A Matrix Alignment Approach for Link Prediction

Jerry Scripps, Pang-Ning Tan, Feilong Chen, and Abdol-Hossein Esfahanian
Computer Science and Engineering
Michigan State University, E. Lansing, MI 48824
$\{scripps,ptan,chenfeil,esfahani\}$@msu.edu

Abstract

This paper introduces a new discriminative learning technique for link prediction based on the matrix alignment approach. Our algorithm automatically determines the most predictive features of the link structure by aligning the adjacency matrix of a network with weighted similarity matrices computed from node attributes and neighborhood topological features. Experimental results on a variety of network data have demonstrated the effectiveness of this approach.

1 Introduction

Link prediction is a technique used to predict the formation of ties within a network. It can be used to recommend new relationships such as friends in a social network or to uncover previously unknown links such as regulatory interactions among genes. More generally, the link prediction problem can be formulated as a binary classification problem [4, 5]—given a node pair, we seek to accurately predict whether there is an edge between them based on their node attributes, neighborhood structure, or other properties of the network topology. Such a problem can be approached in two ways: (1) using a generative approach, where the focus is on learning a model of the joint probability density of the nodes, links, subgroups, etc., and then make a prediction by using Bayes rule [2, 6, 8], or (2) using a discriminative approach, which directly learns a target function that will map an input node pair to its class [3, 4, 5]. In this paper we introduce a new discriminative learning technique for link prediction based on the matrix alignment approach.

We assume that the node attributes contain information needed to make the prediction but that the links would help us to prioritize the attributes. For example, in social networks, people may become linked (friends, relatives, coworkers, accomplices, etc.) because they
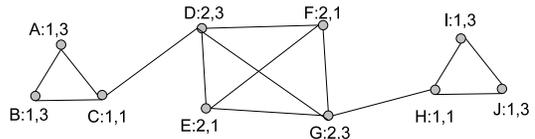

Figure 1. Clustering a small network

have shared some common characteristics or interest. While many attributes about a person may be known (e.g., eye color, height, books they like to read, school they attend, etc), it is a small set of them that are important when befriending. The challenge is to determine which subset of attributes are important to establish the links observed in a network. A link between two persons may also be determined by examining their existing ties (e.g. do they have common friends or are they popular figures?) It is not our purpose here to debate the merits of using attributes versus neighborhood topological features [5]. Indeed, what may be appropriate for one data set may not be for another. The objective of this paper is to present a flexible framework that allows us to identify the relevant attributes or topological features that are most well-aligned with the link structure.

To further illustrate the motivation behind our approach, consider the network shown in Figure 1. The figure shows a network of ten nodes, their identifiers and attribute values (e.g. node A has the attributes 1 and 3). The question we would like to answer is, would it be more likely that node C would link to E or J? Using just the attributes it would appear that they would be equally likely since C has exactly one attribute value in common with both of the others. Using the topological features, it is probably more likely that C would link to E since they have a shorter path length, more common neighbors, etc. However, by examining the network we can tell that nodes that are linked are more likely to have identical first attributes than second. So we should assign a higher likelihood to C linking to J, than to E.

For the network in Figure 1 attributes are more predictive than the topological features but in other networks it could be otherwise. Our proposed approach will automatically determine the most predictive attributes and topological features by aligning the adjacency matrix with weighted similarity matrices computed from the attributes and topological features. The weights of the similarity matrices are determined by solving a system of linear equations.

2 Methodology

Consider a physical or social network represented as a graph $G = (V, E)$, where $V = \{1, 2, ..., |V|\}$ is the set of nodes and $E \subseteq V \times V$ is the set of edges. Let $A = (a_{ij})_{n \times n}$ denote an adjacency matrix representation of the graph, where $a_{ij} = 1$ if there is a link between nodes i and j and zero otherwise. Also let $X = (x_{ik})_{n \times d}$ denote its corresponding data matrix, where x_{ik} is the kth attribute value for node i. Assume that the attributes in X have been properly normalized or standardized so that we may represent the attribute similarity between nodes using the matrix product XX^T. For example, if X is a binary-valued matrix and its rows are normalized to have unit length, the matrix product XX^T would correspond to the cosine similarity between nodes.

2.1 Matrix Alignment

In an ideal network, one can imagine perfect alignment between the links and the attributes – that is where $\forall i, j : sim(x_i, x_j) = a_{ij}$. However, in most networks, such perfect alignment will not exist. The proposed matrix alignment framework uses a set of weights to determine the important attributes for establishing links between nodes. More specifically, our goal is to learn a set of weights $\vec{w} = \{w_1, ... w_d\}$ that minimizes the objective function $L = \|A - XWX^T\|_F^2$, where the diagonal elements of W correspond to $\vec{w}$. Intuitively, the objective function aims to learn a set of weights that maximizes the degree of alignment between the link structure and attribute similarity.

To avoid overfitting, a regularization technique can be employed by adding a penalty term $\lambda \|W - I\|_F^2$ to the objective function, where I is the identity matrix:

$$L = \|A - XWX^T\|_F^2 + \lambda \|W - I\|_F^2$$

This will coerce the weight vector $\vec{w}$ to ones for high values of λ, which is equivalent to assigning equal importance to all the attributes. To solve for the weights the first step is to take the partial derivatives with respect to w_m and set them to zero:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} \cdot x_{im} x_{jm} + \lambda =$$
$$\sum_{k=1}^{d} \left(\sum_{i=1}^{n} \sum_{j=1}^{n} x_{ik} x_{jk} \cdot x_{im} x_{jm} \right) w_k + \lambda w_m$$

We can then arrange these d equations into a system of linear equations, $Z\vec{w} = b$, by letting
$b_m = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} \cdot x_{im} x_{jm} + \lambda$
and
$$Z_{mk} = \begin{cases} \sum_{i,j}^{n} x_{ik} x_{jk} \cdot x_{im} x_{jm} & \text{, if } m \neq k \\ \sum_{i,j}^{n} x_{ik} x_{jk} \cdot x_{im} x_{jm} + \lambda & \text{, if } m = k \end{cases}$$
The weight vector $\vec{w}$ can be computed using Gaussian elimination. Once the vector $\vec{w}$ has been learned the expression $x_i W x_j^T$ provides a relative measure of the likelihood of nodes, i and j forming a link.

2.2 Incorporating Topological Features

Since most link prediction algorithms make use of knowledge of the neighborhood structure, we show how topological features can be incorporated into our framework. Each topological metric for a node-pair (path length, number of common neighbors, etc.) can be treated as an attribute and assigned a weight. Let d_1 be the number of node attributes and d_2 be the number of topological features. So each $Y^{(i)}, 1 < i < d_2$ is an $n \times n$ matrix containing the metric values for each pair of nodes. The objective function for our matrix alignment framework becomes:

$$L = \sum_{i=1}^{n} \sum_{j=1}^{n} \left(a_{ij} - \sum_{k=1}^{d_1} x_{ik} x_{jk} u_k - \sum_{k=1}^{d_2} y_{ij}^{(k)} v_k \right)^2$$

We will solve for the weights $\vec{w} = (\vec{u}, \vec{v})$ as before by taking the partial derivatives $\frac{\partial L}{\partial u_m}$ and $\frac{\partial L}{\partial v_m}$, and setting them to zero. The resulting equations can be re-arranged as a system of linear equations, $Z\vec{w} = b$, by letting

$$b_m = \begin{cases} \sum_{i,j}^{n} a_{ij} \cdot x_{im} x_{jm} & m \leq d_1 \\ \sum_{i,j}^{n} a_{ij} \cdot y_{ij}^{(m)} & m > d_1 \end{cases}$$

$$Z_{mk} = \begin{cases} \sum_{i,j}^{n} x_{ik} x_{jk} \cdot x_{im} x_{jm} & m \leq d_1, k \leq d_1 \\ \sum_{i,j}^{n} y_{ij}^{(k)} \cdot x_{im} x_{jm} & m \leq d_1, k > d_1 \\ \sum_{i,j}^{n} x_{ik} x_{jk} \cdot y_{ij}^{(m-d_1)} & m > d_1, k \leq d_1 \\ \sum_{i,j}^{n} y_{ij}^{(k-d_1)} \cdot y_{ij}^{(m-d_1)} & m > d_1, k > d_1 \end{cases}$$

Again we use Gaussian elimination to find $\vec{w}$. After learning the weights, $\sum_{k=1}^{d_1} x_{ik} x_{jk} u_k + \sum_{k=1}^{d_2} y_{ij}^{(k)} v_k$ provides a relative measure of the likelihood of i and j forming a link according to the similarity of their attribute values and topological features.

2.3 The Matrix Alignment Algorithm

The matrix alignment algorithm presented in Algorithm 1, combines all of the methods described above to predict missing links. It is patterned after the EM algorithm to iteratively calculate the weights and assign values to missing links. The adjacency matrix A is assumed to have missing link values. As the algorithm progresses, it alternately calculates new weights (by using Gaussian elimination) and then predicts the values of the missing links based on the new weights. Predictions are made using a pair of quadratic discriminators [1] g_0 and g_1. g_0 is trained on the weighted similarity scores for each pair of nodes in the training set that are not linked, and g_1 is trained on the weighted scores for linked training set pairs. Each pair in the test set is predicted to be a link/non-link depending on which discriminator g_1/g_0 is higher using the score calculated for the pair.

Input: adjacency matrix A, data matrix X
Output: adjacency matrix A, weights W
repeat
    $W \leftarrow calc\_weights(A, X);$
    // missing link assignment
    foreach *missing link* $a_{ij} \in A$ **do**
        $a_{ij} \leftarrow assignLink(i, j, A, X, W);$
    end
until *W converge* ;
return *W,A*;

Algorithm 1: Learning weights and predicting links

---

3 Experimental Evaluation

3.1 Experimental Setup

The data sets for our experiments were taken from the DBLP bibliographic database[1], the TakingIt-Global.org social networking website, and the WebKB data set[2]. Table 1 summarizes the properties of these data sets. For DBLP, the nodes correspond to authors who have published in conferences on database, artificial intelligence, network and software engineering from 1998 to 2003. The nodes were linked based on the co-authorship relationship. Node attributes correspond to selected words in the title of papers published by the authors. TakingGlobal.org (TIG) is a social networking website for young people to become involved

---

[1] http://dblp.uni-trier.de/
[2] http://www.cs.cmu.edu/ WebKB/

in activities and share their ideas about global issues like poverty, social justice and health. Members (our nodes) establish friendship links (links) and select the activities and events of interest (attributes). The members were grouped by region. The WebKb data set is a collection of web pages collected from four university web sites in January of 1997 linked to each other by hyper-links. Node attributes correspond to significant words chosen from the web pages.

Table 1. Data Sets

Data set	nodes	links	attributes
DBLP	10709	22315	580
0: database	3445	8547	542
1: art. intel.	3492	5797	556
2: networks	2855	5586	524
3: soft. eng.	1238	2042	410
TakingItGlobal.org	5852	29776	123
0: Africa	1128	2387	123
1: Asia	855	1778	123
2: Europe	368	593	123
3: North America	1258	4221	123
4: South America	334	1095	123
5: Middle East	304	674	123
Webkb			
0: Cornell	195	304	1703
1: Texas	187	328	1703
2: Washington	230	446	1703
3: Wisconsin	265	530	1703

For evaluation purposes, 10% of the linked pairs and an equal number of non-linked pairs were randomly selected as the test set. The experiments report the accuracy of prediction, averaged over 10 trials. The regularizer was set to 1 throughout our experiments (except where noted below).

3.2 Experimental Results

Figure 2 summarizes the results of predicting the missing links in the test set using the weights estimated by our matrix alignment algorithm. For all data sets except for DBLP using weights was more accurate for predicting links than the unweighted measure (XX^T). From Section 1 recall that our motivation is that not all attributes are equally helpful when predicting links. When looking at the TIG data we saw that some attributes such as *gender* were low weighted but others such as *events* and *member groups* were highly weighted. Apparently members befriend others with whom they have participated in an event or a group together. For DBLP, the weights were not that helpful for

prediction. In this case increasing the regularizer improves the prediction as can be seen in Figure 3. In these bibliographic data, the unique words that appear in a particular test publication title may not appear in the training titles. So in effect the weights are overfitted to the training data. Increasing the regularizer actually improves all four DBLP data sets. Cross-validation can be used to determine the best regularizer value.
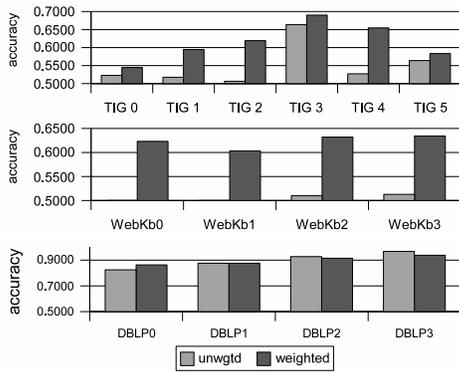

Figure 2. Unweighted vs. Weighted

According to the study by Liben-Nowell and Kleinberg [5], *common neighbors* is often one of the most effective topological metrics for link prediction. Thus we chose it to be added to our framework in order to improve the predictions. The tests compare a linear combination of the unweighted attributes plus *common neighbors* to the weighted combination.
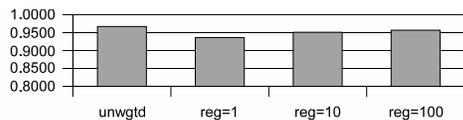

Figure 3. Effect of regularizer on DBLP3

Figure 4 shows the results of the tests. It can be seen that the effect of common neighbors on the TIG and WebKb sets is negligible while for DBLP it is helpful. The DBLP set has many cliques – groups of authors who always publish together and not with anyone else – which is a perfect setting for using common neighbors. The weighted predictions in these tests were all better than the unweighted predictions. Furthermore, the weight of the topology feature was higher for data sets where common neighbors was predictive and lower where it was not helpful.

4 Conclusions and Future Work

We presented a discriminative approach to predicting links that aligns attributes and link metrics to the
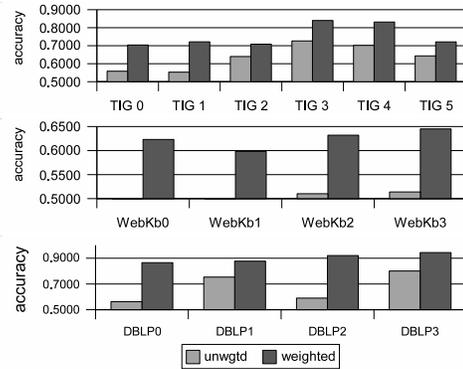

Figure 4. Using Topology

link structure. The two general approaches to link prediction so far have been either generative or discriminative. While each approach has its advantages and disadvantages, it has been suggested [7] in general "that discriminative classifiers are almost always to be preferred to generative ones". Our approach has the advantage of being flexible, allowing many extensions to the framework. The extensions that were presented in this paper were to incorporate topological data and to make use of regularization to avoid overfitting. This framework can also be extended in other ways, e.g., by using nonlinear kernels as the similarity function.

References

[1] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. John Wiley & Sons, New York, 2000.

[2] L. Getoor and C. P. Diehl. Link mining: a survey. *SIGKDD Explorations*, 2005.

[3] M. Al Hasan, V. Chaoji, S. Salem, and M. Zaki. Link prediction using supervised learning. *SDM'06: Workshop on Link Analysis, Counter-terrorism and Security*, 2006.

[4] H. Kashima and N. Abe. A parameterized probabilistic model of network evolution for supervised link prediction. *ICDM*, 2006.

[5] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. *CIKM*, 2003.

[6] J. Neville and D. Jensen. Leveraging relational autocorrelation with latent group models. *ICDM*, 2005.

[7] A. Ng and M. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in Neural Information Processing Systems*, 2002.

[8] B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. *UAI02*, 2002.