Supervised Link Prediction in Weighted Networks

Hially Rodrigues de Sá and Ricardo B. C. Prudêncio

Abstract— **Link prediction is an important task in Social Network Analysis. This problem refers to predicting the emergence of future relationships between nodes in a social network. Our work focuses on a supervised machine learning strategy for link prediction. Here, the target attribute is a class label indicating the existence or absence of a link between a node pair. The predictor attributes are metrics computed from the network structure, describing the given pair. The majority of works for supervised prediction only considers unweighted networks. In this light, our aim is to investigate the relevance of using weights to improve supervised link prediction. Link weights express the 'strength' of relationships and could bring useful information for prediction. However, the relevance of weights for unsupervised strategies of link prediction was not always verified (in some cases, the performance was even harmed). Our preliminary results on supervised prediction on a co-authorship network revealed satisfactory results when weights were considered, which encourage us for further analysis.**

I. Introduction

The advance of Internet provided better experiences for collaboration and interaction between people and organizations. This advance is the basis for the emergence of social networks in the Internet, which are nowadays very popular. A social network can be formally represented as a graph, in which the vertices represent people or organizations, and the connecting edges indicate either social connections or shared characteristics. Social Network Analysis (SNA) is a broad field of research dealing with techniques and strategies for the study of social networks [1]. The analysis and extraction of knowledge of the networks are widely employed when understanding the behavior of a community is a strategic objective [1]. SNA provides opportunities and benefits for different areas such as marketing, economics, health, sociology and safety.

Link prediction is an important task treated by SNA [2]. This task is concerned to the problem of predicting the existence (in the future) of relationships between nodes in a network, based on patterns observed in the existing nodes and relations. Link prediction can help to understand the mechanisms that trigger the evolution in a social network. The literature shows different strategies and approaches to treat this problem [3], [4]. In general, the most popular techniques are divided into three approaches, namely: based on structural measures or patterns in the network; based on the similarity between nodes (content and/or semantics of the nodes); based on probabilistic models; These approaches will be briefly explained in section II.

In our work, we focused on the topological measures based approach and employed a *supervised learning* strategy in which link prediction is treated as a binary classification problem (two nodes establish a relationship or not). A pair of nodes is defined as an instance of the classification problem in which (1) the predictor attributes are metrics computed to describe the pair; and (2) the target attribute indicates the presence (positive label) or absence (negative label) of a relationship between the nodes in the future. Based on a set of such instances, the classification can be performed by different supervised algorithms such as Decision Trees, K-Nearest Neighbors, Neural Networks, among others [2].

Different metrics to describe node pairs were already adopted in previous work, including for instance the number of common neighbors, the path distance between the two nodes, Jaccard's coefficient, Adamic-Adar coefficient, among others [2], [5], [6], [7]. Such metrics explore structural patterns of the network and commonly provide a degree of proximity/similarity between the nodes [5]. The used metrics can be either local (limited to the direct neighbors of nodes) and global (covers the entire network). We highlight here that the most part of the previous work in supervised link prediction consider metrics computed for *unweighted* social networks. In such networks, the *strength* of relationships is not taken into account (only the *existence* of relationships is considered) [2], [6], [4].

Based on the above context, the current paper aims to investigate whether considering weights on links between nodes contributes to improve the performance of supervised link prediction. Subsequently, the unsupervised strategy was applied to make performance comparisons. In this aim, the metrics used as predictor attributes need to be adapted for the case of weighted networks. The existing studies which investigate the impact of link weights were focused on the unsupervised link prediction strategy [7], [5], [8] (possibly the best of these studies). The utility of using weights with this strategy is a controversial issue [8], because in some case studies the prediction performance was significantly harmed when weights in the relationships were considered [8].

Although there are studies investigating the value of weights for unsupervised link prediction, to the best of our knowledge, the current work is the first attempt to investigate this issue specifically for supervised link prediction. In our work, we performed experiments of link prediction on a co-authorship network with and without weights. The reason for using a co-authorship network is that researchers work together to achieve a common goal. Hence, there is a genuine intention or need for the establishment of relations, which can be investigated in detail [7].

The experiments were performed on DBLP (Digital Bi-

Hially Rodrigues de Sá and Ricardo B. C. Prudêncio are with the Center of Informatics, Federal University of Pernambuco, Recife (PE), Brazil (email: {hrs, rbcp}@cin.ufpe.br).

bliography & Library Project) which constitutes a bibliographic dataset that provides a vast amount of information about different research publications in the field of Computer Science. A number of 8 metrics were adopted as predictor attributes (for both unweighted and weighted networks). Different learning algorithms from the WEKA environment [9] were applied in the prediction task. In general, the supervised strategy shows better results than the unsupervised strategy. The experiments demonstrate that the performance of the link prediction in the weighted network is better than the performance of the network without weights, but with a slight difference. Although this result is not positive in absolute terms, it is in contrast to previous work on unsupervised link prediction in which the use of link weights in some cases was not recommended [8].

Section II briefly discusses the link prediction problem. Section III presents social network metrics adopted in our work in the supervised and unsupervised link prediction. Section IV brings the experiments and obtained results. Finally, section V presents some conclusions and future work.

II. Link Prediction

A classic definition of the link prediction problem is expressed by: “Given a snapshot of a social network at time t, we seek to accurately predict the edges that will be added to the network during the interval from time t to a given future time t'” [5]. Among several techniques to treat the problem, the most widespread one (approach) is to explore the topological/structural patterns from the social network of interest [7], [10], [11]. As previously mentioned, the starting point of these techniques is to extract the values/scores of different metrics that represent the closeness between pairs of nodes (see section III). Then, the data obtained are processed to build a model which predicts the hidden links or links that will appear in the future.

The node-wise similarity based approach searches appropriate measures of similarity between two nodes according to the content and/or semantics that they present [2], [3]. Each node on the network can be represented as a vector of features. The more two nodes are similar in terms of their particular attributes, the more they are likely to relate. Cosine coefficient, mutual information and Dice coefficient [3] are examples of techniques in this approach.

The probabilistic models based approach aim to learn the best probabilistic model that abstracts the information of the network. The basic idea is to create the model through a set of parameters θ, given the observed social network G = (V,E) [3]. The existence of the link between a pair of nodes x and y is estimated by the conditional probability $P(e^{<x,y>}|\theta)$ [3]. This approach examines the elements of the network through relational data models, they are able to encapsulate relevant information from nodes, relationships and the graph as a whole. Relational Markov Networks and Relational Bayesian Networks are examples of models in this approach.

Link prediction considering structural patterns can be performed basically by two different strategies. In a first strategy, the pairs of non-connected nodes are initially ranked according to a chosen metric (for instance, the number of common neighbors) [7], [8]. The top L ranked pairs are then assigned as the predicted links. We termed this strategy as an *unsupervised* solution since no labeled training set is adopted to derive a prediction model. The unsupervised strategy presents some limitations. First, what value we should assign to L (i.e., how many top ranked pairs to consider as probable future links) [4]? Another point is that the links are sorted decreasingly according to the score of the chosen metric. In other words, it is always assumed that the links with the highest scores are most likely to occur. However, this is not true depending on the used metric (for instance, preferential attachment) [7]. Finally, the ranking of node pairs is performed using only one metric, and hence, this strategy may not completely explore different structural patterns contained in the network.

Considering the above limitations of unsupervised link prediction, we adopted in our work a supervised machine learning strategy for link prediction (adopted for instance in [4], [2], [11]). This strategy was already introduced in the previous section and will be also described in higher detail in section IV-B. Actually, some authors deployed case studies considering weighted networks, in [5], [12] for example. However, no systematic comparisons were performed to evaluate the real importance of such weights, i.e. whether better results could be achieved with weights compared to the results without considering weights. Our work aims to cover this gap by extending supervised link prediction for weighted networks and by performing comparisons to investigate the influence of using link weights. We expected to contribute to research on link prediction by investigating this specific issue in this context.

III. Proximity Metrics

In this section, we describe the metrics (with and without weights) deployed as predictor attributes in the supervised link prediction. We first provide some definitions and notation which will be useful to understand the descriptions below. Let $\Gamma(x)$ be the set of neighbors of node x in the social network and let $|\Gamma(x)|$ be the degree (number of neighbors) of node x. Let $w(x,y)$ be the link weight between nodes x and y. In our work, we consider undirected graphs and do not consider self-connections. Moreover, $w(x,y) = w(y,x)$.

A. Number of Common Neighbors (CN)

The CN measure for unweighted networks is defined as the number of nodes with direct relationship with both evaluated nodes x and y:

$$CN(x,y) = |\Gamma(x) \cap \Gamma(y)| \tag{1}$$

The CN measure is one the most widespread metrics adopted in link prediction mainly due to its simplicity. Also, it is intuitive since it is expected that a high number of common neighbors make it easier the future contact between two nodes [13]. For weighted networks, the CN measure can be extended as:

$$CN(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} w(x,z) + w(y,z) \tag{2}$$

B. Jaccard's Coefficient (JC)

The JC measure is well explored in Data Mining [14], it assumes higher values for pairs of nodes which share a higher *proportion* of common neighbors relative to total number of neighbors they have. For unweighted networks, the JC measure is defined as:

$$JC(x,y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|} \tag{3}$$

For weighted networks, the JC coefficient can be extended as:

$$JC(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{w(x,z) + w(y,z)}{\sum_{a \in \Gamma(x)} w(a,x) + \sum_{b \in \Gamma(y)} w(b,y)} \tag{4}$$

C. Preferential Attachment (PA)

The PA measure assumes that the probability that a new link is created from a node x is proportional to the node degree $|\Gamma(y)|$ (i.e., nodes that currently have a high number of relationships tend to create more links in the future). Barabasi and Bonabeau [15], and Newman [13] have proposed that the probability of a future link between two nodes can be expressed by the product of their number of collaborators. For unweighted networks the PA measure is given by:

$$PA(x,y) = |\Gamma(x)| * |\Gamma(y)| \tag{5}$$

For weighted networks, the PA measure can be extended as:

$$PA(x,y) = \sum_{a \in \Gamma(x)} w(a,x) * \sum_{b \in \Gamma(x)} w(b,y) \tag{6}$$

D. Adamic-Adar Coefficient (AA)

The AA measure for unweighted networks is defined as:

$$AA(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{log(|\Gamma(z)|)} \tag{7}$$

Adamic and Adar [16] formulated this metric related to the Jaccard's coefficient, it defines a higher importance to the common neighbors which have fewer neighbors. Hence, it measures how exclusive (or strong) is the relationship between a common neighbor and the evaluated pair of nodes. The AA measure is extended for weighted networks as:

$$AA(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{w(x,z) + w(y,z)}{log(1 + \sum_{c \in \Gamma(z)} w(z,c))} \tag{8}$$

E. Path Distance (PD)

The PD for unweighted networks is simply the minimum number of nodes that must be followed from x in order to reach y in the graph [1]. As a special case, when two nodes x and y have a common neighbor then $PD(x,y) = 1$. The lower is the PD measure, the higher is the chance to establish a link. For weighted networks, we identify the minimum path between the pair of evaluated nodes, considering a score $1/w(a,b)$ to the distance between adjacent nodes a and b.

F. Resource Allocation Index (RA)

Resource Allocation Index and Adamic-Adar Coefficient have similar formulas (and both can express the idea of exclusivity between nodes), but they come from different motivations. RA is based on physical processes of resource allocation [17] and can be applied on networks formed by airports (flow of aircrafts and passengers) or networks formed by electric power stations (power distribution), for example. The measure was proposed by Zhou et al. [18] and for unweighted networks is expressed by:

$$RA(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{|\Gamma(z)|} \tag{9}$$

For weighted networks, the RA measure can be defined as:

$$RA(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{w(x,z) + w(y,z)}{\sum_{c \in \Gamma(z)} w(z,c)} \tag{10}$$

G. Local Path (LP)

The LP measure counts all paths of length exactly 2 and 3 between two nonadjacent nodes. Unlike the other metrics that only analyze the interactions of the direct neighbors, this measure has a greater range and can capture more information about the neighborhood of the nodes [18]. Obviously, paths of length 2 are more relevant than paths of length 3, so an adjustment factor e can be applied in the measure. The measure for networks without weights is given by:

$$LP(x,y) = \sum_{l=2}^{\infty} |paths_{x,y}^{<l>}| + e \sum_{l=3}^{\infty} |paths_{x,y}^{<l>}| \tag{11}$$

Considering x and y to the nodes to be evaluated, for weighted networks such measure can be extended as follows: for each path of length exactly 2 is used $w(x,z) + w(y,z)$, where z is a common neighbor of x and y; for each path of length exactly 3 is used $w(x,a) + w(a,b) + w(b,y)$, where

a is a node adjacent to the x and not to y, b is adjacent to the node y and not to x, and a and b are directly connected. So:

$$\begin{aligned} LP(x,y) &= \sum_{l=2}^{\infty} w(x,z) + w(y,z) + \\ &\quad e\sum_{l=3}^{\infty} w(x,a) + w(a,b) + w(b,y) \end{aligned} \tag{12}$$

H. Local Clustering Coefficient (CC)

The CC measure indicates the tendency to form links between neighboring nodes [19], the CC of a node quantifies how close its neighbors are becoming a clique (complete graph), in other words, the local density around the node. The measure is based on counting of triangles, a triangle is formed by node i (analyzed node) that binds the other two nodes m and n, the triangle is considered closed when m and n are directly connected. In an undirected graph, consider t_i as the number of closed triangles attached to node i, the Local Clustering Coefficient around of the node i is calculated by dividing t_i by the largest possible number of distinct triangles linked to i. The weighted and unweighted versions of this coefficient were discussed in [19]. The measure for networks without weights is given by:

$$CC(i) = \frac{2t_i}{|\Gamma(i)| * (|\Gamma(i)| - 1)} \tag{13}$$

The measure to network with weights is defined by:

$$\begin{aligned} CC(i) &= \frac{1}{|\Gamma(i)| * (|\Gamma(i)| - 1)} * \\ &\quad \sum_{m,n \in \Gamma(i)} \frac{w(i,m) + w(i,n)}{2 * \sum_{z \in \Gamma(i)} \frac{w(i,z)}{|\Gamma(i)|}} a_{im} a_{mn} a_{in} \end{aligned} \tag{14}$$

Consider $a_{im}a_{mn}a_{in}$ a closed triangles formed by nodes i, m, n. In this article, we consider the sum of the CC of two nodes x and y as the metric of similarity/proximity to analyze the link prediction. Thus: $CC(x, y) = CC(x) + CC(y)$.

IV. Experiments and Results

In this section, we describe the social network data used in our experiments (section IV-A) and the process adopted to generate a learning dataset from this network (section IV-B). We also present the learning algorithms adopted for the link prediction (section IV-C) and their results, and finally the experiments with the unsupervised strategy (section IV-D).

A. Social Network Data

The social network adopted in our experiments was the co-authorship network from DBLP [1] formed between 1995 and 2004. It consists of 175208 distinct authors and 139882 distinct articles, generating a total of 542681 co-authorships (links). In our work, we performed experiments using three versions of this social network:

- (1) an unweighted version in which a link is presented between a pair of authors if they co-authored at least one paper in the collection;
- (2) a weighted version of the network in which each link between a pair of authors is weighted by the total number of papers co-authored by the two authors;
- (3) a weighted version of co-authorship networks was adopted in [20], [13] in which each link is weighted by the contribution of the authors in their co-authored papers. If a paper has n authors, the specific contribution of a pair of authors for the paper is given by $1/(n-1)$. The link between two authors is then computed as the sum of their specific contributions considering all the co-authored papers. Hence, the link weight indicates how exclusive is the relationship between the authors.

Assigning weights in relationships indicates how ‘strong’ is the link between two authors instead of only considering the binary relation between them (establish a link, or not) [7]. Besides implementing all the metrics in the traditional mode (without weights), variants of these same metrics were implemented using the weight of links in your calculations (as described in section III). The purpose of creating new versions of the metrics is to investigate whether the use of weights in relationships provides better prediction results.

B. Dataset for supervised link prediction

In order to produce a labeled dataset for supervised learning, we adopted the same procedure described in [5] for supervised link prediction. Initially, it is assumed that the evolution of the social network on time is recorded, i.e. we have information available about the time when each node and edge was recorded in the graph. We then considered the state of the network on two different time periods t and t' (with $t < t'$). We used information of the social network up to time t to predict new links which will be formed up to t'.

Let $G(V, E)$ be the social network of interest. Let $G[t]$ be the sub-graph of G containing the nodes and edges recorded up to time t. In turn, let $G[t']$ be the sub-graph of G observed up to time t' ($t' > t$). In order to generate training examples, we considered the pairs of nodes in $G[t]$ that do not have an edge yet (i.e. no relationships were established up to time t). Each non-connected pair of nodes corresponded to a training instance storing:

- (1) the predictor attributes: features describing the pair of nodes. We adopted the measures described in section III;

[1] Available to download in http://dblp.uni-trier.de/xml/dblp.xml

- (2) the class label (*positive* or *negative*): here, a pair of nodes was labeled as positive if an edge connecting the nodes was now observed in $G[t']$ and labeled as negative otherwise.

In our experiments on the co-authorship network, the first sub-graph ($G[t]$) corresponded to the information available up to 1999. The second sub-graph ($G[t']$) in turn was defined up to 2004. Hence, our link prediction task consisted of predicting new links appearing in the period from 2000 to 2004 based on the graph information extracted from 1995 to 1999. From sub-graph ($G[t]$), a number of 800 non-connected pairs of authors was randomly identified and the corresponding features were extracted, thus generating 800 examples. From these instances, 400 examples were assigned as positive, and 400 pairs in turn were assigned as negative (the classes are equally distributed).

Three different datasets were created with the all the identified pairs of authors, each dataset was generated using one of three cited versions of the network (section IV-A). The default Accuracy of classification is 50.00%, which reflects the percentage of the majority class. The default Accuracy represents the performance achieved by a simple algorithm that classifies all instances according to the majority class. Although it is not very useful in practice, it can be used as a base for comparison of other algorithms.

C. Experiments with supervised strategy

In our experiments, different learning algorithms were used to link prediction in the datasets described in the previous section. All algorithms were available in the WEKA environment [9]:

- (1) NaivesBayes (NB) - Implementation of the Naive Bayes classifier [21];
- (2) J48 - Implementation of the C4.5 algorithm (Decision Trees) [22];
- (3) IBk - Implementation of the k-nearest neighbors algorithm [23];
- (4) LibSVM - A widespread library for support vector machines [24];
- (5) LibLinear - Linear implementation of support vector machines [25].

In our experiments, all experiments were applied with the default parameters of WEKA. Particularly, the LibSVM is defined with RBF kernel and cost parameter equal to 1. The algorithms were executed in the module ‘Experimenter’ of WEKA on the three datasets described in sections IV-A and IV-B. In all experiments, the algorithms were evaluated with stratified 10-fold cross-validation. For more reliable results, the cross-validation procedure was executed 10 times for each algorithm and dataset. A t-test was performed with a significance level of 95% in order to statistically compare the algorithms. Tables I, II and III present the performance measures obtained for each dataset and algorithm:

First of all, the performance obtained by the evaluated algorithms was compared to the default Accuracy (percentage of the majority class - 50.00%) in order to verify the viability of the supervised link prediction task. Table I shows that the Accuracy obtained by all algorithms were higher than the default Accuracy, indicating that useful knowledge can in fact be acquired from the available datasets. The difference in performance was verified by the t-test (at 95% of confidence) for all algorithms and datasets employed. As can be seen in table I and table II, WEKA statistically ranked LibLinear with the highest Accuracy, followed by J48. On the other hand, WEKA presents J48 as the most statistically precise, but LibLinear is very close. Regarding Recall, NB got the best rates among all algorithms, independent of the analyzed network.

TABLE I

ACCURACY OBTAINED BY THE ALGORITHMS ON EACH DATASET

Algorithm/Dataset	Unweighted network	Weighted network [1]	Weighted network [2]
J48	70.99%	70.30%	68.13%
LibLinear	69.13%	71.13%	70.28%
NB	66.15%	67.83%	65.64%
IBk	65.03%	67.13%	66.88%
LibSVM	65.94%	69.72%	67.50%

[1] number of co-authorships
[2] contribution of authors

TABLE II

PRECISION (P) AND RECALL (R) RATE OBTAINED BY THE ALGORITHMS ON EACH DATASET

Algorithm/Dataset	Unweighted network	Weighted network [1]	Weighted network [2]
J48	P:69% R:76%	P:70% R:71%	P:65% R:80%
LibLinear	P:67% R:77%	P:68% R:81%	P:67% R:82%
NB	P:62% R:83%	P:63% R:89%	P:61% R:88%
IBk	P:65% R:64%	P:68% R:66%	P:67% R:68%
LibSVM	P:62% R:82%	P:65% R:85%	P:63% R:87%

[1] number of co-authorships
[2] contribution of authors

There is a known trade-off between Precision and Recall in the field of Information Retrieval [26]. Some studies in the literature have shown that there is a tendency that one measure increases as the other one decreases [26]. The traditional F-measure (or F-score), which is the harmonic mean of Precision and Recall, can balance the two measures. According to the table III (and statistically tested by WEKA), the LibLinear generally has the best rates for F-measure.

The Area Under the ROC Curve (AUC) is an important performance measure that relates the sensitivity (true positive rate) and specificity (true negative rate) of a classifier [27]. All algorithms obtained the AUC higher than the majority classifier (with AUC = 0.50). By the statistical ranking generated by WEKA, the algorithms J48 and NB obtained the best results for the AUC.

The above analysis was performed to verify the utility of the available data for link prediction as well as to indicate the best algorithms according to different evaluation measures.

TABLE III

AREA UNDER ROC CURVE (A) AND F-MEASURE (F) OBTAINED BY THE ALGORITHMS ON EACH DATASET

Algorithm/Dataset	Unweighted network	Weighted network [1]	Weighted network [2]
J48	A:0.73 F:0.72	A:0.73 F:0.70	A:0.69 F:0.71
LibLinear	A:0.69 F:0.71	A:0.71 F:0.74	A:0.70 F:0.74
NB	A:0.70 F:0.71	A:0.72 F:0.73	A:0.72 F:0.72
IBk	A:0.65 F:0.65	A:0.68 F:0.67	A:0.67 F:0.67
LibSVM	A:0.66 F:0.71	A:0.70 F:0.74	A:0.67 F:0.73

[1] number of co-authorships

[2] contribution of authors

The main issue of our work however is to investigate whether to deploy link weights can be useful in the prediction task. It is possible to observe in tables I, II and III that each classifier had similar performances on the three networks. However we can point out that in most cases, better results were achieved for the networks with weights (slightly better). Also, the better performance on the weighted networks was more consistent considering different algorithms. In almost all comparisons between the networks, the unweighted network had a lower performance compared to at least one of the weighted networks. Although these results are not conclusive, they indicate that improvements in performance (even a little) can be achieved by considering link weights on the link prediction task.

As an unexpected result, we noticed that the results observed on weighted network 1 (i.e. considering the number of co-authored papers) were better than the results achieved on the weighted network 2 (i.e. considering contribution of authors in their co-authored papers), especially in the Accuracy. This result was not expected since the weights in network 2 potentially bring more information than network 1. Aiming to find an explanation, we evaluate the worth of each attribute by computing the value of the Chi-Squared statistic with respect to the class. Evaluating the two weighted networks, it was observed that the contributions of an predictor attributes across the two networks were very similar. The exception was the attribute $Path\ Distance$. The merit of this attribute on the network 1 (108.323 points) was much higher than its merit in the network 2 (33.555 points). Based on this, we can suppose that the smaller merit of the $Path\ Distance$ attribute on the network 2 contributed to harm the classification performance on this network. More experiments however should be performed to consolidate this assumption.

It is important to discuss our results in the light of previous work on unsupervised link prediction which investigated the influence of link weights. As said in section I, the results of unsupervised link prediction can be significantly harmed by adopting link weights. Although the use of weights in our experiments usually led to a consistent improvement in performance, it is important to mention that we did not observe any significant decrease in performance (as it was observed in [8] for unsupervised link prediction). This way, the improvement in performance achieved in our experiments for some algorithms encourage us to perform more analysis on supervised link prediction for weighted networks.

D. Experiments with the unsupervised strategy

Aiming to expand our analysis, we also applied the unsupervised link prediction and perform a comparison to the supervised strategy. In the traditional application of unsupervised link prediction, one isolated metric is chosen to rank the pairs of non-connected nodes. The top ranked examples are then considered as the future links that are more likely to appear. In our work we could adopt one of the 8 different proximity metrics described in section III to generate such ranking. In order to avoid choosing only one metric, we alternatively combined the 8 metrics by adopting the following procedure. First, for each metric we generated a ranking of the available examples and recorded the correspond rank value of each instance. Following, we computed the average rank of each instance across the metrics. Finally, a resulting ranking of examples is then generated directly from their average rank values.

In this section, we adopted the Precision, Recall and F-Measure to evaluate the unsupervised link prediction as usually deployed in previous work. In our experiments, we generated the curves of Precision and Recall of positive examples by evaluating the rank of examples from the top to the bottom of the list. The curves of Precision, Recall and F-Measure are shown in figures 1, 2 and 3.

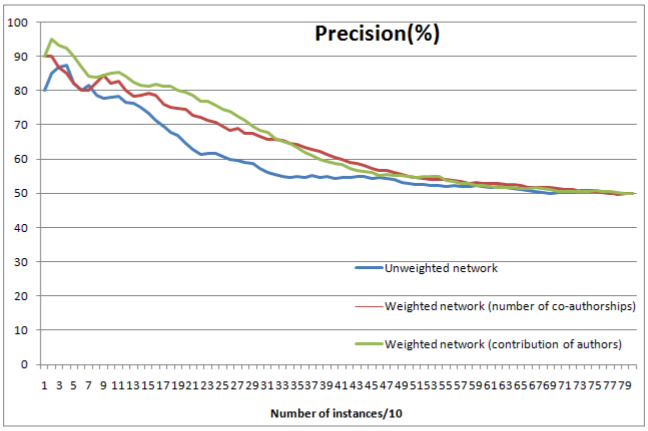

Fig. 1. Precision obtained on each dataset

By observing the three curves, it is possible to observe that the performance measures of the networks converged in accordance with the processed instances. It occurs independent of the network since the final Precision and Recall are respectively 50% (400 positive instances among 800 instances) and 100% (among the 800 instances, all positive instances are present). Moreover, as the F-measure is derived from Precision and Recall, the final F-measure is around 66.66%. The differences in performance across the networks are more

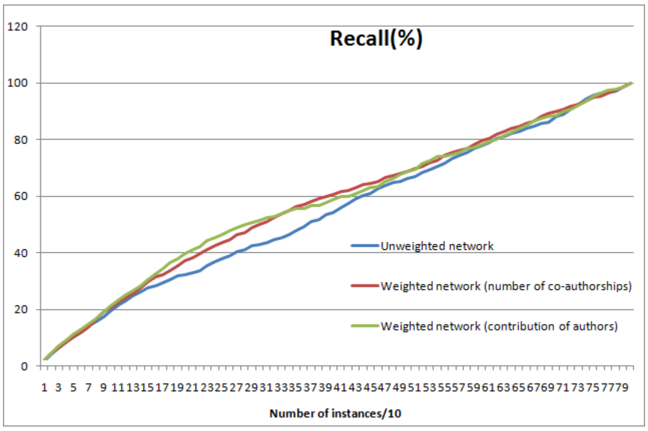

Fig. 2. Recall obtained on the each dataset

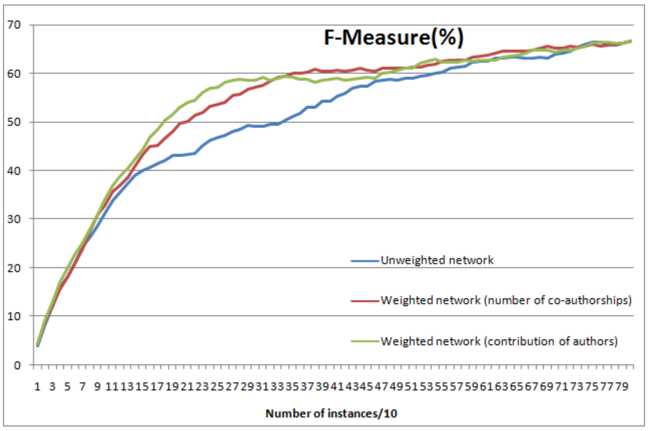

Fig. 3. F-Measure obtained on each dataset

visible in the first halves of the curves. The weighted network by the contribution of the authors provided the best results up to the first 340 instances. In turn, the weighted network by the number of co-authorships becomes the best one later. The best results were observed for the weighted networks in almost all points in the curves. The justification for these results is that the types of weights more accentuated the values of the metrics of positive instances (the differences between values with and without weights were higher). This way, for the weighted networks, the majority of positive instances concentrated on the top. These results indicate the using weights can also bring benefits for the unsupervised link prediction.

By comparing the F-measure observed for the supervised link prediction (see table III) with the curve of F-measure for the unsupervised strategy, we observed that the unsupervised strategy does not exceed the results of the supervised strategy (which was typically greater than 70%). The only exception is IBk performance on the network without weights which was 65%. This comparison is important since it provided additional evidence that the supervised strategy offers more advantage compared to the unsupervised strategy.

V. Conclusion

This paper has focused on the analysis of supervised learning link prediction for weighted networks. As a contribution, this article analyzes and compares the supervised link prediction in a co-authorship network with and without weights and also confronts the results with the unsupervised link prediction, unlike previous works that focused only on the unsupervised strategy [8], [7]. The experiments on supervised link prediction revealed satisfactory results when link weights were considered, and generally speaking the supervised strategy shows better performance than the unsupervised strategy.

This work is still limited regarding the number of the case studies considered. In fact, our experiments were restricted to a co-authorship network. Although co-authorship data have been used by many authors in literature, other social networks with different dynamics and structures can be considered in the future. In order to better investigate the possible benefits of using weights on the links, future work will focus on analyzing networks in different contexts.

References

[1] S. Wasserman and K. Faust, *Social Network Analysis. Methods and Applications.* Cambridge University Pres, 1994.

[2] M. A. Hasan, 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.

[3] E. W. Xiang, "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.

[4] C. Wang, V. Satuluri, and S. Parthasarathy, "Local probabilistic models for link prediction," *Seventh IEEE International Conference on Data Mining ICDM 2007*, vol. 173, no. 14, pp. 322–331, 2007.

[5] D. Liben-Nowell and J. Kleinberg, "The link prediction problem for social networks," in *Proceedings of the 12th international conference on information and knowledge management*, 2003, pp. 556–559.

[6] Z. Huang, X. Li, and H. Chen, "Link prediction approach to collaborative filtering," in *JCDL '05: Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries*, 2005, pp. 141–142.

[7] T. Murata and S. Moriyasu, "Link prediction based on structural properties of online social networks," *New Generation Computing*, vol. 26, no. 3, pp. 245–257, 2008.

[8] L. Lü and T. Zhou, "Role of weak ties in link prediction of complex networks," in *CNIKM '09: Proceeding of the 1st ACM international workshop on Complex networks meet information & knowledge management*, 2009, pp. 55–58.

[9] M. Hall, 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.

[10] Z. Huan, "Link prediction based on graph topology: The predictive value of the generalized clustering coefficient," in *Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (LinkKDD2006)*, 2006.

[11] A. Potgieter, K. April, R. Cooke, and I. Osunmakinde, "Temporality in link prediction: Understanding social complexity," *Sprouts: Working Papers on Information Systems*, vol. 7, no. 9, 2007.

[12] M. Pavlov and R. Ichise, "Finding experts by link prediction in co-authorship networks," in *Proceedings of the Workshop on Finding Experts on the Web with Semantics (FEWS2007) at ISWC/ASWC2007, Busan, South Korea*, 2007.

[13] M. E. J. Newman, "Clustering and preferential attachment in growing networks," *Physical Review Letters E*, vol. 64, 2001.

[14] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*. Addison Wesley, 2005.

[15] A. L. Barabasi and E. Bonabeau, "Scale-free networks," *Scientific American*, vol. 288, no. 5, pp. 60–69, 2003.

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

[17] Q. Ou, Y.-D. Jin, T. Zhou, B.-H. Wang, and B.-Q. Yin, "Power-law strength-degree correlation from resource-allocation dynamics on weighted networks," *Phys Rev E Stat Nonlin Soft Matter Phys*, vol. 75, no. 2 Pt 1, p. 021102, 2007.

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

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

[20] X. Liu, J. Bollen, M. L. Nelson, and H. V. de Sompel, "Coauthorship networks in the digital library research community," *Information Processing and Management*, vol. 41, no. 6, pp. 1462–1480, 2005.

[21] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2000.

[22] J. R. Quinlan, *C4.5: Programs for Machine Learning (Morgan Kaufmann Series in Machine Learning)*. Morgan Kaufmann, 1993.

[23] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," *Machine Learning*, vol. 6, no. 1, pp. 37–66, 1991.

[24] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[25] R. E. Fan, K. W. Chang, C. J. Hsieh, X. R. Wang, and C. J. Lin, "Liblinear: A library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, 2008.

[26] M. Buckland and F. Gey, "The relationship between recall and precision," *J. Am. Soc. Inf. Sci.*, vol. 45, no. 1, pp. 12–19, 1994.

[27] T. Fawcett, "An introduction to roc analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.