure	AUROC
NaiveBayes			*		*
LibLinear					
LibSVM		*			
Ibk					
Furia					
J48	*		*	*	*

Table 5.12. Best performance measures according to algorithms for the EDCN.

Algorithm/Performance Measure	Accuracy	Precision	Recall	F-Measure	AUROC
NaiveBayes			*	*	
LibLinear					
LibSVM	*				
Ibk					*
Furia		*			
J48			*	*	

To summarize results of the supervised learning algorithms, which contain one fuzzy rule based algorithm:

- In almost all comparisons between the networks, at least one of the weighted networks has equal results with unweighted networks or better results than unweighted networks. This is valid for both the computer science and the eye disease co-authorship networks.
- In most cases, among the both the computer science and the eye disease weighted networks, network1 (regarding contributions of authors) has the same performance with network2 (regarding number of paper) or has better performance than network2.
- In this study, the probabilistic algorithm (NaiveBayes) and the decision tree algorithm (J48) have better results than other algorithms. On the other hand, the fuzzy rule based algorithm (FURIA) cannot obtain the good performance as much as J48 and NaiveBayes.

5.5. Experiments with the Unsupervised Strategy

One of the aims of this project is to compare the supervised and unsupervised link prediction strategies. In the unsupervised strategy (it is not a kind of learning method), usually one metric value is chosen and then this metric is ranked. In this method to decide a future link, a threshold value is determined and then higher values from the threshold are assigned as connected and lower values from that threshold are assigned as unconnected pairs. In this study, instead of using only one metric, all metrics are used together with their weighted versions for ranking [5]. First of all for each metric value, their minimum and maximum values are calculated. Then according to these minimum and maximum values, a rank value is assigned for every metric value. Finally, a total resulting ranking value is provided for each instance.

5.6. Results of the Unsupervised Strategy

In this section, the unsupervised strategy's results are represented both the computer science and the eye disease co-authorship networks. As performance measures, precision, recall and F-measure are used. Each graph contains three different colored lines:

- Green lines implies weighted network, which regards contributions of authors.
- Blue lines implies weighted network, which regards number of papers.
- Red lines implies unweighted network.

5.6.1 Result of the Computer Science Co-authorship Network

It is seen that for every performance measure in the unsupervised strategy, at least one of the weighted networks has better performance than unweighted network at most of the point for especially Recall and F-measure rates in Figure 5.11., 5.12. and 5.13. As default of accuracy is 50 %, minimum precision value is 0,5 in both networks. Top ranked value has the greatest precision value with 1. Then as ranked value decrease, precision values starts to decrease as well. The first top ranked values has maximum recall value with 1. Then recall values starts to decrease down to 0. On the other hand, maximum F-measure value is around 66 %. All others algorithms for weighted and unweighted networks in the supervised method (in Table 5.7.) have better result than the unsupervised strategy in F-measure calculation. It is almost valid also for the eye disease co-authorship network. Hence these results show that supervised method has better results than the unsupervised strategy.

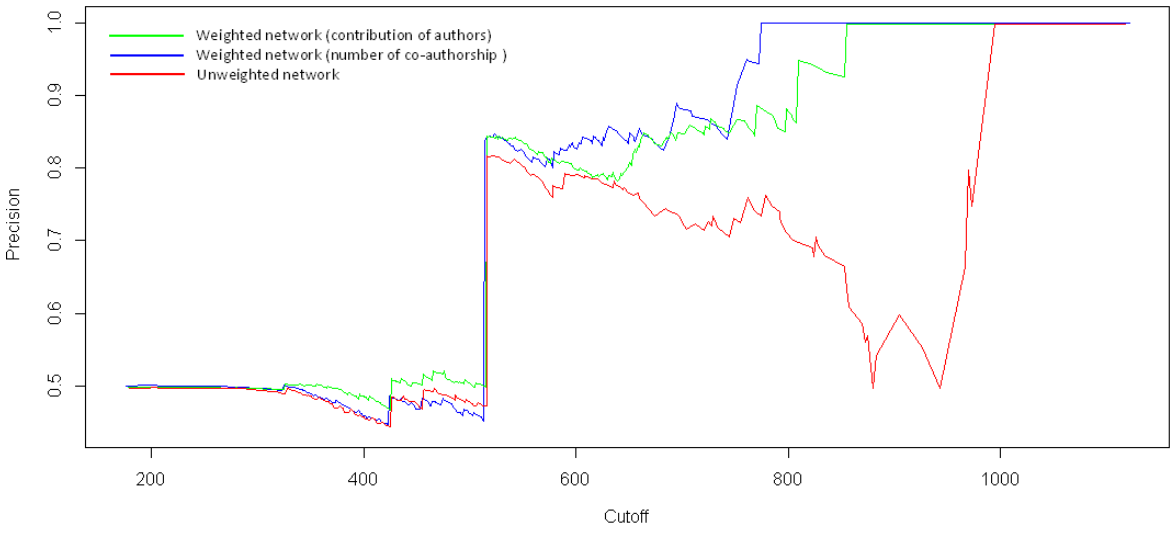

Figure 5.11. Precision rates of the CSCN obtained by the unsupervised strategy.

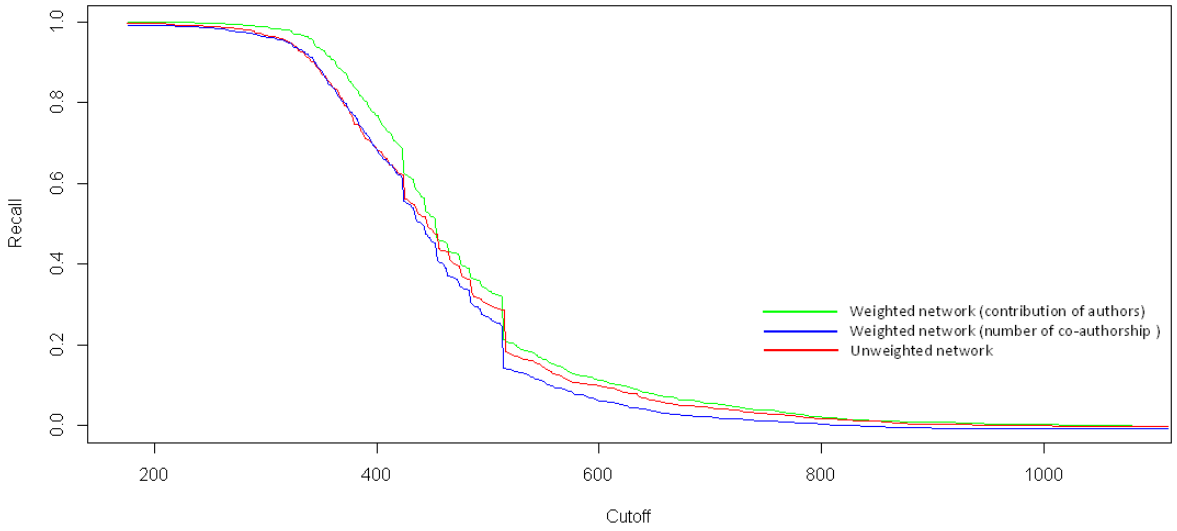

Figure 5.12. Recall rates of the CSCN obtained by the unsupervised strategy.

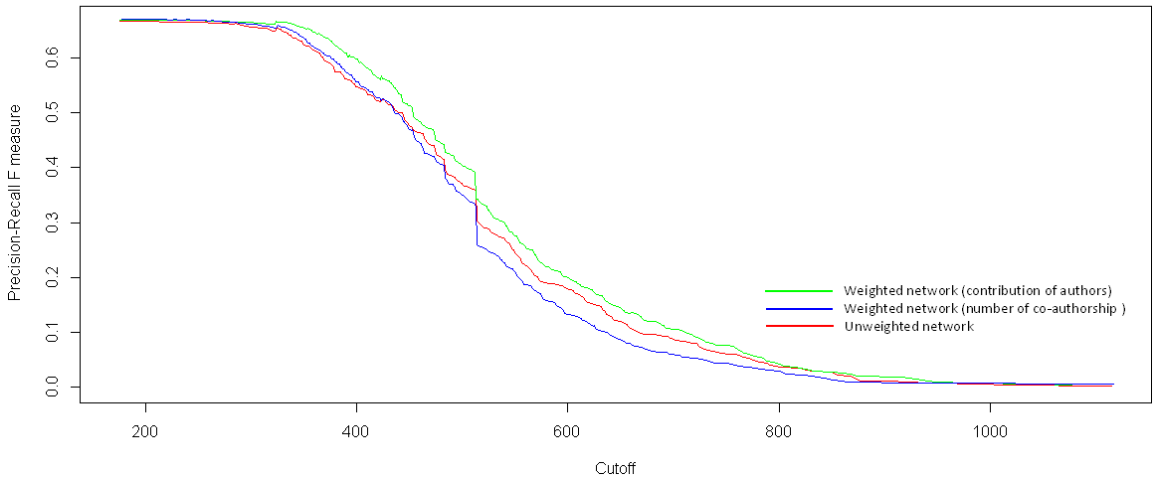

Figure 5.13. F-measure rates of the CSCN obtained by the unsupervised strategy.

5.6.2 Result of the Eye Disease Network

As default of accuracy is 50 %, minimum precision value is 0,5 in this network for three types. Around 40 % of top ranked value has greatest precision value with 1 in Figure 5.14. Then as ranked value decrease, precision values starts to decrease down to 0,5 as well. The first top ranked values has maximum recall value with 1 in Figure 5.15. Then

recall values start to decrease down to 0. For the eye disease co-authorship network, it is seen that at least one weighted network shows better performance than unweighted network at most of the points as being in the computer science co-authorship network. In this network, maximum F-measure value is around 66 % as it is seen in Figure 5.16. Only supervised IBK algorithm in the eye disease network (Table 5.8.) is lower than 66 %. Except IBK, all others algorithms for at least one weighted network in the supervised method have better result than the unsupervised strategy in F-measure calculation in this network.

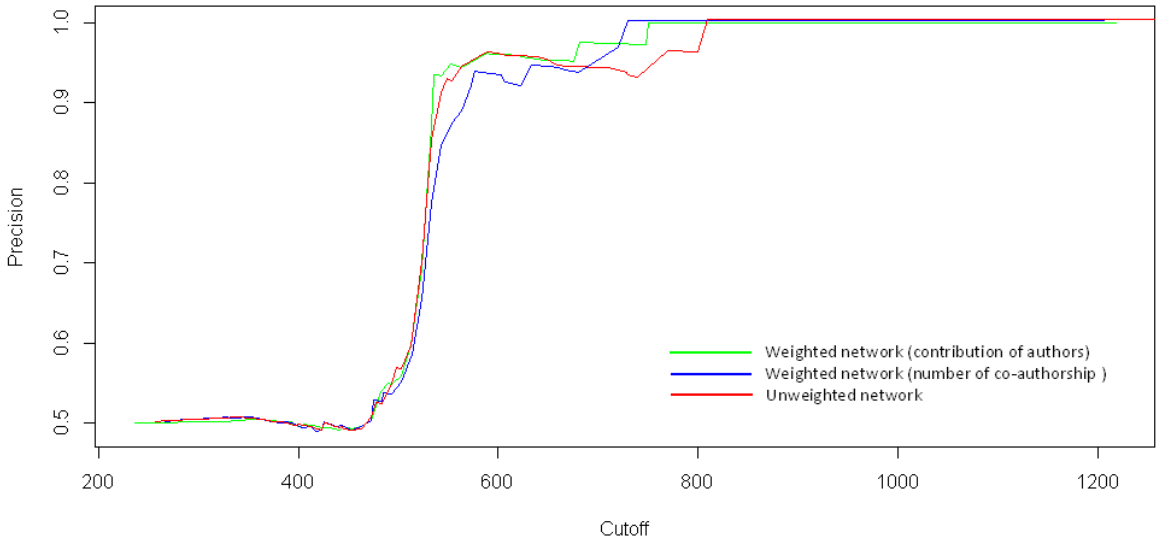

Figure 5.14. Precision rates of the EDCN obtained by the unsupervised strategy.

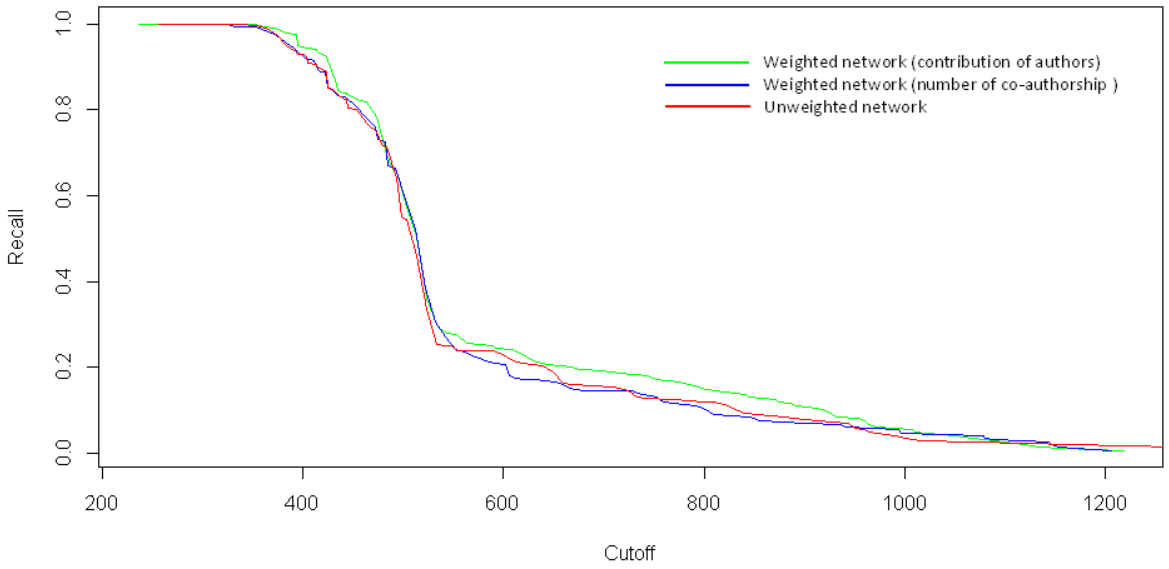

Figure 5.15. Recall rates of the EDCN obtained by the unsupervised strategy.

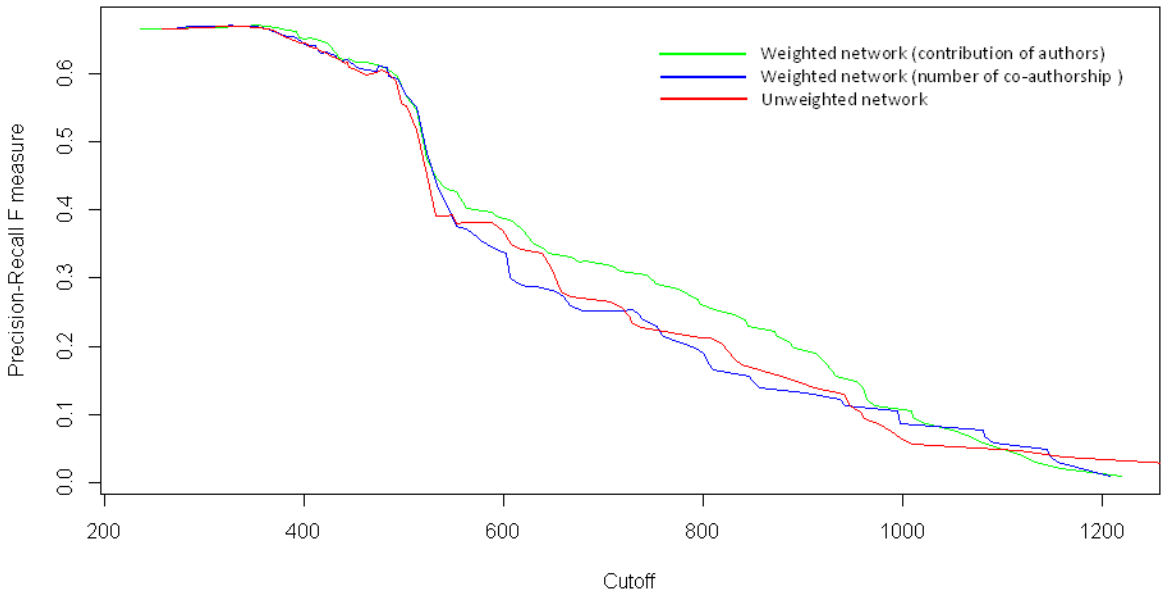

Figure 5.16. F measure rates of the EDCN obtained by the unsupervised strategy.

To summarize results of the unsupervised strategy for both the computer science and the eye disease co-authorsihp networks:

- For every performance measure in the unsupervised strategy, at least one of the weighted networks has better performance than unweighted network at most of the

point for especially Recall and F-measure rates. It shows that even with the unsupervised strategy, weighted networks have better results in comparison to unweighted networks.

- Supervised method has better results than the unsupervised strategy according to F-measure results.

6. CONCLUSION

This study performed a link prediction process with supervised algorithms, which include one fuzzy rule based algorithm in two different kinds of co-authorship networks, which have weighted and unweighted version. Furthermore, the unsupervised strategy also applied in the experiments in order to compare results of supervised algorithms. One of the contributions of this paper was using a fuzzy rule based algorithm to see the improvements in the results and the other one was applying two different co-authorship networks to provide the validity of experiments' results. In this study it was proved that in most cases weighted networks have better performances in both supervised methods and the unsupervised strategy in comparison with unweighted networks. Furthermore, most of the supervised algorithms showed better performance than the unsupervised strategy. Another result was that the decision tree algorithm and the probabilistic algorithm obtained better results than fuzzy rule based algorithm. As a future work, weighted heterogonous networks may be used with different algorithms to provide more improvements in the link prediction task.

REFERENCES

1. Leskovec, J., D.Huttenlocher, and J. Kleinberg, "Predicting positive and negative links in online social networks", *Proceedings of the 19th international conference on World wide web. ACM*, pp. 641–650, 2010.

2. Sun, Y., R. Barber, M. Gupta, C. Aggarwal, and J. Han, "Co-author relationship prediction in heterogeneous bibliographic networks", *in Proc. 2011 Int. Conf. Advances in Social Network Analysis and Mining (ASONAM'11)*, Kaohsiung, Taiwan, July 2011.

3. Liben-Nowell, D. and J. Kleinberg, "The link prediction problem for social networks", *Proceedings of the twelfth international conference on Information and knowledge management*, pp. 556–559, 2003.

4. Gimenes, G., H. Gualdron, T. R. Raddo, and J. F. R. Jr., "Supervised-learning link recommendation in the dblp co-authoring network", *In IEEE PerCom Works. on Social and Community Intelligence*, pages 563-569, 2014.

5. De Sa, H. and R. Prudencio, "Supervised link prediction in weighted networks", *in Joint Conference on Neural Networks*, pp. 2281– 2288, 2011.

6. Fields, C., "How small is the center of science? Short cross-disciplinary cycles in co-authorship graphs", *Scientometrics*, 102:1287–1306, 2014.

7. Huang, S., Y. Tang, F. Tang, and J. Li, "Link prediction based on time-varied weight in co-authorship network", *Proceedings of the 2014 IEEE 18th International Conference on Computer Supported Cooperative Work in Design CSCWD*, pp. 706–709, 2014.

8. Lei, C. and J. Ruan, "A novel link prediction algorithm for reconstructing protein-protein interaction networks by topological similarity", *Bioinformatics*, vol. 29, no. 3, pp. 355–364, 2013.

9. Liu, Z., Q.-M. Zhang, L. Lu, and T. Zhou, “Link prediction in complex networks: A local naive Bayes model”, *EPL*, vol. 96, no.4, Article ID 48007, 2011.

10. Symeonidis, P., N. Iakovidou, N. Mantas, and Y. Manolopoulos, “From biological to social networks: Link prediction based on multi-way spectral clustering”, *Data & Knowledge Engineering*, 87(0):226–242, 2013.

11. Shibata, N., Y. Kajikawa and I. Sakata, “Link prediction in citation networks”, *JASIST*, 2012.

12. Wang, D., D. Pedreschi, C. Song, F. Giannotti, and A.-L. Barabasi, “Human mobility, social ties, and link prediction”, *in SIGKDD. ACM*, pp. 1100–1108, 2011.

13. Bringmann, B., M. Berlingerio, F. Bonchi, and A. Gionis, “Learning and predicting the evolution of social networks”, *IEEE Intelligent Systems*, 25:26–35, 2010.

14. Hasan, M.A., V. Chaoji, S. Salem, and M. Zaki, “Link prediction using supervised learning”, *in In Proc. of SDM 06 workshop on Link Analysis, Counterterrorism and Security*, 2006.

15. Barabasi, A., H. Jeong, Z. Neda, E. Ravasz, A. Schubert, and T. Vicsek, “Evolution of the social network of scientific collaborations”, *Physica A: Statistical Mechanics and its Applications*, vol. 311, no. 3-4, pp. 590–614, 2002.

16. Newman, M., “Clustering and preferential attachment in growing networks”, *Physical Review E*, vol. 64, no. 2, 2001.

17. Newman, M., “The structure of scientific collaboration networks”, *Proceedings of the National Academy of Sciences of the United States of America*, vol. 98, no. 2, p. 404, 2001.

18. Salton, G. and M. McGill, “Introduction to modern information retrieval”, *McGraw-Hill*, New York, NY, 1983.

19. Adamic, L. and E. Adar, “Friends and neighbors on the web”, *Social Networks*, vol. 25, no. 3, pp. 211–230, 2003.

20. Murata, T. and S. Moriyasu, “Link prediction of social networks based on weighted proximity measures”, *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence. IEEE Computer Society*, pp. 85–88, 2007.

21. Bonilla, J. Z., “The nature of co-authorship: A note on recognition sharing and scientific argumentation”, *Synthese*, 191(1), 97–108, 2014.

22. DBLP Official Web Site, “DBLP”, 2017, http://dblp.uni-trier.de/, accessed at May 2017.

23. Zaiane, O., Jiyang Chen, and Randy Goebel, “DBConnect: Mining Research Community on DBLP Data”, *Web Mining and Social Network Analysis Workshop in conjunction with ACM SIGKDD conference*, pp 74-81, San Jose, USA, 2007.

24. Gopubmed Official Web Site, “Gopubmed”, 2017, http://www.gopubmed.com/web/gopubmed/, accessed at May 2017.

25. Yu, Q., C. Long, Y. Lv, H. Shao, P. He, and Z. Duan, “Predicting Co-Author Relationship in Medical Co-Authorship Networks”, *PlOS ONE*, vol. 9, no. 7, 2014.

26. Hall, M., E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The weka data mining software: an update”, *SIGKDD Explorations*, vol. 11, no. 1, pp. 10–18, 2009.

27. Gimenes, G. P., H. Gualdron, J. F. Rodrigues Jr., and M. Gazziro, “Multimodal graph-based analysis over the DBLP repository: critical discoveries and hypotheses”, *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, 1129-1135, 2015.

28. Lichtenwalter, R., J. Lussier and N. Chawla, “New perspectives and methods in link prediction”, *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM*, pp. 243–252, 2010.

29. Lu, L. and T. Zhou, “Link prediction in complex networks: A survey”, *Physica A*, vol. 390, no. 6, pp. 1150–1170, 2011.

30. Huang, Z., X. Li, and H. Chen, “Link prediction approach to collaborative filtering”, *in JCDL, M. Marlino, T. Sumner, and F. M. S. III, Eds. ACM*, pp. 141–142, 2005.

31. Xiang, E. W., “A survey on link prediction models for social network data”, *Ph.D. dissertation, Department of Computer Science and Engineering, The Hong Kong University of Science and Technology*, 2008.

32. Zhou, T., L. Lu, and Y.-C. Zhang, “Predicting missing links via local information”, *The European Physical Journal B - Condensed Matter and Complex Systems*, vol. 71, no. 4, pp. 623–630, 2009.

33. Saramaki, J., M. Kivela, J.-P. Onnela, K. Kaski, and J. Kert'esz, “Generalizations of the clustering coefficient to weighted complex networks”, *Physical Review E*, vol. 75, no. 2, p. 027105, 2007.

34. Witten, I. H., E. Frank, and M. A. Hall, “Data Mining - Practical Machine Learning Tools and Techniques”, *Morgan Kaufmann*, 2011.

35. Qin, Z., “Naive Bayes Classification Given Probability Estimation Trees”, *The 5th International Conference on Machine Learning and Applications (ICMLA’06)*, Orlando, Florida, USA, December 14- 16, 2006.

36. Ruggieri, S., “Efficient c4.5”, *IEEE Trans. Knowledge and Data Eng.*,vol. 14, no. 2, pp. 438-444, Mar./Apr. 2002.

37. Jain, A.K. and J. Mao, “A k-nearest neighbor artificial neural network classifier”, *in Proc. Int. Joint Conf. Neural Networks*, Seattle, WA, pp. 515-520, July 1991.

38. Viswanath, P. and T.H. Sarma, “An improvement to k-nearest neighbor classifier”, *IEEE Recent Advances in Intelligent Computational Systems (RAICS),* pp. 227-231, Sept. 22-24, 2011.

39. Hearst, M., “Support Vector Machines”, *IEEE Intelligent Systems*, vol. 13, no. 4, pp. 18-28, July 1998.

40. Osuna, E., R. Freund, and E Cirosi, “Training support vector machines: an application

to face detection", *Proceedings of IEEE*, pages 130-136, 1997.

41. Chang, C.-C. and C.-J. Lin, "LIBSVM: a library for support vector machines", 2001, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm/", accessed at May 2017.

42. Qin, B., Yuni Xia, Sunil Prabhakar, and Yicheng Tu, "A Rule-Based Classification Algorithm for Uncertain Data", *Proceedings of IEEE International Conference on Data Engineering*, pp.1666-1640, 2009.

43. Hühn, J. and E. Hüllermeier, "FURIA: An algorithm for un-ordered fuzzy rule induction", *Data Min Knowledge Discovery*, vol. 19, no. 3, pp. 293–319, 2009.