On the Beaten Path: Exploitation of Entities Interactions For Predicting Potential Link

Young-Woo Seo and Katia Sycara

CMU-RI-TR-06-36

August 2006

Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Abstract

We propose a new non-parametric link analysis algorithm that predicts a potential link between entities given a set of different relational patterns. The proposed method first represents different types of relations among entities by constructing the corresponding number of factorized matrices from the original entity-by-relation matrices. The prediction of a possible link between entities is done by linearly summing the weighted distances in the latent spaces. A logistic regression is used to estimate regression coefficients of distances in the latent spaces. From the experimental comparisons with various algorithms, our algorithm performs best in precision and second-best in recall measure.

Contents

1 Introduction

Many real world data sets can be represented as collections of linked objects. For example, a collection of scientific papers can be represented as a graphical structure in which papers (i.e., vertices) are connected by citations (i.e., edges) or researchers are connected to one another by authoring papers together. Similarly organizations can be represented as a directed graph where people are connected by organizational structure, social relationships, communications patterns, or collaborations.

One of the main purposes of social network analysis, link analysis, and relational data mining is to understand the relations between entities and the implications of these relationships. In particular, given a particular context such as research collaboration, the task will be to answer questions like "Who will be my next co-author? Whom will I work with in the near future?"

However traditional statistical inference methods that assume independence among instances will not be appropriate to the task of the analysis of such linked data. Instead of treating examples independently the linkage information should be exploited to derive potential correlations [5].

A fair amount of research has been done at various contexts, hypertext-analysis for web document classification [2], link discovery for counter-terrorism [9], frequent item discovery for "market basket problem" [3], predicting collaborators in the future [7], identifying cohesive groups [8].

In this paper we propose a new non parametric link analysis algorithm that predicts a potential link between entities given a set of different relational patterns. The proposed algorithm represents different types of relations among people by constructing a corresponding number of factorized matrices from the original entity-by-relation matrices. The prediction of a possible link between entities is done by linearly summing the weighted distances in the latent spaces. Logistic regression is used to estimate regression coefficients of distances in the latent spaces.

2 Related Work

Statistical approaches to graph analysis typically assume that there are a set of n actors (i.e., entities) and their binary relationships. The data are usually represented in the form of an $n \times n$ adjacency matrix X where cell $x_{ij} = 1$ indicates the presence (or absence) of some form of (directed) relationship between entities i and j [10]. The most common approach assumes that the edge, x_{ij}, is drawn from an underlying distribution for a set of binary random variables, saying that observed relations are noisy as to whether a link actually exists or not. Then the goal of statistical modeling is to infer a probable model for $P(\theta|X)$ that requires a number of parameters to explain the pattern of observed relations.

cGraph Kubica and his colleagues proposed a parametric link analysis algorithm called cGraph [7] that approximates the underlying graph structure by considering different types of relations between entities. Maximum likelihood estimation (MLE) is used to learn the underlying graphical structure for the given linked data, assuming that there is an unknown (generative) model that generates the pairwise relationships

between entities. In particular, they proposed two graph generation models: the random walk model and the random tree model. These models are about how an entity is connected to an existing graph, considering the conditional probability of a new entity given a set of entities. However the link ordering (i.e., which entity is regarded earlier than the other) might cause a computational problem because the ordering of links is not known a priori. By treating each possible link ordering as equally probable, the cGraph algorithm can approximate the weight of an edge from A to B as a percentage of possible link orderings in which A_1 and B_2. In particular, instead of assuming Boolean edges that have been widely used in social network analysis and indicate the presence (or absence) of a relationship [10], [1], cGraph estimates the weight of an edge, W_{AB}, by counting co-occurrences between two entities while differentiating among the type of interactions (e.g., co-authoring papers, adviser-advisee, co-interests on research topic, etc.).

$$W_{AB}(t) = \frac{\sum_{L:(A,B)\subset L} \left(\frac{U(L.type)T(L.time,t)}{|L|-1} \right)}{\sum_{L:A\subset L} U(L.type)T(L.time,t)}$$

where $|L|$ is the size of the link, $U(L.type)$ is the typical weighting of a link type, $L.type$, and $T(L.time, t)$ is the temporal weighting of a link at time t where the link occurred at time $L.time$. The underlying idea of having a temporal weighting function is that the recent links are more indicative of the current graph than links that occurred some time ago. An exponential decay function, $T(L, t)$ is used to realize this idea.

$$T(L,t) = \begin{cases} e^{-\beta(t-L.time)} & t \geq L.time \\ 0 & t < L.time \end{cases}$$

With the estimated parameters, such as the link type weighting and the temporal decay parameter, about a given data set using the above equations and cross validation, a proximity measure is defined to identify a set of closely related entities:

$$prox(A,B) = \sum_{v \in V_{A,B,D}} \left(\prod_{e_i \in v} P(e_i | A_j \forall j < i) \right)$$

where $V_{A,B,D}$ is the set of all nonself-intersecting paths of length less than or equal to D from A to B and $e_i \in v$ is a directed edge in v.

Empirical Bayes Screening The Empirical Bayes Screening (EBS) algorithm is a statistical method that ranks links between entities according to their "interestingness" [3]. The unique feature of this algorithm is its preference for those links which cannot be detected by pairwise independence. For example, the link between two entities is estimated based on the assumption that entity i and j have occurred independently: $e_{ij} = N \times P_i \times P_j$, where $P_i = n_i/N$, n_i is the number of occurrences of entity i, and N is the number of examples in the dataset. Generally the link of k entities is estimated by the loglinear models. Specifically, we construct all 2-way marginal tables (i.e., contingency tables) and use an iterative proportional fitting algorithm to obtain the loglinear estimate e_{ijk}. In particular, EBS assumes that the links among k entities are drawn from a hierarchical distribution. That is, the links among k entities form

poisson distributions with parameters, $\lambda_{ijk}e_{ijk}$, where λs in turn come from a prior $\pi(\lambda|\theta)$ distribution

$$\begin{aligned} P(n_{ijk}|\lambda) &\sim Poisson(\lambda_{ijk}e_{ijk}) \\ \lambda_{ijk} &\sim Gamma(\alpha, \beta) \end{aligned}$$

The parameter $\theta = \{\alpha, \beta\}$ is estimated from the data using Empirical Bayes. The product of unconditional distribution of n_{ijk} is then maximized to obtain the maximum likelihood estimates for θ

$$\begin{aligned} f(n) &= \int Poisson(n|\lambda e)\pi(\lambda|\hat{\theta})d\lambda \\ \hat{\theta} &= \arg\max_{\theta} \prod_{\forall n} f(n) \end{aligned}$$

To predict a link given data, the number of occurrences of each possible link is computed by using the training set. In other words, for each (n, e), the algorithm use the Bayes rule to compute the estimate of true occurrence based on independence assumption $\lambda \sim \frac{Poisson(n|\lambda e)\pi(\lambda|\hat{\theta})}{f(n)}$ and then calculate the geometric mean $\Lambda = \frac{\exp(\Psi(\hat{\alpha}+n_i))}{\hat{\beta}+e_i}$ for ranking all possible links.

3 Analysis of the Beaten Path for Finding Interactional Patterns

People interact with others in various contexts. In a particular community these interactions can often be represented in clear patterns, such as co-publication, co-involvement with a project, sharing research interests, or a special academic relationship (e.g., adviser/advisee). A matrix is used to represent each interactional pattern in a matrix, X, where each row i corresponds to an entity and each column j represents a (relational) link. An element in the matrix is the value of whether entity i participated in the link j. The actual value of the element can be either Boolean (i.e., absence or presence of interaction) or real value (i.e., strength of link). The real-valued matrix is more realistic representation in that it provide the strength of a link instead of indicating absence or presence of relation.

Our hypothesis about the relation link is that a link abstracting multiple types of relations would be more informative than one represented by a type of relation. cGraph [7] is one of the algorithms that implements this idea. In particular, when cGraph estimate the weight of an edge between two entities by counting co-occurrences, it considers the different types of relations at that edge. This approach is similar to the idea that exponential graph model from social network analysis assumes [1]. In particular, the conditional probability of the observed data x given θ, $P(X = x|\theta)$ is estimated by a log linear model:

$$P(\mathbf{X} = \mathbf{x}|\theta) = \frac{\exp\{\theta^T z(\mathbf{x})\}}{k(\theta)} = \frac{\exp\{\theta_1 z_1(\mathbf{x}) + \ldots + \theta_r z_r(\mathbf{x})\}}{k(\theta)}$$

where $z_k(\mathbf{x})$ is a function of any graph-theoretic characteristics of the relation (e.g., gender, age, race, co-authoring a paper, etc.), θ_k is a coefficient, and $k(\theta)$ is a normalizing constant ensuring that the probabilities sum to unity.

The idea for implementing our hypothesis is to construct multiple interaction matrices in which we can calculate the distance between entities from different perspectives and then to estimate how each of the calculated distances contribute to the final decision on their relational tie – how strongly two entities are related to each other.

Suppose there are $|R|$ relations of interest and the same number of entity-by-relation matrices. An entity-by-relation matrix is projected onto a latent space in which their original relationships are represented clearly and accordingly the distance (or similarity) among entities is calculated relatively easy. Since a distance between two entities in a latent space represents the weakness (or strength) of their relational tie in terms of a particular relation that the latent space captures, the weakness of the relational tie between those entities can then be calculated by summing all those distances. However this only works if all the entities have roughly the same amount of interactions under all the relations of interest. Since in reality some entities only have a relatively small number of interactions relative to those of others, it is necessary to figure out the corresponding number of coefficients for relation variables. This is a common regression problem that can be solved by any regression estimator.

Therefore the probability of potential link between two entities X^l and X^{l+1}, ($P(Y|X^l, X^{l+1})$) can be estimated by linearly summing the weighted distances in the latent spaces.

$$\begin{aligned} P(Y|X^l, X^{l+1}) &= reg(\theta^T \mathbf{z}(X^l, X^{l+1})) \\ &= reg\left(\theta_1 z_1(X^l, X^{l+1}))+, ..., +\theta_{|R|} z_{|R|}(X^l, X^{l+1}))\right) \end{aligned} \quad (1)$$

where $reg(\cdot)$ is a regression function, θ^T is a random vector of parameters, $\mathbf{z}(X, X^l)$ is a vector of any distance functions that each of them calculates the distance between two given entities in a latent space, and $|R|$ is the total number of relations of interest.

Let us first detail why we use a latent space (i.e., factorized matrix) instead of the original entity-by-relation space (i.e., Boolean matrix). Each element in an entity-by-relation (e.g., people-by-publication) matrix represents the presence (or absence) of a link between entity and relation. If one pulls out a column (i.e., a publication) from the matrix, he will know how many persons in the row space wrote the paper together. In the same way, each row represents the number of publications by a person. Then the similarity (e.g., inner product between two row vectors) can be calculated by considering how many publications two people co-author. Note we are not concerned about the temporal property of a relation. However since this approach heavily relies on co-occurrence statistics, it will fail to infer the relational tie from the case in which two entities do not have a large overlap on their link patterns, even though they have a strong relational tie. For example, a pair of advisor and advisee is a strong academic relationship, but the pair might have only a publication. To cope with this problem, we project the entity-by-relation matrix X onto a latent space in which two entities with few overlaps in their links that are highly correlated can have a high similarity. In particular, we use singular value decomposition (SVD) to capture the characteristics of the original relationships (e.g., between entities or between relations) by orthogonal

vectors and factor values. SVD decomposes an entity-by-relation matrix, X, into sets of left and right singular vectors. The left singular vectors correspond to canonical sets of entities, which are latent groups, while the right singular vectors correspond to the "mixing proportions" in which these groups should be combined to approximate the observed links. It is a least-square method in that the projection of the matrix X onto the latent space is chose such that the representations in the original space are changed as little as possible when measured by the sum of the squares of the differences:

$$\begin{aligned} \Delta &= ||X - \hat{X}_k||_2 \\ \text{where,} & \\ \hat{X}_k &= U_k \Sigma_k V_k^T \end{aligned}$$

where $X = m \times n, (m > n)$, $\text{rank}(X) = r$, U is a $m \times m$ orthogonal matrix of which columns are the eigenvectors of XX^T and V is a $n \times n$ orthogonal matrix of which columns are the eigenvectors of $X^T X$, and Σ $(m \times n)$ is the diagonal matrix of singular values $\sigma_1 > \sigma_2 > \sigma_3 > ... > \sigma_n$, where $\sigma_i > 0$ for $1 \leq i \leq r$, $\sigma_j = 0$ for $j \geq r + 1$. The singular values are square roots of eigenvectors from either XX^T or $X^T X$. $\hat{X}_k$ is a re-constructed version of the original matrix X by the k singular values of X that is the closest [1] approximation to X. $\hat{X}_k$ captures most of the important underlying structure in the association of entities and relations, yet at the same time removes the noise or variability. Entities that occur in similar relations (e.g., publications), for example, will be near each other in the k-dimensional latent space even if they never co-occur in the same relation. Likewise, some relations (e.g., publications), which do not share any entities, may nonetheless be near it in k-dimensional latent space.

The choice of k is quite important. Ideally, we want a value of k that is large enough to fit all the real structure in the original data, but, at the same time, small enough so that we do not also fit unimportant details. In the experiment, we heuristically determined the value of k that captures a large portion of the variance in the original data.

After projection of several entity-by-relation matrices onto latent spaces [2], the next step is to estimate $\vec{\theta} =< \theta_1, ..., \theta_{|R|} >^T$, which is a random vector of parameters, in equation 1. We choose logisitic regression for this regression task because it is relatively simple to implement, but usually shows a comparable result to more complex ones. The common usage of logistic regression is a binary classifier that requires a dichotomy of $P(Y = 1|X)$ to determine whether X belongs to the positive or negative class. In our case, it is a multi-class problem in which the number of classes $|C|$ is equal to the total number of entities, N. Thus these regression coefficients are a matrix, $\theta = (N - 1) \times |R|$ [3]. The complete form of the idea in equation 1 is as follows:

$$P(Y = y_k|\mathbf{z}(X^l, X^{l+1})) = \frac{\exp(\theta^T \mathbf{z}(X^l, X^{l+1}))}{1 + \sum_{j \in |C|-1} \exp(\theta^T \mathbf{z}(X^l, X^{l+1}))} \quad (2)$$

[1] The term "closest" means $\hat{X}_k$ minimizes the sum of the squares of the difference of the elements of X and $\hat{X}_k$.

[2] There might be some "relation" between two different latent spaces, but in this work we do not consider to model these relations.

[3] For $k < |C|$, $P(Y = y_k|X) = \frac{\exp(\theta^T X)}{1+\exp(\theta^T X)}$. For $k = |C|$, $P(Y = y_{|C|}|X) = \frac{1}{1+\exp(\theta^T X)}$

	true positive	true negative
output positive	a	b
output negative	c	d

Table 1: Per-fold contingency table for a binary prediction.

where $Y \in \{Y_1, ..., Y_{|C|}\}$ and $|C| = N$.

However training $N-1$ logistic regression is computationally expensive, assuming we use a gradient-descent as weight update rule (either online or batch). And also it has another theoretical problem that this approach does not assume independently and identically distributed samples, meaning that the interaction patterns of a person is generated from a completely different probability distribution from another.

We assume instead that the interaction patterns of all people are generated from an identical distribution, but each person's pattern is generated independently. In addition to computational tractability, this assumption makes the training process much easier. In particular, we generate a matrix that is comprised of two parts: presence of relations (i.e., positive examples) and absence of relations (i.e., negative examples). A row of the matrix is one of all possible pairs of entities. In the experiment, we train our logistic regression in this way.

4 Experiments

The objective of the proposed algorithm is to identify cohesive groups by exploiting interactional patterns between entities. In particular, given a particular context such as research collaboration, it will be a task of answering questions like "Who will be my next co-author? Whom will I work with in the near future?" In other words, it is a task of predicting potential links between entities in near future. This task can be further categorized into a prediction of *All* possible collaborators, which identifies which are the k entities with which the entity is most likely to appear in future links, and a prediction of *New* collaborators, which identifies which are the k entities with which the entity has never appeared in a link and is most likely to appear with in future links [7]. In this experiment, we are only concerned about the prediction of *All* possible collaborators given the linked data because our method does not consider a temporal property of linked data.

We used a k-fold (sequential) cross validation on the data set to compare the performance of the algorithms. This approach is reasonable in that we do not know the ground-truth of link data and the usual split of the data set into two parts might have a skewed (or sparse) matrix as either training or test data, which makes the task of link prediction difficult.

For the actual performance measure, often it is not clear to evaluate which algorithm is better than the other by a particular performance measure. In order to evaluate the performance of various algorithms extensively, we first compiled the per-fold contingency table (i.e., Table 1) and then defined three different performance measures by using that table: precision $= a/(a+b)$ if $a+b > 0$ and $a > 0$, otherwise un-

defined; recall $= a/(a+c)$ if $a+b>0$ and $a>0$, otherwise undefined; accuracy $= (a+d)/(a+b+c+d)$ if $a>0$ or $d>0$, otherwise undefined. In particular, the accuracy can tell us how well a method perform on average, but it cannot tell us its usefulness on a particular class (e.g., positive or negative). To this end, we elaborate the accuracy in terms of "recall" and "precision." The recall defines the number of examples correctly predicted as a fraction of all correct examples whereas the precision defines the number of examples correctly predicted as fraction of all the examples predicted by a method. Kubica and his colleagues [7] used only recall which is defined by the the ratio of predicted collaborators to the number of people who actually co-occurred in the test set.

By using table 1, we compute the above measures for each fold of k-fold cross validation. And then in order to measure global performance we used the *micro-average* that is obtained by merging the contingency tables of the k folds (i.e., by summing the corresponding cells), and by using the merged table to produce global performance measures.

4.1 Data Set

We tested the proposed algorithm with the modified Institute data [7]. The *Institute* data is originally comprised of linked entities in three different types (co-publication, common research interest and advisor/advisee) that was compiled from public data on Carnegie Mellon University Robotics Institute. We added another link "project co-involvement" to the original *Institute* data. In particular, there are 456 people connected with each other by over 3,383 links, excluding self-relations. In the modified Institute data, there are four different relation (or interactional) types: co-publication (3,051 links), common research interest (70 links), co-involvement of project (181 links), and advisor/advisee (81 links).

4.2 Experimental Settings

The performance of the proposed algorithm for identifying cohesive groups was tested against a number of other algorithms, such as cGraph and similarity-based methods.

Suppose X is an entity-by-relation matrix in which a row represents an entity (i.e., people) and a column represents an instance of the relation (e.g., a publication, a project). The most basic approach is to treat a row as a multi-dimensional vector and to rank pairs in decreasing order of a similarity function (or in increasing order of a distance function).

For an entity, x, let $\Gamma(x)$ denote the vector of linkage pattern of x in an entity-by-relation matrix.

- *Jaccard's coefficient* The Jaccard coefficient measures the probability that two row vectors have an instance of the relation. The similarity between two row vector x_i and x_j is measured by $link\_score(x_i, x_j) = |\Gamma(x_i) \cap \Gamma(x_j)|/|\Gamma(x_i) \cup \Gamma(x_j)|$
- *Cosine similarity* The cosine similarity measures the cosine angle between two entity vectors, $link\_score(x_i, x_j) = \Gamma(x_i) \cdot \Gamma(x_j)/|\Gamma(x_i)||\Gamma(x_j)|$. The $link\_score(x_i, x_j)$

is 1.0 if two vectors are co-linear whereas the score is 0.0 if they are orthogonal to each other.

- *Preferential attachment* The basic idea of this method is that the probability of any type of relation in future (e.g., co-authoring paper, co-involvement project) is correlated with the product of the number of collaborators of two entitled [8]. The number of collaborators of an entity in a certain relation can be calculated by first constructing a entity-by-entity matrix and by summing the corresponding row (or column).

- *Katz* defines a measure that directly sums over the collection of paths, exponentially damped by length to count short paths more heavily. $score(x_i, x_j) = \sum_{l=1}^{\infty} \beta^l \cdot |\mathrm{paths}_{i,j}^l|$, where $\mathrm{paths}_{i,j}^l$ is the set of all length-l paths from x_i to x_j.

To train a logistic regression classifier, we built a matrix that contains the information from the original four people-by-relation matrices. The rows of the matrix are all possible pairs of given entities. In particular, a row vector is a vector of four different latent distances between two selected entities. [4] Its target value is 1 if there is more than one interaction between those entities. Otherwise it is 0. The size of actual data set is 103,740 $\times$ 4, where there are 103,740 pairs of 456 entities without self-intersecting and 4 different latent space distances used. The final matrix has a very skewed distribution in that there is approximately 5% data that has more than one relation – 6,140 positive out of 103,740 examples.

When projecting the original entity-by-relation matrix, we heuristically determined the actual value of k as the corresponding number of singular values in the singular matrix Σ that cover 80% of the variance. For example, we choose 73 largest singular values to approximating the original person-by-project matrix of size 456 $\times$ 181.[5]

4.3 Experimental Results

Table 2 summarizes the results of comparing various algorithms on the *Institute* data set. Although the experiments were not extensive, our algorithm was best in precision and second-best in recall measure. This is quite promising in that there are only 5% positive examples available. We believe that projecting the original entity-by-relation matrices onto latent spaces is quite effective so that and a simple logistic regression trained by gradient descent learnt regression coefficients well. In particular, a factorized matrix captures the characteristics of the original relationships by orthogonal vectors and factor values. In order to verify the usefulness of the projection of different relation matrices on latent spaces, we plotted the distances between entities in different latent spaces. Figure 1 confirmed one of our hypotheses that four different latent spaces represent well the relationships among a given set of people. We were surprised that the simple methods such as *Cosine similarity* and *Katz* algorithm showed relatively good performance. Since there are approximately 5% (6,140 out of 103,740) positive

[4] The distances between two entities were measured by L2 norm in each of four different latent spaces such as co-project, co-publication, co-research interest, and academic relationship.

[5] For person-by-publication matrix 285 out of 456, for person-by-interest 52 out of 70, for person-by-academic relation (advisor/advisee) 66 out of 81 were chosen.

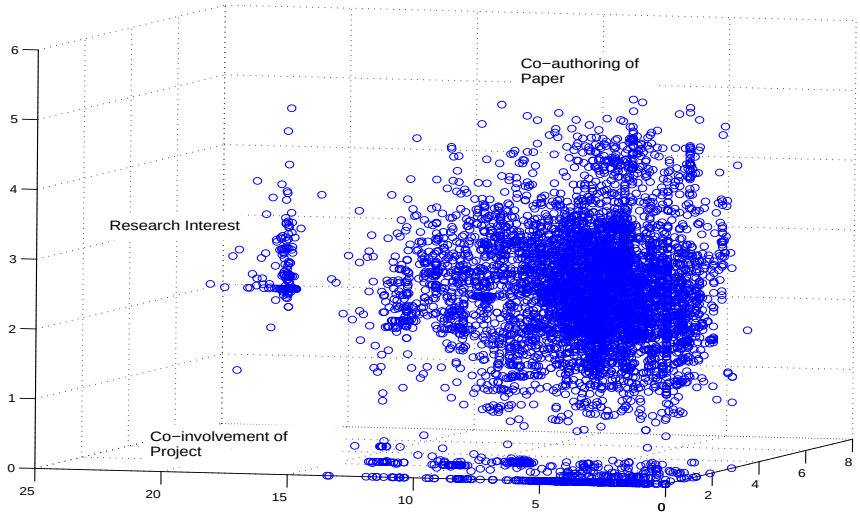

Figure 1: The relationships among people abstracted by three different latent spaces (research interest, co-authoring of paper, and co-involvement of project.

	cGraph	our method	Katz	Cosine	PA	Jaccard
micro-avg Recall (%)	40.20	37.43	22.30	19.10	18.90	08.20
micro-avg Precision (%)	55.00	57.04	31.42	28.24	23.10	12.11
micro-avg Accuracy (%)	96.02	94.08	71.20	54.11	51.12	44.14

Table 2: The average 10 fold sequential cross validation results in different performance measures. PA stands for the preferential attachment.

examples in the re-complied *Institute* data, the performance of our method might be affected by such a negative-biased data set. The Receiver Operating Characteristic (ROC) curve is used to further investigate the performance of the proposed algorithm because the ROC curve is not sensitive to the distribution of classes. Each of the axes in a ROC curve only uses one aspect of a given data set. In particular, the y-axis represents the "true positive" rate that is calculated by the ratio of the number of examples labeled as positive to the number of total positive examples in the data set whereas the x-axis presents the "false positive" rate that is the ratio of the number of instances incorrectly assigned to positive (i.e., predicted as positive that are truly negative) to the number of total negative examples [4]. Figure 2 shows the ROC curves for the proposed method from a 10 fold sequential cross validation test. The performance of our method is stable across the different folds.

5 Conclusions

In this paper we proposed a new non-parametric link analysis algorithm that predicts a potential link between entities given a set of different relational patterns. The proposed algorithm represents different types of relations among people by constructing the corresponding factorized matrices from the original entity-by-relation matrices. A factorized matrix is a projection of the original matrix onto a latent space. The pre-

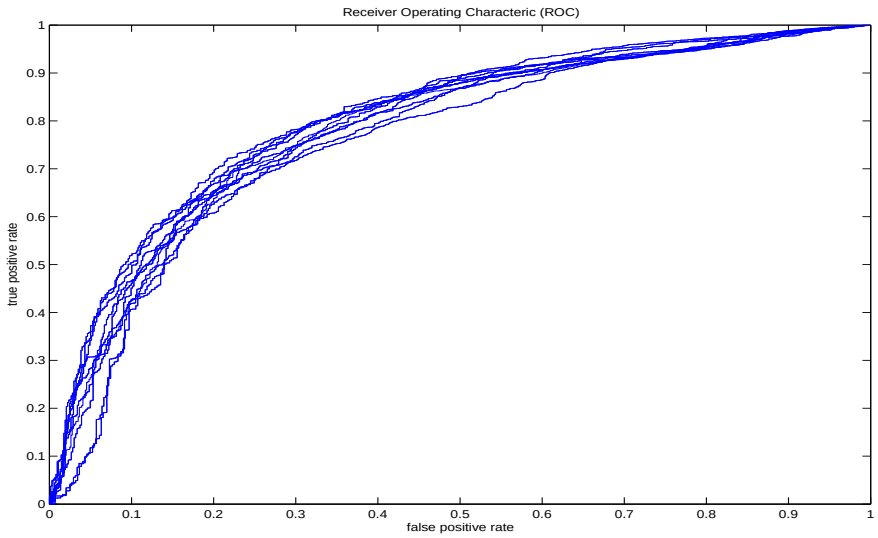

Figure 2: ROC curves of 10 fold sequential cross validation test.

diction of a possible link between entities is done by linearly summing the weighted distances in the latent spaces. A distance between two entities is measured in a latent space. Logistic regression is used to estimate regression coefficients of distances in the latent spaces.

We compared the performance of our method with various algorithms on the modified *Institute* data set. Although the experiments were not extensive, the performance of our method was best in precision and second-best in recall measure. This is quite promising in that there are only 5% positive examples available. We believe that the projection of the original entity-by-relation matrices onto latent spaces is quite effective so that a simple logistic regression trained by gradient descent learnt regression coefficients well. In particular, a factorized matrix captures the characteristics of the original relationships by orthogonal vectors and factor values.

As future work, we would like to test the usefulness of our method with other relational data sets, such the *Citeseer* data or the *IMDB* data. Since our method assumes that there are different types of interactions between entities, we might need to investigate how to convert a data set based on an interaction type into one based on several different types of interaction. This can be done by analyzing given data further. For example, co-publication data such as the *Citeseer* can be converted into one with different publications topics – publications on computer science to publications on artificial intelligence, architecture, operating systems, etc.

Another possible extension is to make use of the kernelization trick that projects the original entity-by-relation matrix into a high dimension. A kernelized logisitic regression might be used for this idea. However it would be computationally very expensive because the size of kernel is 103,740 $\times$ 103,740. Komarek and Moore proposed several techniques that improve the performance of a regular logistic regression [6].

Finally, instead of answering "Whom will I work with in the near future?" one might want to know "Who will be the most relevant collaborator in a particular context?" A context will be best determined by the content of the interaction. For example, a particular research field is one of the contexts for the research community. This can be done by analyzing the textual description of what a person is working on (e.g., abstract

of the papers, persons' home pages, or research statement). To this end, we would like to incorporate information retrieval (or text mining) techniques into the proposed method. This make the proposed method to distinguish the context of a link as well as its strength.

References

[1] C. J. Anderson, S. Wasserman, and B. Crouch. A p* primer: Logit models for social networks. *Social Networks*, 21:37–66, 1999.

[2] D. Cohn and T. Hofmann. The missing link - a probabilistic model of document content and hypertext connectivity. In *Proceedings of Neural Information Processing Systems*, 2000.

[3] W. DuMouchel and D. Pregibon. Empirical bayes screening for multi-item associations. In *Proceedings of ACM SIGKDD*, pages 67–76, 2001.

[4] T. Fawcett. Roc graphs: Notes and practical considerations for researchers. 2003.

[5] L. Getoor. Link mining: A new data mining challenge. *ACM SIGKDD Explorations Newsletter*, 5(1):84–89, 2003.

[6] P. Komarek and A. Moore. Making logistic regression a core data mining tool: A practical investigation of accuracy, speed, and simplicity. 2005.

[7] J. Kubica, A. Moore, D. Cohn, and J. Schneider. Finding underlying connections: A fast graph-based method for link analysis and collaborative queries. In *Proceedings of International Conference on Machine Learning*, pages 392–399, 2003.

[8] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *Proceedings of Conference on Information and Knowledge Management*, pages 556–559, 2003.

[9] R. J. Mooney, P. Melville, and L. R. Tang. Relational data mining with inductive logic programming for link discovery. In *Proceedings of the National Science Foundation Workshop on Next Generation Data Mining*, 2002.

[10] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.