Link Prediction using Supervised Learning *

Mohammad Al Hasan, Vineet Chaoji, Saeed Salem, and Mohammed Zaki
Rensselaer Polytechnic Institute, Troy, New York 12180
{alhasan, chaojv, salems, zaki}@cs.rpi.edu

Abstract

Social network analysis has attracted much attention in recent years. Link prediction is a key research direction within this area. In this paper, we study link prediction as a supervised learning task. Along the way, we identify a set of features that are key to the performance under the supervised learning setup. The identified features are very easy to compute, and at the same time surprisingly effective in solving the link prediction problem. We also explain the effectiveness of the features from their class density distribution. Then we compare different classes of supervised learning algorithms in terms of their prediction performance using various performance metrics, such as accuracy, precision-recall, F-values, squared error etc. with a 5-fold cross validation. Our results on two practical social network datasets shows that most of the well-known classification algorithms (decision tree, k-NN, multilayer perceptron, SVM, RBF network) can predict links with comparable performances, but SVM outperforms all of them with narrow margin in all performance measures. Again, ranking of features with popular feature ranking algorithms shows that a small subset of features always plays a significant role in link prediction.

1 Introduction and Background

Social networks are a popular way to model the interaction among the people in a group or community. They can be visualized as graphs, where a vertex corresponds to a person in some group and an edge represents some form of association between the corresponding persons. The associations are usually driven by mutual interests that are intrinsic to a group. However, social networks are very dynamic objects, since new edges and vertices are added to the graph over the time. Understanding the dynamics that drives the evolution of social network is a complex problem due to a large number of variable parameters. But, a comparatively easier problem is to understand the association between two specific nodes. Several variations of the above problems make interesting research topics. For instance, some of the interesting questions that can be posed are – how does the association pattern change over time, what are the factors that drive the associations, how is the association between two nodes affected by other nodes. The specific problem instance that we address in this research is to predict the likelihood of a future association between two nodes, knowing that there is no association between the nodes in the current state of the graph. This problem is commonly known as the *Link Prediction* problem.

We use the coauthorship graph from scientific publication data for our experiments. We prepare datasets from the coauthorship graphs, where each data point corresponds to a pair of authors, who never coauthored in training years. Depending on the fact whether they coauthored in the testing year or not, the data point has either a positive label or a negative label. We apply different types of supervised learning algorithms to build binary classifier models that distinguish the set of authors who will coauthor in the testing year from the rest who will not coauthor.

Predicting prospective links in coauthorship graph is an important research direction, since it is identical, both conceptually and structurally to many practical social network problems. The primary reason is that a coauthorship network is a true example of social network, where the scientists in the community collaborate to achieve a mutual goal. Researchers [20] have shown that this graph also obeys the power-law distribution, an important property of a typical social network. To name some practical problems that very closely match with the one we study in this research, we consider the task of analyzing and monitoring terrorist networks. The objective in analyzing terrorist networks is to conjecture that particular individuals are working together even though their interactions cannot be identified from the current information base. Intuitively, we are predicting hidden links in a social network formed by the group of terrorists. In general, link prediction provides a measure of social proximity between two vertices in a

*This material is based upon work funded in whole or in part by the US Government and any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the US Government. This work was supported in part by NSF CAREER Award IIS-0092978, DOE Career Award DE-FG02-02ER25538, and NSF grants EIA-0103708 and EMT-0432098.

social group, which, if known, can be used to optimize an objective function over the entire group, especially in the domain of collaborative filtering [22], Knowledge Management Systems [8], etc. It can also help in modeling the way a disease, a rumor, a fashion or a joke, or an Internet virus propagates via a social network [13].

Our research has the following contributions:

1. We explain the procedural aspect of constructing a machine learning dataset to perform link prediction.

2. We identify a short list of features for link prediction in a particular domain, specifically, in the coauthorship domain. These features are powerful enough to provide remarkable accuracy and general enough to be applicable in other social network domains. They are also very inexpensive to obtain.

3. We experiment with a set of learning algorithms to evaluate their performance in link prediction problem and perform a comparative analysis among them.

4. We evaluate each feature; first visually, by comparing their class density distribution and then algorithmically through some well known feature ranking algorithms.

2 Related Work

Although most of the early research in social network has been done by social scientists and psychologists [19], numerous efforts have been made by computer scientists recently. Most of the work has concentrated on analyzing the social network graphs [2, 9]. Few efforts have been made to solve the link prediction problem, specially for social network domain. The closest match with our work is that of D. Liben, et al. [17], where the authors extracted several features from the network topology of a coauthorship network. Their experiments evaluated the effectiveness of these features for the link prediction problem. The effectiveness was judged by the factor by which the prediction accuracy was improved over a random predictor. This work provides an excellent starting point for link prediction as the features they extracted can be used in a supervised learning framework to perform link prediction in a more systematic manner. But, they used features based on network topology only. We, on the other hand, added several non-topological features and found that they improve the accuracy of link prediction substantially. In practice, such non-topological data are available (for example, overlap of interest between two persons) and they should be exploited to achieve a substantial improvement in the results. Moreover, we compare different machine learning algorithms for the link prediction task.

Another recent work by Faloutsos et al. [10], although does not directly perform link prediction, is worth mentioning in this context. They introduced an object, called *connection subgraph*, which is defined as a small subgraph that best captures the relationship between two nodes in a social network. They also proposed efficient algorithm based on electrical circuit laws to find the connection subgraph from large social network efficiently. Connection subgraph can be used to effectively compute several topological feature values for supervised link prediction problem, especially when the network is very large.

There are many other interesting recent efforts [11, 3, 5] related to social network, but none of these were targeted explicitly to solve the link prediction problem. Nevertheless, experiences and ideas from these papers were helpful in many aspects of this work. Goldenberg et al. [11] used Bayesian Networks to analyze the social network graphs. Baumes et al. [3] used graph clustering approach to identify sub-communities in a social network. Cai et al. [5] used the concept of relation network, to project a social network graph into several relation graphs and mine those graphs to effectively answer user's queries. In their model, they extensively used optimization algorithms to find the most optimal combination of existing relations that best match the user's query.

3 Data and Experimental Setup

Consider a social network $G = \langle V, E \rangle$ in which each edge $e = \langle u, v \rangle \in E$ represents an interaction between u and v at a particular time t. In our experimental domain the interaction is defined as coauthoring a research article. Each article bears, at least, author information and publication year. To predict a link, we partition the range of publication years into two non-overlapping sub-ranges. The first sub-range is selected as training years and the later one as the testing years. Then, we prepare the classification dataset, by choosing those author pairs, that appeared in the training years, but did not publish any papers together in those years. Each such pair either represents a positive example or a negative example, depending on whether those author pairs published at least one paper in the testing years or not. Coauthoring a paper in testing years by a pair of authors, establishes a link between them, which was not there in the training years. Classification model of link prediction problem needs to predict this link by successfully distinguishing the positive classes from the dataset. Thus, link prediction problem can be posed as a binary classification problem, that can be solved

by employing effective features in a supervised learning framework.

In this research, we use two bibliographic datasets: Elsevier BIOBASE (`http://www.elsevier.com`) and DBLP (`http://dblp.uni-trier.de/xml/`), that have information about different research publications in the field of biology and computer science, respectively. For BIOBASE, we used 5 years of data from 1998 to 2002, where the first 4 years are used as training and the last as testing. For DBLP, we used 15 years of data, from 1990 to 2004. First 11 years were used as training and the last 4 years as testing. Pairs of authors that represent positive class or negative class were chosen randomly from the list of pairs that qualify. Then we constructed the feature vector for each pair of authors. A detailed description of the features is given in the following sub-section. The datasets have been summarized in table 1.

Dataset	Number of papers	Number of authors
BIOBASE	831478	156561
DBLP	540459	1564617

Table 1: Statistics of Datasets

3.1 Feature Set Choosing an appropriate feature set is the most critical part of any machine learning algorithm. For link prediction, we should choose features that represent some form of proximity between the pair of vertices that represent a data point. However, the definition of such features may vary from domain to domain for link prediction. In this research, we name these as *proximity features*. For example, for the case of coauthorship network, two authors are close (in the sense of a social network) to each other, if their research work evolves around a larger set of identical keywords. A similar analogy can be given for a terrorist network, wherein, two suspects can be close, if they are experts in an identical set of dangerous skills. In this research, although we restrict our discussion to the feature set for coauthorship link analysis, the above generic definition of proximity measure provides a clear direction to choose conceptually identical features in other network domains. One favorable property of these features is that they are very cheap to compute.

Beside the proximity measure, there exist individual attributes that can also provide helpful clues for link prediction. Since, these attributes only pertain to one node in the social network, some aggregation functions need to be used to combine the attribute values of the corresponding nodes in a node-pair. We name these as *aggregated features*. To illustrate further, let's consider the following example. We choose two arbitrary scientists x and y from the social network. The probability that x and y coauthor is, say $p1$. Then, we choose one scientist z, from the same network, who works mostly on multi-disciplinary research, thus has established a rich set of connections in the community. Now, if $p2$ is the probability that x will coauthor with z, the value of $p2$ is always higher than $p1$, with the available information that z is a prolific researcher. We summarize the idea with this statement: if either (or both) of the scientists are prolific, it is more likely that they will collaborate. Before aggregation, the individual measure is how prolific a particular scientist is and the corresponding individual feature is the number of different areas (s)he has worked on. Summing the value to combine these, yields an aggregated feature that is meaningful for the pair of authors for link prediction. In this example, the higher the attribute value, the more likely that they will collaborate. A similar individual feature, in a terrorist network, can be the number of languages a suspect can speak. Again, aggregating the value produces an aggregated feature for link prediction in a terrorist network.

Finally, we like to discuss about the most important set of features that arise from the network topology. Most importantly, they are applicable equally to all domains since their values depends only on the structure of the network. Here, we name these as *topological features*. Several recent initiatives [17, 14, 15] have studied network topological features for different application areas, like link analysis, collaborative filtering, etc. However, for link prediction the most obvious among these feature is the shortest distance among the pair of nodes being considered. The shorter the distance, the better the chance that they will collaborate. There are other similar measures, like number of common neighbors, Jaccard's coefficient, edge disjoint k shortest distances, etc. For a more detailed list, see [17].

There are some features, that could be a part of more than one category. For example, we can aggregate a topological feature that corresponds to a single network node. However, in our discussion, we place them under the category that we consider to be most appropriate.

Next we provide a short description of all the features that we used for link prediction in a coauthorship network. We also describe our intuitive argument on choosing them as a feature for link prediction problem. Note that, not all the features were applied to both the datasets, due to the unavailability of information.

3.1.1 Proximity Features In the BIOBASE database, we only had one such feature. Since keyword data was not available in DBLP dataset, we could not

use this feature there.

• **Keyword Match Count** This feature directly measures the proximity of a pair of nodes (authors). Here we list all the keywords that the individual authors had introduced in his papers and take a intersection of both the sets. The larger the size of the intersection, the more likely they are to work in related areas and hence a better candidate to be a future coauthor pair.

3.1.2 Aggregated Features As we described earlier, these features are usually related to a single node. We used the simplest aggregation function, namely, *sum* to convert the feature to a meaningful candidate for link prediction. A more complex aggregation function can be introduced if it seems appropriate.

• **Sum of Papers** The value of this feature is calculated by adding the number of papers that the pair of authors published in the training years. Since, all authors did not appear in all the training years, we normalized the paper count of each author by the years they appeared in. The choice of this feature comes from the fact that authors having higher paper count are more prolific. If either (or both) of the authors is (are) prolific, the probability is higher that this pair will coauthor compared to the probability for the case of any random pair of authors.

• **Sum of Neighbors** This feature represents the social connectivity of the pair of authors, by adding the number of neighbors they have. Here, neighborhood is obtained from the coauthorship information. Several variants of this feature exist. A more accurate measure would consider the weighted sum of neighbors, where the weights represent the number of publication that a node has with that specific neighbor. We considered all the weights to be 1. This feature is intended to embed the fact that a more connected person is more likely to establish new coauthor links. Note that, this feature can also be placed under topological features, where the number of neighbors can be found by the degree of a node.

• **Sum of Keyword Counts** In scientific publication, keywords play a vital role in representing the specific domain of work of researchers. Researchers that have a wide range of interests or those who work on interdisciplinary research usually use more keywords. In this sense they have better chance to collaborate with new researchers. Here, also we used the sum function to aggregate this attribute for both the author pair.

• **Sum of Classification Code** Usually, research publication are categorized in code strings to organize related areas. Similar to keyword count, a publication that has multiple codes is more likely to be an interdisciplinary work, and researchers in these area usually have more collaborators.

• **Sum of** log**(Secondary Neighbors Count)** While number of primary neighbors is significant, the number of secondary neighbors sometimes play an important role, especially in a scientific research collaboration. If an author is directly connected to another author who is highly connected (consider a new graduate student with a very well-known adviser), the former person has a better chance to coauthor with a distant node through the later person. Since, the number of secondary neighbors in social network usually grow exponentially, we take the logarithm of the secondary neighbor count of the pair of authors before we sum the individual node values. This attribute can also be placed under topological feature as it can be computed only from the network topology. Calculation of this feature is somewhat costly.

3.1.3 Topological Features We used the following three features in our research, but there are other features that can be useful as well.

• **Shortest Distance** This feature is one of the most significant in link prediction as we found in our research. Kleinberg [16, 20] discovered that in social network most of the nodes are connected with a very short distance. This remarkable characteristic makes it a very good feature for link prediction. We used smallest hop count as the shortest distance between two nodes. There are several variants of this feature. Instead of computing one shortest distance, we can compute k edge-disjoint shortest distance. Each of these can be one feature. Importance of the feature gradually decreases as k increases. Moreover, a shortest distance can be weighted, where each edge has an actual weight instead of a value 1 as it is for unweighted shortest distance. For any pair of nodes, the weight on the edge can be chosen to be the reciprocal of the number of papers the corresponding author pair has coauthored. However, each of these variants are more costly to compute.

• **Clustering Index** Many initiatives within social network research [17, 20] have indicated clustering index as an important features of a node in a social network. It is reported that a node that in dense locally is more likely to grow more edges compared to one that is located in a more sparse neighborhood. The clustering index measures the localized density. Newman [20] defines clustering index as the fraction of pairs of a person's collaborators who have also collaborated with one another. Mathematically, If u is a node of a graph, The clustering index of u is: $\frac{3 \times number\ of\ triangles\ with\ u\ as\ one\ node}{number\ of\ connected\ triples\ with\ u\ as\ one\ node}$

• **Shortest Distance in Author-KW graph** We considered this as a topological attribute, although it requires an extended social network to compute it. To compute this attribute we extended the social network by adding Keyword(KW) nodes. Each KW node is connected to an author node by an edge if that keyword is used by the author in any of his papers. Moreover, two keywords that appear together in any paper are also connected by an edge. A shortest distance between two nodes in this extended graph is computed to get this attribute value.

In addition to these features, we also tried several other features, like Jaccard's coefficient, Adamic/Adar [1], etc., mostly related to network topology. Unfortunately, they did not provide any significant improvement on the classifier performance.

We normalize the feature values to have zero mean and one standard deviation before using them in the classification model.

3.2 Classification Algorithms There exist a plethora of classification algorithms for supervised learning. Although their performances are comparable, some usually work better than others for a specific dataset or domain. In this research, we experimented with seven different classification algorithms. For some of these, we tried more than one variation and reported the result that showed the best performance. The algorithms that we used are SVM (two different kernels), Decision Tree, Multilayer Perceptron, K-Nearest Neighbors (different variations of distance measure), Naive Bayes, RBF Network and Bagging. For SVM, we used the SVM-Light implementation (`http://svmlight.joachims.org`). For K-Nearest neighbors, we programmed the algorithm using Matlab. For the rest of the algorithms, a well known machine learning library, WEKA [24] was used.

Then we compared the performance of the above classifiers using different performance metrics like accuracy, precision-recall, F-value, squared-error etc. For all the algorithms, we used 5-fold cross validation for the results reported. For algorithms that have tunable parameters, like SVM, K-Nearest Neighbors, etc., we used a separate validation set to find the optimum parameter values. In SVM the trade-off between training error and margin of 8 was found to be optimum. For k-nearest neighbor, a value of 12 for k yielded the best performance for BIOBASE dataset and a value of 32 for the DBLP dataset. For others, default parameter values of WEKA worked quite well. However, for most of the models the classifier performance was found not to be very sensitive with respect to model parameter values unless they were quite off from the optimal setting.

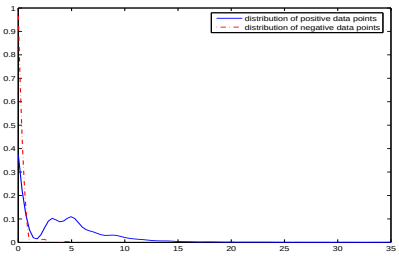
4 Results and Discussions

Table 2 and 3 show the performance comparison for different classifiers on the BIOBASE and DBLP datasets respectively. In both the datasets, counts of positive class and the negative class were almost the same. So, a baseline classifier would have an accuracy around 50% by classifying all the testing data points to be equal to 1 or 0, whereas all the models that we tried reached an accuracy above 80%. This indicates that the features that we had selected have good discriminating ability. For BIOBASE dataset we used 9 features and for the DBLP dataset we used only 4 features. There was not enough information available with the DBLP dataset. Name of the feature used, for each of the dataset are available from table 4 and 5.

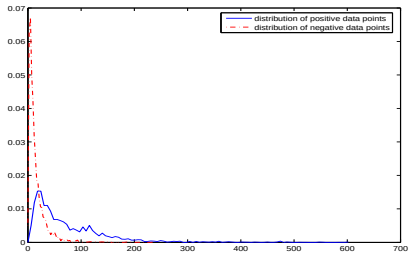
On accuracy metrics, SVM with **RBF** kernel performed the best for both the datasets with an accuracy of 90.56% and 83.18%, respectively. Naturally, the performance on DBLP dataset is worse compared to BIOBASE as fewer features were used in the former dataset. Moreover, DBLP dataset was obtained using 15 years of published articles and the accuracy of link prediction deteriorates over the longer range of time span since the institution affiliations, coauthors and research areas of researchers may vary over time. So, predicting links in this dataset is comparably more difficult than the BIOBASE dataset, where we used only 5 years of data. In both the datasets, other popular classifiers, like decision tree, k-nearest neighbors and multilayer perceptron also have similar performances, usually 0.5% to 1% less accurate than SVM. Such a small difference is not statistically significant, so no conclusion can be drawn from the accuracy metric about the most suited algorithm for the link prediction.

To further analyze the performance, we also applied the most popular ensemble classification techniques, *bagging* for link prediction. Bagging groups the decisions from a number of classifiers, hence the resulting model is no more susceptible to variance errors. Performance improvement of bagging, over the independent classifiers are high when the overlap of the misclassification sets of the independent classifiers is small [7]. The bagging accuracy for the datasets is 90.87 and 82.13, which indicates almost no improvements. This implies that majority of misclassifications are from the bias error introduced by inconsistent feature values in those samples. Hence, most of the classifiers failed on these samples.

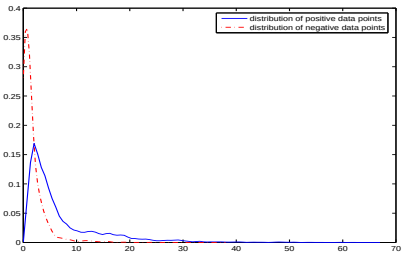
To understand the inconsistency in feature values, we investigate the distribution of positively and negatively labeled samples for four important features in each dataset as shown in figure 1 and 2. The distribution of feature values are plotted along the y-axis for



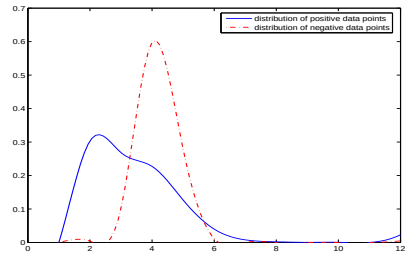
(a) Keyword Match



(b) Sum of Neighbors count

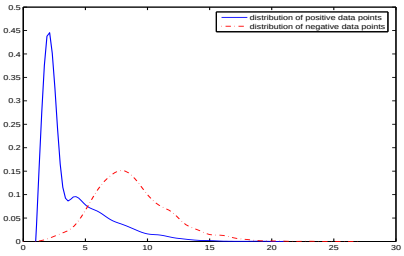


(c) Sum of Papers count

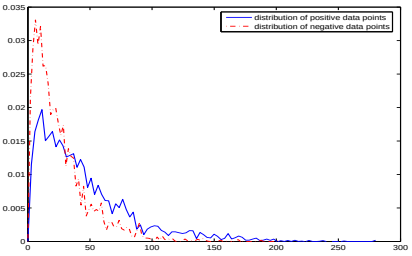


(d) Shortest distance

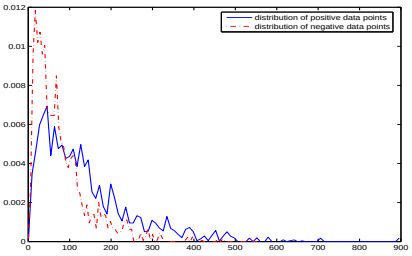
Figure 1: Evaluation of features using class density distribution in BIOBASE dataset



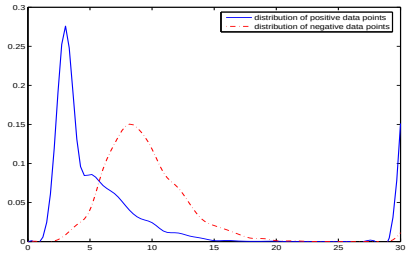
(a) Shortest Distance



(b) Sum of paper count



(c) Sum of neighbors count



(d) Second shortest distance

Figure 2: Evaluation of features using class density distribution in DBLP dataset

Classification model	Accuracy	Precision	Recall	F-value	Squared Error
Decision Tree	90.01	91.60	89.10	90.40	0.1306
SVM(Linear Kernel)	87.78	92.80	83.18	86.82	0.1221
SVM(RBF Kernel)	90.56	92.43	88.66	90.51	0.0945
K_Nearest Neighbors	88.17	92.26	83.63	87.73	0.1826
Multilayer Perceptron	89.78	93.00	87.10	90.00	0.1387
RBF Network	83.31	94.90	72.10	81.90	0.2542
Naive Bayes	83.32	95.10	71.90	81.90	0.1665
Bagging	90.87	92.5	90.00	91.23	0.1288

Table 2: Performance of different classification algorithms for BIOBASE database

Classification model	Accuracy	Precision	Recall	F-value	Squared Error
Decision Tree	82.56	87.70	79.5	83.40	0.3569
SVM(Linear Kernel)	83.04	85.88	82.92	84.37	0.1818
SVM(RBF Kernel)	83.18	87.66	80.93	84.16	0.1760
K_Nearest Neighbors	82.42	85.10	82.52	83.79	0.2354
Multilayer Perceptron	82.73	87.70	80.20	83.70	0.3481
RBF Network	78.49	78.90	83.40	81.10	0.4041
Naive Bayes	81.24	87.60	76.90	81.90	0.4073
Bagging	82.13	86.70	80.00	83.22	0.3509

Table 3: Performance of different classification algorithms for DBLP dataset

various feature values. For comparison sake, we normalize the distribution so that the area under both the curves is the same. For most of the features, the distribution of positive and negative class exhibit significant difference, thus facilitating the classification algorithm to pick patterns from the feature value to correctly classify the samples. However, there is a small overlap region between the distributions for some features. The fraction of population that lies in the critical overlap region for most of the features are most likely the candidates for misclassification. We shall discuss more about the distribution later.

Among all the classifiers, RBF network model performs the worst in both the datasets and may not be the one that is suitable for the link prediction problem. RBF networks are usually affected severely by irrelevant or inconsistent features and link prediction datasets are heavily noisy, hence, the performance value for RBF is poor. On the other hand, we have naive Bayes algorithm, which also performed bad. Naive Bayes is probably not powerful enough to catch the patterns in the data set which are helpful for classification.

In the same tables, we also list Precision, Recall and F-value for the positive class. F-value is the harmonic mean of precision-recall that is sometimes considered a better performance measure for a classification model in comparison to accuracy, especially if the population of the classes are biased in the training dataset. Considering the F-value metric, the rank of the classifiers do not really change, indicating that all the models have similar precision-recall behavior. Now, comparing the precision and recall columns, we find that most of the classifiers have precision value significantly higher than the recall value for the positive class. This indicates that our models have more false negatives than false positives. Intuitively, the models are missing actual links more than they are predicting false links. For coauthorship network, it makes sense because there exist some coauthor pairs that seem to coauthor merely by coincidence. Moreover, it can happen that the link is actually not there in real life also, but the dataset had it because of name aggregation. Note that in the dataset that we used, all the names that had the same spelling were considered to be the same person, which is not always correct. This problem has been addressed in many concurrent researches and several entity disambiguation methodologies have been proposed [18, 4] to cope with it. So, a better performance will be observed, if such methodologies are applied to the dataset as a preprocessing step before feeding it into the learning algorithms.

Finally, we use the average squared error as our last performance comparison metric. Recent research [6] shows that this metric is remarkably robust and has

Attribute name	Information gain	Gain Ratio	Chi-Square Attribute Eval.	SVM feature evaluator	Avg. Rank
Sum of Papers	3	4	3	4	3
Sum of Neighbors	1	3	1	2	2
Sum of KW count	6	6	6	3	5
Sum of Classification count	5	5	5	6	5
KW match count	2	1	2	1	1
Sum of log of Sec. Neighbor. count	7	7	7	8	7
Shortest distance	4	2	4	5	4
Clustering Index	9	9	9	7	8
Shortest dist. in KW-Author graph	8	8	8	9	8

Table 4: Rank of different Attributes using different algorithms for BIOBASE dataset

Attribute name	Information gain	Gain Ratio	Chi-Square Attribute Eval.	SVM feature evaluator	Avg. Rank
Sum of Papers	4	4	4	2	4
Sum of Neighbors	3	3	3	4	3
Shortest distance	1	1	1	1	1
Second shortest distance	2	2	2	3	2

Table 5: Rank of different Attributes using different algorithms for DBLP dataset

higher average correlation to the other metrics, hence an excellent metric to compare the performance of different classifiers. However, finding average squared error in binary classification setup requires predicting the posterior probability instead of predicting just the class label. In fact, a model that could predict the true underlying probability for each test case would be optimal [6]. From the probability, squared error can be computed very easily. In a unbiased environment, the cost associated with the misclassification of positive and the negative class is the same and no calibration of probability is required. So, if the value of the predicting probability is above 0.5, the sample is predicted as positive class and the difference of 1 and the value is considered the error. In contrast, if the value is below 0.5, the sample is predicted as negative class and the difference of 0 and the value is considered to be the error. In the worst case, we have an error value of 0.5 and the label can be predicted only by tossing a fair coin. Finally, a root mean squared error is computed over all the samples. We used the above discussed approach while computing the squared error. Here, we observe a dramatic difference in performances among the different classifiers. SVM (RBF) outperforms all the others in this metric with a healthy margin in both the datasets. In both the datasets, squared error of SVM is more than 30% less than the second best algorithm. This confirms its effectiveness over all the other classification algorithms for the link prediction task.

One of our objectives was to compare the features to judge their relative strength in a link prediction task. We ran several algorithms for this. Table 4 and 5 provide a comparison of several features by showing their rank of importance as obtained by different algorithms. Last column shows an average rank that is rounded to the nearest integer.

From the result shown in table 4, in BIOBASE dataset the **keyword match count** was the top ranked attribute. **Sum of neighbors** and **sum of papers** come next in the order of significance. **Shortest distance** is the top ranked among the topological features. From the figure 1 that shows the distribution of some powerful features, we can easily understand the reasoning behind the ranking. For instance, in the keyword match feature, no negative class sample had more than 5 keyword matches and about 95% samples had the match value equal to zero. Whereas, positive class samples have keyword match values from 0 to 20, and the distribution mean is almost equal to 6. Similar noticeable differences in distribution are also observed for other features. From the ranking algorithm, **clustering index** and **author-keyword**

distance are found to be the lowest ranked attributes. Some researchers indicated that clustering index is a significant attribute for link prediction, but at least in BIOBASE dataset it does not seem to have that promising effect.

From the results shown in table 5, shortest distance is the best feature in DBLP dataset. The strength of this feature is also well presented by the distribution shown in figure 2. From this figure, for positive class the mean distance between the author pairs is around 3, whereas the same for the negative class is almost 8. In this dataset, we also used second shortest distance, which is the distance calculated from another shortest path that has no common edge with the first shortest path. The mean value for positive class here is around 4 and that for negative class is around 9. Similar differences in distribution are also observed for the other two features, like sum of papers and sum of authors. Note that, for both the features, the negative class is concentrated heavily towards the smaller feature values compared to the positive class. Ranking algorithms ranks the attributes in the following order: shortest distance, second shortest distance, sum of papers and sum of neighbors. This order properly reflects the distribution patterns shown in figure 2.

5 Issues regarding Real-life Dataset

From the results and discussions in the previous section, readers must be convinced that link prediction can be solved with high accuracy using very few features in supervised learning setup. However, in real life there exists several issues to be dealt with to obtain such a satisfactory performance. Since, most serious applications of link prediction in recent days is in the domain of security and anti-terrorism, majority of discussions implicitly assume such an application.

In our experiments, we used standard cross-validation approach to report the performance, so training and testing datasets are drawn from the same distribution. In real life, data comes from heterogeneous sources and an analyst needs to make sure that the classification model that is used on a testing dataset is built from a dataset with the same distribution; without it, the result from the algorithms can be completely useless. Distribution of the feature values can be analyzed to understand whether there are any noticeable differences between training and testing dataset. If it is suspected that the distribution is different, a probability value instead of class label should be predicted. Then the probability should be calibrated accordingly for the testing dataset to predict the class label.

Sometimes, datasets can be highly imbalanced. If we are looking for links, that represent rare events, the number of samples with positive classes could be exceptionally low. Highly imbalanced dataset deteriorate the performance of the classification algorithms and special care should be taken for that. Fortunately, there are algorithms [12, 25] that have been adapted for imbalanced datasets, so an approach outlined in these algorithms should be followed in this situation.

For link prediction, specially in security applications, missing actual links poses severe threat compared to predicting false links. So, a very high value of recall is desired. This requires that we bias the model towards predicting more of the positive class than to predicting the negative class. It can be easily achieved in the classification model, specially in those that are norm-based, like SVM, k-nearest neighbors, etc. by assigning a suitable higher cost to the misclassification of positive class.

In terrorist social networks, finding samples to train a supervised classification model poses another big challenge. Although huge efforts are being employed to obtain terrorism related information, the strong counter effort from the terrorist groups to hide their connections undermines the effectiveness of the data extraction. In this situation, data could be highly noisy, and even worse, some of the attribute values could be unknown. The performance of the link prediction can deteriorate significantly in that case. Fortunately, there are classification algorithms [23], that have been developed to work around the missing values. Moreover, information in the datasets are changing in real-time, so the classifier models need to be updated frequently.

6 Future Work

Our research currently considers link prediction only in the coauthorship domain. In future, we would like to consider a number of datasets from different domains to better understand the link prediction problem. We would also like to define a degree of confidence for link prediction instead of providing a hard binary classification.

Moreover, our current attribute set does not have any attributes that capture causal relationships. It is possible that some of the attribute values that we consider are time dependent, i.e. their values should be evaluated by using different weights for different years. In future, we like to consider these kind of attributes. There are online social networks, such as LinkedIn (`http://www.linkedin.com`) and Friendster (`http://www.friendster.com`), where they will be very useful. These online networks would like to predict which users would share common interests. These interests are likely to change over time which would affect the likelihood of a link between two users. This is similar to keeping

track of dynamic user groups.

7 Conclusion

Link prediction in a social network is an important problem and it is very helpful in analyzing and understanding social groups. Such understanding can lead to efficient implementation of tools to identify hidden groups or to find missing members of groups, etc. which are the most common problems in security and criminal investigation research. In this research we suggest categories of features that should be considered for link prediction in any social network application. Of course, the exact value of a feature would depend on the application at hand. For example, in a terrorist network, two terrorists could have strong proximity either if they have the same skills or if they have complementary skills.

Through this work, we have shown that the link prediction problem can be handled effectively by modeling it as a classification problem. We have shown that most of the popular classification models can solve the problem with an acceptable accuracy, but the state of the art classification algorithm, SVM, beats all of them in many performance metrics. Moreover, we provided a comparison of the features and ranked them according to their prediction ability using different feature analysis algorithms. We believe that these ranks are meaningful and can help other researchers to choose attributes for link prediction problem in a similar domain.

8 Acknowledgment

We like to thank Central Intelligence Agency (CIA) and IBM for providing support for this research via IBM Contract No. 2003*S518700*00. The BIOBASE dataset was provided to us by CIA through the KDD Challenge Program, 2005.

References

[1] L. A. Adamic and E. Adar, *Friend and Neighbors on the Web*, Social Networks, 25(3), 2003, pp. 211-300.

[2] A. L. Barabasi, H. Jeong, Z. Neda, A Schubert and T Vicsek, *Evolution of the Social Network for Scientific Collaboration*, PHYSICA A, 311 (2002), pp. 3.

[3] J. Baumes and M. M. Ismail, *Finding Communities by Clustering a Graph into Overlapping Subgraph*, Intl. Conf. on Applied Computing,(2005).

[4] I. Bhattacharya and L. Getoor, *Deduplication and Group Detection using Links.*, LinkKDD, 2004.

[5] D. Cai, Z. Shao, X. He, X. Yan and J. Han, *Mining Hidden Communities in Heterogeneous Social Networks*, LinkKDD 2005.

[6] R. Caruana and A. Niculescu-Mizil, *Data Mining in Metric Space: An Empirical Analysis of Supervised Learning Performance Criteria*, KDD 2004.

[7] R. Caruana, A. Niculescu-Mizil, G. Crew and A. Ksikes, *Ensemble Selection from Libraries of Models*, Intl. Conf. of Machine Learning, 2004.

[8] R. Cross, A. Parket, L. Prusak and S. Borgatti, *Knowing What We Know: Supporting Knowledge Creation and Sharing in Social Networks*, Organizational Dynamics, 30 (2001), pp. 100-120.

[9] S. N. Dorogovtsev, J. F. Mendes, *Evolution of Networks*, Advan. Physics., 51 (2002), pp. 1079.

[10] C. Faloutsos, K. McCurley and A. Tomkins, *Fast Discovery of Connection Subgraphs*, Intl. Conf. Knowledge. Discovery and Data Mining, 2004.

[11] A. Goldenberg, and A. W. Moore, *Bayes Net Graphs to Understand Co-authorship Networks?*, LinkKDD 2005.

[12] G. Gu and E. Chang, *Aligning Boundary in Kernel Space for Learning Imbalanced Dataset*, ICDM, 2004.

[13] Q. Hu, X. Zhang and D .Saha, *Modeling Virus and Anti-Virus Dynamics in Topology-Aware Networks*, GLOBECOM, 2004, IEEE.

[14] Z. Huang, X. Li and H. Chen, *Link Prediction Approach to Collaborative Filtering*, Join Conference on Digital Libraries, Denver, CO, 2005

[15] Z. Huang and D. Zeng, *Why Does Collaborative Filtering Work? - Recommendation Model Validation and Selection by Analyzing Bipartite Random Graphs*, Workshop of Information Technologies and Systems, Las Vegas, NV, 2005

[16] J. M. Kleinberg, *Navigation in a Small World*, Nature, 406 (2000), pp 845.

[17] D. Liben-Nowell and J. Kleinberg, *The Link Prediction Problem for Social Networks*, LinkKDD, 2004.

[18] B. Malin, *Unsupervised Name Disambiguation via Social Network Similarity.* Workshop on Link Analysis, Counter-terrorism, and Security, 2005.

[19] S. Milgram, *The Small World Problem*, Psychology Today, 2 (1967), pp 60-67.

[20] M. J. Newman, *The Structure of Scientific Collaboration Networks*, Proc. National Acad. of Sciences, 98 (2001), pp 404-409.

[21] M. J. Newman, *Fast Algorithm for Detecting Community Structure in Networks*, LinkKDD 2004.

[22] M. Ohira, N. Ohsugi, T. Ohoka and K. Matsumoto, *Accelerating Cross-Project Knowledge Collaboration using Collaborative Filtering and Social Networks*, Intl. Conf. of Software Engineering, 2005.

[23] J. Myers, K. Laskey, and K. DeJong, *Learning Bayesian Networks from Incomplete Data using Evolutionary Algorithms*, Fifteen Conf. on Uncertainty in Artificial Intelligence, Toronto, Canada, 1999

[24] I. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann, San Francisco, 2005.

[25] Z. Zheng, X. Wu, R. Srihari, *Feature Selection for Text Categorization on Imbalanced Data*, SIGKDD Explorations, 2002, pp. 80-89.