Local Probabilistic Models for Link Prediction[*]

Chao Wang Venu Satuluri Srinivasan Parthasarathy
Department of Computer Science and Engineering, The Ohio State University

Abstract

One of the core tasks in social network analysis is to predict the formation of links (i.e. various types of relationships) over time. Previous research has generally represented the social network in the form of a graph and has leveraged topological and semantic measures of similarity between two nodes to evaluate the probability of link formation. Here we introduce a novel local probabilistic graphical model method that can scale to large graphs to estimate the joint co-occurrence probability of two nodes. Such a probability measure captures information that is not captured by either topological measures or measures of semantic similarity, which are the dominant measures used for link prediction. We demonstrate the effectiveness of the co-occurrence probability feature by using it both in isolation and in combination with other topological and semantic features for predicting co-authorship collaborations on three real datasets.

1 Introduction

In recent times, there has been a lot of interest in understanding and characterizing the properties of large scale networks or graphs. Part of this interest is because of the generality of the graph model: many domains, such as social networks, gene regulatory networks and the World Wide Web can be naturally thought of as graphs. An important problem in this context is that of link prediction. Informally link prediction is concerned with the problem of predicting the (future) existence of links among nodes in a graph. Link prediction is useful in many application domains, ranging from recommender systems to the detection of unseen links in terrorism networks, from protein interaction networks to the prediction of collaborations among scientists, and from prediction of friendship formations to the prediction of web hyperlinks.

In this article we focus on the problem of link prediction particularly in the context of evolving co-authorship networks. This has been a hotbed of recent research activity where much of the focus has been on encapsulating the topological and/or semantic information embedded in such networks to address the link prediction problem. In contrast in this article we explore the realm of probabilistic models derived from frequency statistics and use the resulting predictions from the probabilistic models as additional features to further enhance predictions made by topological-based and semantic-based link prediction algorithms.

Specifically our probabilistic model is driven by two aspects. First, given the candidate link (say between nodes X and Y) whose probability is to be estimated, we identify the *central neighborhood set* (say W, X, Y, Z), which are the nodes that are deemed germane to the estimation procedure. The identification of the central neighborhood set is governed by the local topology of the social network as viewed from the perspective of the two nodes whose link probability is to be estimated.

Second, once the central neighborhood set(W, X, Y, Z) is identified we learn a maximum entropy Markov random field model that estimates the joint probability of the nodes comprising the central neighborhood set, i.e., $p(W, X, Y, Z)$. In this context one can leverage the fact that most co-authorship networks are computed from an event log (an event corresponding to a publication). Multi-way statistics (e.g. non-derivable frequent itemsets [4] whose elements are drawn from (W, X, Y, Z)) on these event logs can be used to constrain and learn the model parameters efficiently [19]. The resulting model can then be used to estimate the link probability between X and Y which we henceforth denote as the *co-occurrence probability*.

In our empirical results we demonstrate that the co-occurrence probabilities inferred from the resulting model can be computed in a scalable manner and is highly discriminatory for link prediction when compared with state-of-the-art topological and semantic features on several real world datasets. Moreover, we demonstrate that the resulting co-occurrence probability can also be effectively combined with these other features and then one can employ any classification algorithm to predict if a link will be formed between two nodes. Specifically, we employ a simple yet novel variant of the Katz score as a topological feature, one

[*] This work is supported in part by the following research grants: DOE Award No. DE-FG02-04ER25611; NSF CAREER Grant IIS-0347662; Contact email - srini@cse.ohio-state.edu

that scales reasonably well at some cost to accuracy. Additionally we describe and use straightforward state-of-the-art methods to measure the semantic overlap among nodes based on the topics they work on, to further enhance the feature vector and improve overall link prediction performance.

2 Related Work

The seminal work of Liben-Nowell and Kleinberg [12] was the first comprehensive study on the utility of topological features derived from graphs for predicting links in social networks. They examine various topological features, including graph shortest distance, common neighbors, preferential attachment, Adamic-Adar, Jaccard, SimRank, hitting time, rooted PageRank, and Katz. They find that topological information is quite useful when compared to a random predictor. In particular, Adamic-Adar and the Katz measure appear to be more effective than the other topological features. Recently, Huang [7] proposes to use another topological feature – generalized clustering coefficient – to solve the link prediction problem.

An important limitation of these works is that they only use a single (topological) feature for the link prediction task. Intuitively, it seems that one can achieve better performance by utilizing the other sources of information, such as the content or semantic attributes of the nodes. A natural way to do this would be to use the multiple sources of information as features to be fed into a classifier that is trained to discriminate between positive instances (i.e. links that form) and negative instances (links that do not form) by making use of all the features. This is the approach adopted by Hasan *et al.* [6] and O'Madadhain *et al.* [14]. Hasan *et al.* [6] have used topological features (such as the shortest distance between the two nodes), aggregated features (such as the sum of neighbors) and semantic features (such as the number of matching keywords). They report keyword match count to be their most useful feature on one dataset, which indicates that a lot is to be gained by taking into consideration the semantic similarity in the publications of the two authors. O'Madadhain *et al.* [14] also have investigated the use of content-based attributes such as the KL-divergence of the topic distributions of the two nodes, their geographic proximity, and similarity of journal publication patterns.

The work of Popescul *et al.* [16] is another interesting approach to integrating different kinds of information. They represent the data in a relational format, generate candidates for features through database join queries, select features using statistical model selection criteria and use Logistic Regression using the selected features for classification. A potential problem with this approach is that the features so generated are simple aggregation functions of the column values in the result set of the join queries (such as count and average). A more complex feature such as cosine similarity between bag-of-words representations cannot be easily expressed using simple SQL aggregation functions.

Researchers have also examined the use of probabilistic models for solving the link prediction problem. Taskar *et al.* [18] use discriminatively trained relational Markov networks to define a joint probabilistic model over the entire graph (i.e. over the links as well as the content attributes of the nodes). The trained model is used to collectively classify the test data. Kashima and Abe [10] propose a parameterized probabilist model of network evolution and then use it for link prediction. They assume the network structure is in a stationary state and propose an EM algorithm to estimate model parameters. They report encouraging results on two small biological datasets. However, both collective classification and training global probabilistic models can be expensive to compute and typically do not scale well to medium and large scale networks.

In related work, Rattigan and Jensen [17] argue that the link prediction problem is too hard to solve because of the extreme class skew problem. Social networks are usually very sparse and positive links only hold a very small amount of all possible pairs of nodes. As an alternative, they propose a simpler problem – anomalous link discovery. Specifically, they constrain their focus on the links that have been formed and infer their anomaly (surprisingness) scores from the previous data.

3 Methods

We consider three sources of information of the network data for link prediction. We have a large number of local events that are accumulated along time, where by local event we mean an interaction among a set of objects in the network. For example, the publication of a paper would represent a local event involving all the authors of the paper. We refer to the collection of such local events as the event log. This is the raw format of the network data and provides the first source of information for link prediction. Such an event log is typically converted to a graph representation, in which nodes represent objects in the network and two nodes are connected to each other if they co-occur in at least one local event. This graph provides the second source of information for link prediction. Finally, we have access to other attributes of an object in the network, such as the research areas of the authors in an author collaboration network, usually referred to as content or semantic information. This semantic information provides the third source of information for link prediction.

It is difficult to capture all information from different sources with one single feature. For this reason, we examine three types of features – co-occurrence probability features, topological features and semantic features - coming from the first, second and third source, respectively. In the text below, we discuss how we derive these features in turn.

3.1 Deriving Co-occurrence Probability Features

For a pair of nodes that have never co-occurred in the event log, our aim is to estimate the chances of their co-occurring in the future, i.e. of a link forming in the future between those two nodes. In order to estimate the co-occurrence probability of the given two nodes in a principled manner, we use probabilistic graphical models. Specifically, we employ undirected graphical models, also called *Markov Random Fields (MRFs)*, to model the local neighborhood containing the two nodes. We stress the fact that we build a local model, rather than a global model, as building global models can become prohibitively expensive for large scale networks.

There are two main stages in our approach to use graphical models in this context - (a) determining the nodes that will be included in the local model, and (b) using frequent non-derivable itemsets to determine the structure of the graphical model as well as learn the parameters of the model. Once the model is learned, we use exact inference techniques to determine the co-occurrence probability of the pair of nodes under consideration. We need not resort to approximate inference, as we build a local model, leading to a low treewidth (maximum clique size in the graph formed by triangulating the model minus 1) for the model.

3.1.1 Determining the Central Neighborhood Set of Two Nodes

For a given pair of nodes, we retrieve a small set of nodes that we believe to be most relevant to estimating the co-occurrence probability of the given pair of nodes. We refer to this set of nodes as the *central neighborhood set* of the two nodes. At one extreme, we can include any node that lies on any path between the two nodes as belonging to the central neighborhood set. However, this would lead to a large probabilistic model, over which learning can be expensive. For this reason, we set a parameter *size* that specifies the number of nodes to be present in the central neighborhood set.

We need a mechanism of selecting the nodes to include in the central neighborhood set. Intuitively, the nodes that lie along paths of shorter length are more relevant. Hence, we propose a method of enumerating simple paths length-wise, i.e. we first collect all nodes that lie on length-2 simple paths, and then those on length-3 simple paths and so on. (A simple path is a path without cycles.) The algorithm for enumerating all simple paths of a given length is presented in Figure 1. However, this order of enumerating nodes may not be enough as there may be many nodes that lie on paths of a given length. Hence, we need a way of ordering paths of the same length. For this purpose, we define the *frequency score* of a path as the sum of the occurrence counts of all nodes along the paths. Now, among paths of the same length, we enumerate paths with higher frequency scores before paths with lower frequency scores. The pseudo-code of the full algorithm for selecting the central neighborhood set of a pair of nodes is presented in Figure 2.

```
Enumerate_Simple_Paths($G$, $s$, $t$, $K$)
 $Input:\ G, a\ graph;\ s,\ starting\ node;\ t,\ ending\ node;$
 $K,\ path\ length$
 $Output: P,\ a\ set\ of\ simple\ paths\ of\ length\ K$

 $\{*\ Find\ distance\text{-}(K-1)\ neighbors\ of\ s$ without
 $visiting\ t\ and\ bookkeeping\ all\ path\ information\ *\}$
 $N = Breadth - first - search(G, s, K-1, t);$

 foreach $(e \in N)$
     $\{*\ If\ e\ and\ t\ are\ connected\ in\ G\ *\}$
     if $(e.Connect(t,\ G))$
         $add\ path(s \rightarrow e \rightarrow t)\ to\ P;$
 return $P$
```

Figure 1. Enumerating all simple paths of length K between two nodes

```
Select_Central_Node_Set($G$, $s$, $t$, $maxSize$)
 $Input:\ G, a\ graph;\ s,\ staring\ node;\ t,\ ending\ node;$
 $maxSize,\ central\ neighborhood\ set\ size\ threshold;$
 $Output:\ C,\ central\ neighborhood\ set\ between\ s\ and\ t;$

 $C \leftarrow \emptyset;$
 $i \leftarrow 2;$
 while $i < PATH\_LENGTH\_THRESHOLD$
     $P_i \leftarrow Enumerate\_Simple\_Paths(G, s, t, i);$
     $Sort\ all\ paths\ \in\ P_i\ by\ length\ and\ frequency\ score;$
     foreach $path\ p \in P_i$
         if $|C| < maxSize$
             $add\ all\ nodes\ along\ p\ to\ C;$
     $i \leftarrow i + 1;$
 return $(C)$
```

Figure 2. Selecting the central neighborhood set for two nodes

We also use a path length threshold in the algorithm, because in practice, we cannot afford to enumerate all simple paths between two nodes in a large graph. In our study, we consider paths up to length 4 because we found that this threshold works well in capturing the contextual information between two nodes. In situations where there does not exist such paths between two nodes, we define the central neighborhood set to be the two nodes themselves. Interestingly, we note that in this case, the local probabilistic model reduces to a simple independence model.

3.1.2 Learning Local Markov Random Fields

For a given pair of nodes and their corresponding central neighborhood set, how should one learn a local probabilistic model for it? We adopt an approach of using non-derivable frequent itemsets from the underlying network log events data to learn local probabilistic models. The event log is essentially a transactional dataset and we apply widely-used frequent itemset mining techniques [2, 20, 5] on it to collect occurrence statistics of network objects. These statistics can be leveraged afterward to learn local probabilistic models on the central neighborhood set. The basic idea of using a set of itemsets to learn a model is as follow: Each itemset and its occurrence statistic can be viewed as a constraint on the underlying unknown distribution. A model that satisfies all present occurrence constraints and in the meanwhile has the *maximum entropy* ("as uniform as possible") is used as the estimate of the underlying unknown distribution. One can verify that this maximum entropy distribution specifies a *Markov Random Field*. More information about this can be found in [15, 19].

In our study we pre-compute all frequent itemsets from the underlying log events. Social networks are usually very sparse – the proportion of formed links is very low as opposed to the number of all possible pairs of nodes. For this reason we use a support threshold of one to collect frequent itemset patterns. As a result, all positive occurrence evidence will be captured. We note however, at this support threshold level, frequent itemsets are too many to use. Fortunately, we only need to mine and use *non-derivable itemsets* for model learning. Simply speaking, non-derivable itemsets are those itemsets whose occurrence statistics can not be inferred from other itemset patterns. As such, non-derivable itemsets provide non-redundant constraints and we can employ them to learn probabilistic models without any information loss. Calders *et al.* [4] propose an efficient depth-first search method to mine non-derivable itemsets. We use their implementation to mine non-derivable itemsets.

```
Co-occurrence_Probability_Feature_Induction$(C, s, t, NDI)$
 $Input:\ C,\ central\ neighborhood\ set\ between\ s\ and\ t;$
 $s,\ starting\ node;\ t,\ ending\ node;$
 $NDI,\ collection\ of\ non\ derivable\ itemsets$
 $Output:\ f,\ co\text{-}occurrence\ probability\ feature\ of\ s\ and\ t$

 $\{*\ Retrieve\ ndi\ patterns\ relevant\ to\ C\ *\}$
 $R \leftarrow \emptyset$
 foreach $(ndi \in NDI)$
     if $(ndi \in C)$
         $add\ ndi\ to\ R;$

 $\{*\ Learn\ an\ MRF\ model\ on\ C\ using\ R*\}$
 $M\ =\ learn\_MRF(C,\ R);$

 $\{*\ Infer\ co-occurrence\ prob.\ of\ s\ and\ t\ from\ M*\}$
 $f = Inference(M,$ s, t$);$
 return $(f)$


Learn_MRF$(C, R)$
 $Input: C,\ common\ neighborhood;$
 $R,\ collection\ of\ itemsets;$
 $Output:\ MRF\ M$

 $\{*\ Obtain\ all\ variables\ in\ C\ and\ initialize\ M*\}$
 $Initialize\_Parameters(M);$

 while $(Not\ all\ constraints\ are\ satisfied)$
 foreach $constraint\ e \in R$
     $Update\ M\ to\ force\ it\ to\ satisfy\ e;$
 return $(M);$
```

Figure 3. Inducing co-occurrence probability feature for a pair of nodes

To predict if two nodes will be linked, we first identify from the network the central neighborhood set of two involved nodes. Then we select all itemsets that lie entirely within this set and use them as evidence to learn an MRF. The learned MRF is local in that it specifies a joint distribution over only those nodes in this set. Then we estimate the joint co-occurrence probability feature of the link through inference over the local model. The formal algorithm is presented in Figure 3. We illustrate this whole process with a simple example presented in Figure 4. Assume we want to predict the link between nodes a and b and there are two paths connecting them in the graph: $p_1 = a \rightarrow c \rightarrow b$ and $p_2 = a \rightarrow d \rightarrow e \rightarrow b$. Also assume that we use both p_1 and p_2 to identify the central neighborhood set between a and b. As a result, the central neighborhood set C is given by: $C = \{a, b, c, d, e\}$. Next we retrieve all non-derivable itemsets that lie entirely within this set. Let us assume that the itemsets retrieved are: {a, b, c, d, e, ac, ad, bc, be, de}. Their occurrence statistics are collected from the log events and are presented in the figure. We employ all of these patterns and their occurrence statistics to learn a local probabilistic model M over C: $M = P(a, b, c, d, e)$. M specifies a joint distribution on all variables in C and its clique potential functions are listed as follows (μ's are model parameters to be learned and $I()$ is indicator function). The shaded area

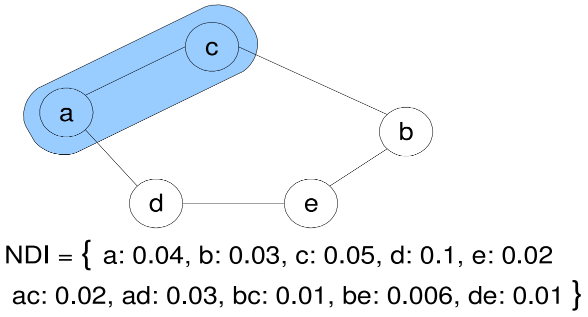

c
a
b
d
e
NDI = { a: 0.04, b: 0.03, c: 0.05, d: 0.1, e: 0.02
ac: 0.02, ad: 0.03, bc: 0.01, be: 0.006, de: 0.01 }

Figure 4. An example of local model-based co-occurrence probability feature induction

in Figure 4 shows a clique of M −{a,c}.

$$\begin{aligned} \psi_{a,c} = \mu_1^{I(a=1)} \cdot \mu_2^{I(c=1)} \cdot \mu_3^{I(a=c=1)} \\ \psi_{a,d} = \mu_4^{I(d=1)} \cdot \mu_5^{I(a=d=1)} \\ \psi_{b,c} = \mu_6^{I(b=1)} \cdot \mu_7^{I(b=c=1)} \\ \psi_{b,e} = \mu_8^{I(e=1)} \cdot \mu_9^{I(b=e=1)} \\ \psi_{d,e} = \mu_{10}^{I(d=e=1)} \end{aligned}$$

Then we derive the co-occurrence probability of a and b by computing the marginal probability of $p(a = 1, b = 1)$ on M.

We use an iterative scaling algorithm [9] *Learn_MRF()* to learn a local MRF for the central neighborhood set. The idea is to iterate over all itemset constraints and repeatedly update the model to force it to satisfy the current itemset constraint, until the model converges. After the model is constructed, we do inference over it to estimate the joint co-occurrence probability of s and t. For the *Inference()* procedure in the algorithm, we can plug in exact inference algorithms for it since our model is local. In our study, we use the *Junction Tree* inference algorithm [11].

3.2 Deriving Topological Features

The Katz measure is a weighted sum of the number of paths in the graph that connect two nodes, with shorter paths being given the more weight. This leads to the following measure:

$$Katz(s,t) = \Sigma_{i=1}^{\infty} \beta^i p_i$$

Here p_i is the number of paths of length i connecting nodes s and t, while β is a damping factor. It has been shown that Katz is among the most effective topological measures for the link prediction task [12]. It outperforms *shortest distance*, *hitting time* and many others. It can be verified that the matrix of Katz scores can be computed by $(I - \beta M)^{-1} - I$, where M is the adjacency matrix of the graph [12].

However, this method does not scale well to handle the network data under the consideration since computing matrix inverse for large graphs is very expensive. As such, we come up with a way to approximately compute the Katz score. Specifically, we only consider paths of length up to certain threshold to compute the Katz score. The new measure is as follows:

$$aKatz(s,t) = \Sigma_{i=1}^{k} \beta^i p_i$$

where p_i and β have the same meaning as above, while k is a new input parameter specifying the maximum path length we consider. Since the score terms damp exponentially with the longer length, this new measure captures the most significant portion of the exact Katz score. We find that k of 4 can give a good approximation of the Katz scores in practice. We design graph algorithms to evaluate this new measure. To this end, we follow the similar process of identifying the central neighborhood set for two nodes shown above, enumerate all simple paths up to length k from s to t and use the above formula to compute an approximate Katz score. We execute breadth-first-search from s up to k levels without visiting t, while keeping track of all paths formed so far. We will denote this approximate Katz measure as *aKatz* throughout the rest of the paper.

3.3 Deriving Semantic Features

The degree of semantic similarity among entities is something that can be useful to predict links that might not be captured by either topological or frequency-based features. For example, in the context of co-authorship networks, we use the following method to compute the semantic similarity for two authors:

1. Collect the words in the titles of each author (removing stop words), so that an author is represented as a set of words (akin to a text document).

2. Derive a bag of words representation for each author, weighting each word by its TFIDF (Term Frequency - Inverse Document Frequency) measure.

3. Compute the cosine between the TFIDF feature vectors of the two authors whose semantic similarity we need to determine.

Previously, Hasan *et al.* [6] have used keyword match count between two authors as a feature and have reported the feature to be the most useful feature. Our method for computing semantic similarity makes use of the well-known techniques such as TFIDF feature vector representation and the cosine measure to compute similarity - the former can weight words automatically and the latter is a widely-used and effective measure for computing similarity between text documents represented in the vector space model.

3.4 Combining Different Features Using Supervised Learning Framework

Since we have three types of features - the co-occurrence probability feature, the topological similarity feature and the semantic similarity feature - we need an effective way to combine these features. For this we resort to supervised learning. In order to do this, we need to come up with a way to partition the original dataset into training and testing datasets. A supervised learning approach to the link prediction problem has been taken previously by Hasan *et al.* [6] and Madadhain *et al.* [14], and the two works have taken different approaches to partitioning the dataset into training and testing sets. We find the approach taken by Madadhain *et al.* [14] to be cleaner and follow the same, which we describe below. An illustration of our approach can be found in Figure 5.

We form a labeled training dataset as follows: we take all the links that are formed in the 9th year (T9 in Figure 5) and label them as positive training instances. Of the links that are not formed in the first 9 years, we randomly sample a subset and label them as negative training instances. We sample 10 times as many negative instances as positive instances. The features for each of these instances are constructed from the first 8 years of data. A classifier is then trained on the labeled training set - any off-the-shelf classifier can be used, we chose to use Logistic Regression[1], since it is computationally efficient, and produces well-calibrated class probabilities that can be used to rank predictions.

The testing dataset is formed in a similar fashion: the links that are formed in the 10th year (T10 in Figure 5) are treated as testing instances that need to be predicted as positive, and we include a sample of the links that are not formed in the whole of the dataset as testing instances whose ground truth labeling is negative. The features that are used by the classifier trained previously are formed from the first 9 years of data.

4 Evaluation

In this section, we report the experimental results of our proposed approach.

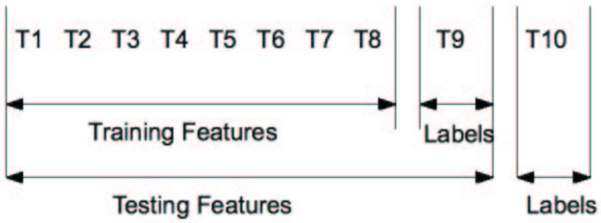


Figure 5. Split of the datasets into Training and Testing data. T1, T2, .. , T10 represent the 10 time intervals the dataset spans.

Dataset	No. of authors	No. of papers	No. of edges
DBLP	23136	18613	56829
Genetics	41846	12074	164690
Biochemistry	50119	16072	191250

Table 1. Summary of the three datasets that were constructed

4.1 Datasets

We evaluated performance on three real datasets in all, described below. The details of the datasets are summarized in Table 1.

- The DBLP dataset was generated using the DBLP collection of Computer Science articles[2]. This dataset contains the publication details of the proceedings of 28 conferences related to Data Mining, Databases and Machine Learning from the years 1997 to 2006.

- The Genetics dataset contains articles published from 1996 to 2005 in 14 journals related to genetics and molecular biology.

- The Biochemistry dataset contains articles published from 1996 to 2005 in 5 journals related to biochemistry.

The Genetics and the Biochemistry datasets were generated from the PubMed database.[3]

4.2 Class Conditional Distributions of the Different Features

First we examine the distribution of the features among the positive and negative examples. Figure 6a-c plot the distribution for three features on the Genetics dataset. The

[1] We used the Logistic Regression implementation in the popular WEKA suite of Data Mining algorithms

[2] DBLP is located online at http://dblp.uni-trier.de/

[3] The PubMed database can be accessed online at http://www.ncbi.nlm.nih.gov/entrez/

results on the other two datasets are similar and are not plotted here due to space constraints.

One can see that both the co-occurrence probability and the aKatz measure can discriminate among the negative and positive instances. The main difficulty in the link prediction task occurs because the number of links that do not form far outweighs the number of links that do form. The ratio of negative instances to positive instances in our workload is $10:1$.

4.3 The additional information captured by Co-occurrence Probability Feature

We believe that the co-occurrence probability feature captures information a large chunk of which is not captured by either topological metrics such as aKatz or content-based metrics such as semantic similarity. To test this conjecture, we examined the number of correct predictions that were made by the co-occurrence probability feature that were not made by either aKatz or semantic similarity. The results are shown in Table 2. As can be observed, in all three of the datasets there exists a significant percentage - up to 75% on the Genetics dataset - of correct predictions in the top 500 that are captured only by the co-occurrence probability feature and not by the other features. This confirms our hypothesis that the co-occurrence probability feature uses information about the domain that is not captured by other types of features.

4.4 Results on Link Prediction as Classification

We report the classification results when we vary the features used for classification. First, we report the results when the three features – co-occurrence probability, aKatz and semantic similarity – are used in isolation. Then we examine the results when we use all three features. Unless otherwise specified, we use the length threshold of 4 for the aKatz feature and 6 for central neighborhood set size of the local probabilistic model-based co-occurrence probability feature.

4.4.1 Baseline Approaches

For the sake of comparison, the results on two baseline approaches – the *Adamic-Adar* measure [1] and *Preferential Attachment* measure [13] are also presented.

The Adamic-Adar measure was originally meant for computing the similarity of two homepages, but has been adapted for computing the similarity between two nodes in a graph by [12]. Let $\Gamma(x)$ be the set of all neighbors of node x. Then the similarity between two nodes x, y is given by

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

The intuition behind the score is that instead of simply counting the number of neighbors shared by two nodes, we should weight the hub nodes less and rarer nodes more.

Preferential Attachment is a measure based on a generative model for graphs that has been well received [3]. Based on the generative model that specifies that new nodes are more likely to form edges with nodes that have a large number of neighbors, Newman [13] has proposed the score for two nodes x, y as $score(x,y) = |\Gamma(x)| \,.\, |\Gamma(y)|$.

4.4.2 Evaluation Metrics

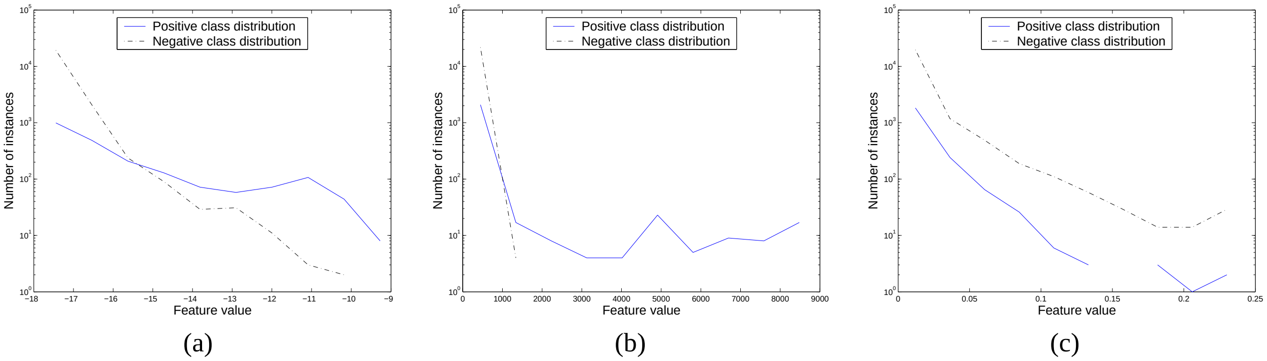
Previous literature has mainly used precision of top-K predictions (K is a user specified parameter - usually the number of true links formed in the testing period) as a metric for evaluation. While this metric has its merits, it has some problems too. Some link prediction methods have relatively high precisions for their top-K predictions when K is small, because they are good at predicting the "easy" links, but the precision drops off dramatically as one increases K. It seems desirable to have an additional evaluation metric that can measure the precision of the classifier without reference to any arbitrary cut-off point. For this reason, we also use AUC (Area Under the ROC Curve) to compare different classifiers, which is a metric that does not need the specification of arbitrary cut-off points and is widely used to evaluate rankings output by a classifier. Huang [8] has previously used this as an evaluation metric in the context of link prediction. An AUC score of 1.0 represents a perfect classifier, and a score of 0.5 is a random classifier. Visually, the closer the ROC curve is to the top left corner of the graph, the better the classifier.

4.4.3 Discussion

Table 3 presents the AUC scores and precision of top-K predictions for different features on different datasets. Following [12], the K for the precision metric has been chosen to be the number of true links in the testing period. We also plot the ROC Curves for the different features and the ensemble method considering all the three features on the three datasets in Figures 7.

The main point to note is that the co-occurrence probability feature consistently outperforms all other features on AUC scores and performs comparably or better on precision. We discuss the results for each dataset in detail below.

One can see that on the DBLP dataset, the co-occurrence probability feature yields the best AUC score (0.8229) when the three features are used in isolation. The aKatz feature



(a) (b) (c)

Figure 6. Class conditional distributions for different features on the Genetics dataset: (a)Co-occurrence Probability (b) aKatz (c) Semantic Similarity

	# correct predictions in c_{500}	# correct predictions in $c_{500} - k_{500} - s_{500}$ (percentage)
DBLP	323	125 (38.7)
Genetics	425	321 (75.5)
Biochemistry	474	246 (51.9)

Table 2. The number of predictions made by the Co-occurrence probability feature alone that were not made by either aKatz or semantic similarity. c_{500}, k_{500} and s_{500} refer respectively to the sets to top 500 predicted links using co-occurrence probability, the set of top 500 predicted links using aKatz and the set of top 500 predicted links using semantic similarity.

Feature	DBLP		Genetics		Biochemistry	
	AUC	Precision (top-K, K=1500)	AUC	Precision (top-K, K=2912)	AUC	Precision (top-K, K=3991)
co-occur. prob. (c)	0.8229	45.80	**0.7904**	46.77	0.8331	52.59
aKatz(k)	0.7501	54.67	0.5888	32.21	0.7644	54.95
semantic(s)	0.7148	35.93	0.5738	16.79	0.6732	20.90
k+c	0.8665	56.06	**0.7904**	46.77	0.8526	**56.38**
c+k+s	**0.8722**	**57.66**	0.7886	**47.08**	**0.8528**	56.32
adamic-adar	0.6148	31.26	0.4864	15.79	0.5384	18.41
preferential attachment (p)	0.7482	36.67	0.7194	35.03	0.8200	51.12
p+k+s	0.8387	52.53	0.7332	37.36	0.8359	54.92

Table 3. Link prediction classification results. The K for calculating precision for each dataset is the number of positive instances (i.e. true links) in the test dataset.

Neighborhood Size	Distance 2		Distance 3		Distance 4	
	AUC	Precision(for top 173)	AUC	Precision(for top 247)	AUC	Precision (for top 382)
2	0.8181	68.78	0.9361	81.78	0.8877	60.20
4	0.9932	98.26	0.9855	93.11	0.8854	59.94
6	0.9373	89.01	0.9943	90.28	0.9806	84.03
8	0.9621	92.48	0.9942	89.06	0.9819	84.29

Table 4. Classification results when varying central neighborhood set size on the Genetics dataset

yields the second best AUC score 0.7501 and it has the best precision for K=1500 (however the precision for this feature drops unexpectedly at some point after K=1500, leading to the lower AUC score). The semantic feature is inferior to the previous two features, yielding 0.7148 AUC score and 35.93% precision. We note that all three features outperform the Adamic-Adar measure significantly in terms of both the AUC score and precision. Preferential Attachment outperforms the semantic feature but is worse than the other two features. Furthermore, when we combine the aKatz and the co-occurrence probability features, we improve the AUC score to 0.8665 as well as the precision to 56.06%. We get the best results when we use all three features together – 0.8722 AUC score and 57.66% precision.

The results are even better on the Genetics dataset, where again the co-occurrence probability feature performs significantly better than the other features. This feature alone can give 0.7904 AUC score and 46.77% precision. The other two features do not perform very well on this dataset when used in isolation, with AUCs dropping to around 0.58 and precisions at 32.21% and 16.79%. When we combine all the three features together, there is not much improvement in the AUC over the co-occurrence probability alone, but there is a slight improvement in the precision to 47.08%. This is because the co-occurrence probability feature has been able to predict a majority of the links that were correctly predicted by the other two features, and predict additional links, leading to not much improvement when using the three features together. Among the baseline methods, Preferential Attachment performs better than aKatz, semantic as well as Adamic-Adar, giving an AUC of 0.71 and a precision of 35.03%.

Coming to the Biochemistry dataset, we again observe the same trend of co-occurrence probability being the most useful feature with an AUC of 0.83 followed by Preferential Attachment with an AUC of 0.82. Precision-wise, aKatz has a slight edge over co-occurrence probability and Preferential Attachment, with aKatz slightly better at 54.9% whereas the latter two have scores of 52.6% and 51.12%. Combining aKatz and co-occurrence probability improves the AUC and the precision to 0.8526 and 56.4%, with additionally combining the semantic similarity giving essentially no improvements. The reason the performance of Preferential Attachment is better on this dataset than Genetics is that the latter is a sparser dataset (it was prepared from 14 journals), which meant that it gave high scores to pairs of prolific authors even though they happened to be in different sub-fields, whereas that would happen less on the Biochemistry dataset.

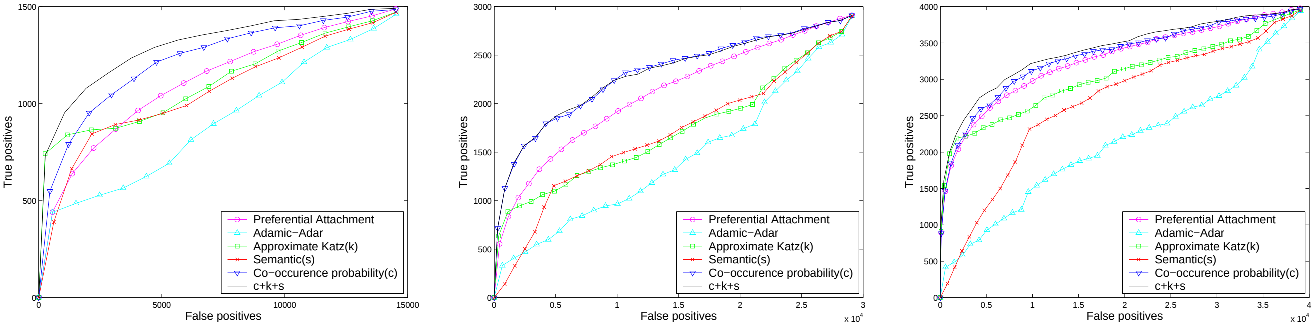
4.5 Results on Varying Central Neighborhood Size

In this section we report the results on varying central neighborhood size for local probabilistic models. We use the Genetics dataset as an example for this set of experiments. The results on other two datasets are consistent. To better validate the use of contextual information for co-occurrence probability estimation, we divide the positive examples (true links) into different classes by their shortest distance. We first examine the case where true links are formed between nodes with shortest distance 2, followed by shortest distance 3 and so on. For each class of links, we generate correspondingly negative examples using the same ratio (1 to 10). We train a separate classifier for each class. We examine the true links up to distance 4.

Specifically on this dataset, we have 173 true links within testing period that are of distance 2, 247 true links of distance 3 and 382 true links of distance 4. Table 4 presents the classification results when we use co-occurrence probability feature alone. We vary the size threshold of the central neighborhood set. The larger threshold is, we tend to use more contextual information when estimating the joint probability for a link. Note that threshold 2 is essentially the independence model. From the results, one can see that overall the local probabilistic model-based approach outperforms the independence model by taking into consideration contextual information of pairs of nodes. For the links of distance 2, the AUC score can be improved from 0.8181 to 0.9621 and we can identify up to 51 more true links. For the links of distance 3, the AUC score can be improved from 0.9361 to 0.9943 and we can identify up to 28 more true links. Finally for the links of distance 4, the AUC score can be improved from 0.8877 to 0.9819 and we can identify up to 92 more true links. We see that the contextual information does indeed improve predictions.

4.6 Results on Timing Performance

Now we report the timing performance on co-occurrence probability feature induction when we vary the size of the central neighborhood set. We use the DBLP dataset as an example for this set of experiments. The results on the other two datasets are similar. Table 5 presents the average time used to induce the co-occurrence probability feature for one link. As one can see, when we increase the size of the central neighborhood set, it takes more time to compute co-occurrence probability feature. This is expected since the cost of learning a local model is higher as we increase the size of the central neighborhood set. Overall, one can see that inducing the co-occurrence probability feature is computationally efficient.



(a) (b) (c)

Figure 7. ROC Curve Comparison on the (a)DBLP (b)Biochemistry (c)Genetics dataset

Size Threshold	Time(ms)
2	4.2
4	6.3
6	8.6
8	10.6

Table 5. Timing results on co-occurrence probability feature induction

5 Conclusions

In this paper, we have presented a simple yet effective approach of leveraging local probabilistic models for link prediction. Specifically, we use topological structure of the network to identify the central neighborhood set of two nodes, and then learn a local MRF model constrained on non-derivable frequent itemsets from this local neighborhood. We then infer the co-occurrence (link) probability from the resulting model and feed it as a feature into a supervised learning algorithm. We have shown that this co-occurrence feature is quite effective for link prediction on real data. When used in combination with other two types of features – topological and semantic features, we find that the resulting classification performance improves. As future work, we would like to examine and test on additional datasets from other domains. Also, we would like to investigate the use of temporal evolution information of these features in link prediction.

References

[1] L. A. Adamic and E. Adar. Friends and neighbors on the web. *Social Networks*, 25(3):211–230, 2003.

[2] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, pages 487–499, 1994.

[3] A.-L. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.

[4] T. Calders and B. Goethals. Depth-first non-derivable itemset mining. In *Proceedings of the SIAM 2005 International Conference on Data Mining*, 2005.

[5] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, pages 1–12, 2000.

[6] M. A. Hasan, V. Chaoji, S. Salem, and M. Zaki. Link prediction using supervised learning. *Workshop on Link Analysis, Counter-terrorism and Security (at SIAM Data Mining Conference)*, 2006.

[7] Z. Huang. Link prediction based on graph topology: The predictive value of generalized clustering coefficient. In *Workshop on Link Analysis: Dynamics and Static of Large Networks (LinkKDD2006)*, 2006.

[8] 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*, pages 141–142, New York, NY, USA, 2005. ACM Press.

[9] F. Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, Cambridge, MA, 1998.

[10] H. Kashima and N. Abe. A parameterized probabilistic model of network evolution for supervised link prediction. In *ICDM*, pages 340–349, 2006.

[11] S. Lauritzen and D. Speigelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society, Series B (Methodological)*, 50(2):157224, 1988.

[12] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, pages 556–559, New York, NY, USA, 2003. ACM Press.

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

[14] J. O'Madadhain, J. Hutchins, and P. Smyth. Prediction and ranking algorithms for event-based network data. *SIGKDD Explor. Newsl.*, 7(2):23–30, 2005.

[15] D. Pavlov, H. Mannila, and P. Smyth. Beyond independence: probabilistic models for query approximation on binary transaction data. *IEEE Transactions on Knowledge and Data Engineering*, 15(6):1409–1421, November 2003.

[16] A. Popescul and L. H. Ungar. Statistical relational learning for link prediction. In *IJCAI03 Workshop on Learning Statistical Models from Relational Data*, 2003.

[17] M. J. Rattigan and D. Jensen. The case for anomalous link discovery. *SIGKDD Explor. Newsl.*, 7(2):41–47, 2005.

[18] B. Taskar, M. F. Wong, P. Abbeel, and D. Koller. Link prediction in relational data. In *Advances in Neural Information Processing Systems 16 [Neural Information Processing Systems*, 2003.

[19] C. Wang and S. Parthasarathy. Summarizing itemset patterns using probabilistic models. In *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 730–735, New York, NY, USA, 2006. ACM Press.

[20] M. J. Zaki, S. Parthasarathy, and W. L. Mitsunori Ogihara. New algorithms for fast discovery of association rules. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, pages 283–286, 1997.