Chance-Constrained Programs for Link Prediction

Janardhan Rao Doppa, Jun Yu, Prasad Tadepalli
School of EECS
Oregon State University
Corvallis, OR 97330
{doppa,yuju,tadepall}@eecs.oregonstate.edu

Lise Getoor
Computer Science Dept.
University of Maryland
College Park, MD 20742
getoor@cs.umd.edu

Abstract

In this paper, we consider the link prediction problem, where we are given a partial snapshot of a network at some time and the goal is to predict additional links at a later time. The accuracy of the current prediction methods is quite low due to the extreme *class skew* and the large number of potential links. In this paper, we describe learning algorithms based on chance constrained programs and show that they exhibit all the properties needed for a good link predictor, namely, allow *preferential bias* to positive or negative class; handle *skewness* in the data; and *scale* to large networks. Our experimental results on three real-world co-authorship networks show significant improvement in prediction accuracy over baseline algorithms.

1 Introduction

Network analysis, including social networks, biological networks, transaction networks, the web, and a large assortment of other settings, has received a lot of interest in recent years. These networks evolve over time and it is a challenging task to understand the dynamics that drives their evolution. Link prediction is an important research direction within this area. The goal here is to predict the potential future interaction between two nodes, given a (partial) current state of the graph.

This problem occurs in several domains. For example: in citation networks describing collaboration among scientists, where we want to predict which pairs of authors are likely to collaborate in future; in social networks, where we want to predict new friendships; and in biological networks where we want to predict which proteins are likely to interact. On the other hand, we may be interested in anomalous links; for example, in financial transaction networks, where unlikely transactions might indicate fraud, and on the web, where they might indicate spam.

There is a large literature on link prediction [4]. Early approaches to this problem are based on defining a measure for analyzing the *proximity* of nodes in the network [1, 16, 10]. For example, shortest path, common neighbors, katz measure, Adamic-adar etc. fall under this category. Liben-Nowell and Klienberg [10] studied the usefulness of all these topological features by experimenting on bibliographic datasets. It was found that, no one measure is superior in all cases [10]. Statistical relational models were also tried with some amount of success [5, 6, 19, 17]. Recently, the link prediction problem is studied in the supervised learning framework by treating it as an instance of binary classification [7, 8, 3, 20, 21]. These methods use the topological and semantic measures defined between nodes as features for learning classifiers. Given a snapshot of the social network at time t for training, they consider all the positive links present at time t as positive examples and consider a large sample of negative links (pair of nodes which are not connected) at time t as negative examples. The learned classifiers performed consistently, although the accuracy of prediction is still very low. There are several reasons for this low prediction accuracy. One of the main reasons is the huge class skew associated with link prediction (in large networks, it's not uncommon for the prior link probability on the order of 0.0001 or less); this makes the prediction problem very hard,

resulting in poor performance. In addition, as networks evolve over time, the negative links grow quadratically whereas positive links grow only linearly with new nodes. Further, in some cases we are more concerned with *link formation*, the problem of predicting new positive links, and in other cases we are more interested in *anomalous link prediction*, the problem of detecting unlikely links. In general, we need the following properties for a good link predictor: allow *preferential bias* to the appropriate class; handle *skewness* in the data; *scale* to large networks.

Chance-constraints and second order cone programs(SOCPs) [11] which are a special class of convex optimization problems have become very popular lately, due to the efficiency with which they can be solved using methods for semi-definite programs, such as interior point methods. They are used in a variety of settings such as feature selection [2], dealing with missing features [18], classification and ordinal regression algorithms that scale to large datasets [15], and formulations to deal with unbalanced data [14, 13]. These probabilistic constraints can be converted into deterministic ones using Chebyschev-Cantelli inequality, resulting in a SOCP. The complexity of SOCPs is moderately higher than linear programs and they can be solved using general purpose SOCP solvers like SeDuMi [1]. These classification algorithms that use chance-constraints satisfy all the requirements needed for learning a good link predictor as mentioned above. In this work, we show how these learning algorithms based on chance-constraints can be used for link prediction to significantly improve its performance. The main contributions of this paper include:

- We identify the important requirements of link prediction task and formulate it using the framework of chance-constrained programming, satisfying all the requirements.
- We show how this framework using chance-constraints can be used in different link prediction scenarios including ones where positive links are more important than negative links (e.g., link formation), and vice versa (e.g., anomalous link discovery) and the cases in which we see a lot of missing features.
- We perform a detailed evaluation on three real-world co-authorship networks: DBLP, Genetics and Biochemisty to investigate the effectiveness of our methods. We show significant improvement in link prediction accuracy.

The outline of the paper is as follows: In Section 2, we explain max-margin learning algorithms based on chance-constraints. We then describe how they can be used for link prediction problems. We describe applications of this framework in a variety of different settings in Section 3. In Section 4, we describe the datasets used for the experiments and the features used by our learning algorithms, discuss the evaluation metrics, and present our empirical evaluation. Finally, we conclude with some future directions in this line of research.

2 Learning algorithms using Chance-Constraints

In this work, we consider the link prediction problem as an instance of binary classification. We are given training data $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$ where, each $x_i \in \Re^n$ is a feature vector defined between two nodes and $y_i \in \{-1, +1\}$ is the corresponding label i.e., +1 and -1 stands for the presence or absence of an edge between the two nodes. In our case, the data is extremely skewed i.e., the number of negative examples $\gg$ the number of positive examples. For now, we work with only linear decision functions of the form $f(x) = w^T x - b$. However, all the formulations described below can be kernelized to construct non-linear classifiers.

2.1 Clustering-based SOCP formulation (CBSOCP)

In this formulation, we assume that class conditional densities of positive and negative points can be modeled as mixture models with component distributions having spherical covariances. Let k_1 and k_2 denote the number of components in the mixture model for positive and negative class respectively. We can cluster the positive and negative points separately and estimate the second order moments (μ, σ^2) of all the clusters. Given these second order moments, we want to find a discriminating hyperplane $w^T x - b = 0$, which separates these positive and negative clusters. More specifically, we want that with a very high probability any point on these clusters to lie on the correct side of the hyperplane.

[1] http://sedumi.ie.lehigh.edu

$$\begin{aligned} Pr(w^T X_i - b \geq 1) \geq \eta_1 : \forall i \in \{1 \cdots k_1\} \\ Pr(w^T X_j - b \leq -1) \geq \eta_2 : \forall j \in \{1 \cdots k_2\} \end{aligned} \tag{1}$$

Here X_i and X_j are random variables corresponding to the components of the mixture models for positive and negative classes, and η_1 and η_2 lower bound the classification accuracy of the two classes. The above probabilistic constraints can be written as deterministic constraints using Chebyshev-Cantelli inequality [12]. For further details on this conversion, readers are referred to [15, 13]. After this conversion and allowing slack variables ξ_i to handle noise leads us to the following soft-margin SOCP optimization problem:

$$\begin{aligned} \min_{w,b,\xi_i} \quad & \sum_{i=1}^{k} \xi_i \\ \text{s.t. } & y_i(w^T \mu_i - b) \geq 1 - \xi_i + \kappa_1 \sigma_i W : \forall i = 1, \cdots, k_1 \\ & y_i(w^T \mu_j - b) \geq 1 - \xi_j + \ \kappa_2 \sigma_j W : \forall j = 1, \cdots, k_2 \\ & W \geq \|w\|_2 \,, \ \ \xi_i \geq 0 : \forall i = 1, \cdots, k_1 + k_2 \end{aligned} \tag{2}$$

where $\kappa_i = \sqrt{\frac{\eta_i}{1-\eta_i}}$ and W is a user-defined parameter which lower bounds the margin between the two classes. The geometric interpretation of this formulation is that of finding a hyperplane which separates the positive and negative spheres whose centers and radii are μ_i and $\kappa_i \sigma_i$ respectively (See Figure 1(b)). Note that if we consider each point as one cluster(σ_i = 0), then the above formulation is exactly the same as SVMs. By solving the above SOCP problem, we get the optimum values of w and b, and a new data point x can be classified as sign$(w^T x - b)$.

This formulation is much more scalable to large datasets because the number of constraints in this formulation is linear in the number of clusters, whereas the number of constraints in SVM formulation is linear in the number of data points. It also allows us to introduce preferential bias by varying η_1 and η_2. In the case of link formation, we want to give more importance to positive links than negative links, i.e., $\eta_1 > \eta_2$.

However, this cannot handle the case of unbalanced data. One simple way to overcome this problem is to balance the data by constraining the number of clusters k_1 and k_2 i.e., $k_1 \approx k_2$. Note that, the assumption that mixture components have spherical covariances is a strong one. We conjecture that considering either diagonal or full covariance matrix instead of spherical covariances might give better results. However, we do not pursue this direction in our current work.

2.2 Max-Margin formulation with specified lower bounds on accuracy of the two classes(LBSOCP)

Suppose X_1 and X_2 represent the random variables which generate the data points from positive and negative class respectively. In this formulation, it is assumed that the class conditional densities can be modeled as Gaussians with means $\mu_i \in \Re^n$ and $\Sigma_i \in \Re^{n \times n}$ for $i = 1, 2$. We also assume that η_1 and η_2, the lower bounds on classification accuracies of the two classes, are given to us. The goal here is to construct a max-margin classifier with desired lower bounds on classification accuracies. Consider the following formulation:

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2} \|w\|_2 \\ \text{s.t. } & Pr(X_1 \in \mathcal{H}_2) \leq 1 - \eta_1 \\ & Pr(X_2 \in \mathcal{H}_1) \leq 1 - \eta_2 \\ & X_1 \sim (\mu_1, \Sigma_1), X_2 \sim (\mu_2, \Sigma_2) \end{aligned} \tag{3}$$

where $\mathcal{H}_1$ and $\mathcal{H}_2$ denote the positive and negative half spaces respectively. The chance-constraints $Pr(X_1 \in \mathcal{H}_2) \leq 1 - \eta_1$ and $Pr(X_2 \in \mathcal{H}_1) \leq 1 - \eta_2$ specify that false-negative and false-positive rate should not exceed $1-\eta_1$ and $1-\eta_2$ respectively. The above chance-constraints can be converted into deterministic ones using multi-variate generalization of Chebyshev-Cantelli inequality [12, 14,

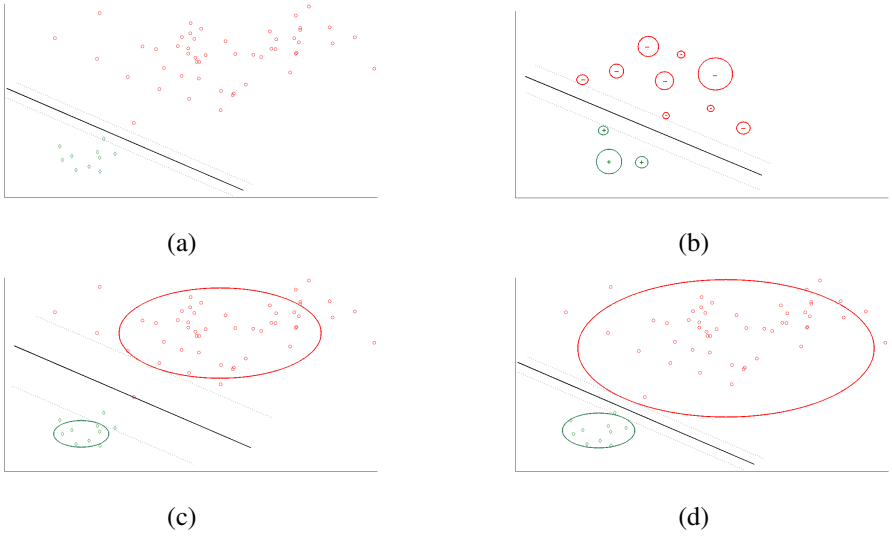

Figure 1: Geometric interpretation for (a) SVM (b) CBSOCP (c) LBSOCP (d) Effect of η_i on margin

13]. After this conversion and re-writing it in standard SOCP form, we get the following formulation,

$$\begin{aligned} \min_{w,b,t} \quad & t \\ \text{s.t. } & t \geq \|w\|_2 \\ & w^T\mu_1 - b \geq 1 + \kappa_1 \left\|C_1^T w\right\|_2 \\ & b - w^T\mu_1 \geq 1 + \kappa_2 \left\|C_2^T w\right\|_2 \end{aligned} \tag{4}$$

where, $\kappa_i = \sqrt{\frac{\eta_i}{1-\eta_i}}$, and C_1 and C_2 are square matrices such that $\Sigma_1 = C_1C_1^T$ and $\Sigma_2 = C_2C_2^T$. Note that, there exist such square matrices since Σ_1 and Σ_2 are positive semi-definite. The geometrical interpretation of the above constraints is that of finding a hyperplane which separates the positive and negative ellipsoids whose centers are at μ_1 and μ_2, shapes determined by C_1 and C_2, and size by κ_1 and κ_2 respectively i.e., $B(\mu_i, C_i, \kappa_i) = \left\{x|(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i) \leq \kappa_i^2\right\}$(see Figure 1 (c)). It is important to note that, the margin of the classifier changes for different values of η_i (see Figure 1 (c) and Figure 1 (d)). By solving the above SOCP problem using standard SOCP solvers like SeDuMi, we get the optimum values of w and b, and a new data point x can be classified as sign$(w^Tx - b)$.

This formulation has all the properties needed for the link prediction task. By varying the values of η_1 and η_2, we can introduce preferential bias towards positive links i.e., $\eta_1 > \eta_2$. It is scalable and can also handle unbalanced data.

3 Applications of the CCP framework

In this section, we will explain how our CCP framework can be used for a variety of applications in network analysis without any further changes. It is important to note that, the framework is flexible enough to be used in both the cases where positive links are more important than negative links and vice versa. For example, in link formation we want to give more importance to positive links i.e, $\eta_1 > \eta_2$ and in the case of anomalous link discovery we want to give more importance to negative links i.e., $\eta_2 > \eta_1$. We consider the applications that fall under both these categories separately and provide generic solutions that can be used across a wide range of applications. Since we are working with max-margin classifiers, our solutions are based on the margin of the learned classifier which is defined as $\left|w^Tx - b\right|$.

Positive links are more important: In this case, we use a validation set to determine the positive threshold m_+, which is defined as the minimum margin above which majority of the positive links

lie. Therefore, any positive link which has a margin more than m_+ will be positive with very high probability. Now during testing, we can rank all the positive links with margin $m > m_+$ according to their margin and such a ranked list can be used in variety of applications. For example, to recommend friends in an online social network(OSN), items in collaborative filtering, etc.

Negative links are more important: Similar to the previous case, we use the validation set to determine the negative threshold m_-, which is defined as the minimum margin above which majority of the negative links lie. Therefore, any negative link which has a margin more than m_- will be negative with very high probability. Now during testing, consider the set of all negative links with margin $m > m_-$. We can use this list of negative links for anomalous link discovery such as fraud detection, i.e., if any of these negatively predicted links is actually seen as a positive link, then it can be flagged as anomalous.

Missing features: In both the above cases, we may have some features missing. For example, if we use node attributes as features like user profiles in OSNs , then we may find incomplete profiles leading to missing features. And, chance-constrained programs can be used to handle this problem as well. For more details on this, readers are referred to [18].

4 Experimental Results and Discussion

In this section, we describe our experimental setup, description of datasets, features used for learning the classifier, evaluation methodology, followed by our results and discussion.

Datasets: We run our experiments on three real-world co-authorship networks, which are the same as the ones used in [20]. DBLP dataset was generated using DBLP collection of computer science articles. [2] It contains all the papers from the proceedings of 28 conferences related to machine learning, data mining and databases from 1997 to 2006. Genetics dataset contains articles published in 14 journals related to genetics and molecular biology from 1996 to 2005. Biochemistry dataset contains articles published in 5 journals related to biochemistry from 1996 to 2005. The genetics and biochemistry datasets were generated from the popular PubMed database.[3]

Dataset	No. of authors	No. of papers	No. of edges
DBLP	23,136	18,613	56,829
Genetics	41,846	12,074	1,64,690
Biochemistry	50,119	16,072	1,91,250

Table 1: Data Description

Experimental setup: We form the training dataset for our experiments in the same way as done in [20], which is as follows: For each dataset we have the data for 10 years. We consider the data from first 9 years for training and the data from the 10th year for testing. We consider all those links formed in the 9th year as positive training examples and among all the negative links (those links that are not formed in the first 9 years), we randomly collect a large sample and label them as negative training examples. Note that the features of these training examples are constructed based on the first 8 years of data. Similarly for the test set, we consider all the links that are formed during the 10th year as positive examples and collect a sample of all the negative links as negative examples. Note that the features of these testing examples are constructed based on the first 9 years of data.

Feature description: We used the same set of features between nodes as used in [20]. Their description is as follows: *Common neighbors:* the number of common neighbors for the two authors during training, *Social connectivity:* the total number of neighbors the two authors have during training, *Sum of papers:* the number of papers the two authors have written together during training, *Approximate Katz measure:* Katz measure approximated to paths up to length 4 with discount factor $\gamma = 0.8$ and *Semantic feature:* the cosine similarity between the titles of the papers written by the two authors during training.

[2] http://dblp.uni-trier.de/

[3] http://www.ncbi.nlm.nih.gov/entrez

Evaluation: We use precision and recall metrics from Information Retrieval context for evaluation, and compare the chance-constraints based algorithms (CBSOCP and LBSOCP) against SVMs and perceptron with uneven margins (PAUM) [9]. We selected PAUM as a strong baseline because it allows us to differentially emphasize the accuracies of the two classes based on positive and negative margins, τ_+ and τ_-. We rank all the test examples according to the margin of the classifiers and calculate precision and recall from Top-k by varying the value of k. Here, precision is defined as the percentage of true-positive links that were predicted correctly among the Top-k and recall is defined as the percentage of true-positive links that were predicted correctly out of the total true-positive links. We report the best results for SVMs by tuning its C parameter on validation set and for CBSOCP, we use $W = 1$, i.e., we want a margin of at least 1 between the two classes. For PAUM, we pick the best values for τ_- from $\{-1.5, -1, -0.5, 0, 0.1, 0.5, 1\}$ and for τ_+ from $\{-1, -0.5, 0, 0.1, 0.5, 1, 2, 5, 10, 50\}$ based on the validation set. We ran PAUM for a maximum of 1000 iterations or until convergence. Due to space constraints, we show the precision and recall curves for only one setting: $\eta_1 = 0.9$ and $\eta_2 = 0.7$. But we see the same kind of behavior for other similar configurations of η_1 and η_2 as well.

The precision and recall curves for all the 3 datasets are shown in Figure 2. As we can see, LBSOCP and CBSOCP significantly outperform SVMs and PAUM in both precision and recall for all the 3 datasets, except for biochemistry where PAUM performs better. Also, LBSOCP performs better than CBSOCP as expected. We achieve a recall of 52.79% and 46.23% using LBSOCP and CBSOCP when compared to 28.5% of SVMs and 33.59% of PAUM for DBLP dataset, 39.28% and 22.87% when compared to 13.39% of SVMs and 7.66% of PAUM for Genetics dataset, and 55.09% and 46.94% when compared to 25.48% of SVMs and 63.37% of PAUM for Biochemistry dataset. Encouragingly, LBSOCP and CBSOCP achieve a very good (80-90% of their overall recall) recall within $\approx$ Top-1000, which makes it a very good candidate for applications like recommendation and collaborative filtering where this property is very important. The reason why SVMs fail badly here is due to highly skewed class distribution. They try to get more negative examples correctly and in this process they find a hyperplane which has large margin for negative examples. We can clearly observe this behavior in our results - SVMs predict majority of its true-positives at the very end of the Top-k. With careful parameter tuning, PAUM can perform better in some cases (as we see with biochemistry dataset), but one cannot quantitatively relate these parameters to the performance.

We show the training times of different learners on various datasets in Table 2. As we can see, both CBSOCP and LBSOCP are orders of magnitude faster than SVM and PAUM, which makes them attractive to large networks. Note that, the SVMs were trained using popular LIBSVM and time would have been shorter if trained with SVMperf.

Learner	SVM	PAUM	CBSOCP	LBSOCP
DBLP	29.03	16.25	0.12	0.03
Genetics	265.77	27.30	2.00	0.02
Biochemistry	307.32	42.30	3.00	0.01

Table 2: Training time results (in secs)

5 Conclusions and Future Work

In this work, we showed how learning algorithms based on chance-constraints can be used to solve link prediction problems. We showed that they significantly improve the prediction accuracy over the traditional classifiers like SVMs. We explained how our framework using chance-constraints can be used in different scenarios—where positive links are more important than negative links (e.g., link formation prediction), where negative links are more important than positive links (e.g., anomalous link discovery) and cases in which we see a lot of missing features. In the future, we would like to experiment with other co-authorship networks like *arXiv* and test this framework in other settings like collaborative filtering, anomalous link discovery and link prediction cases with missing features. We would like to extend the current framework to a relational setting similar to Taskar's work [19]. However, formulating it as relational or structured prediction poses an enormous inference problem, especially in large networks. One possible approach is to take a middle path between complete independence and arbitrary relational structure.

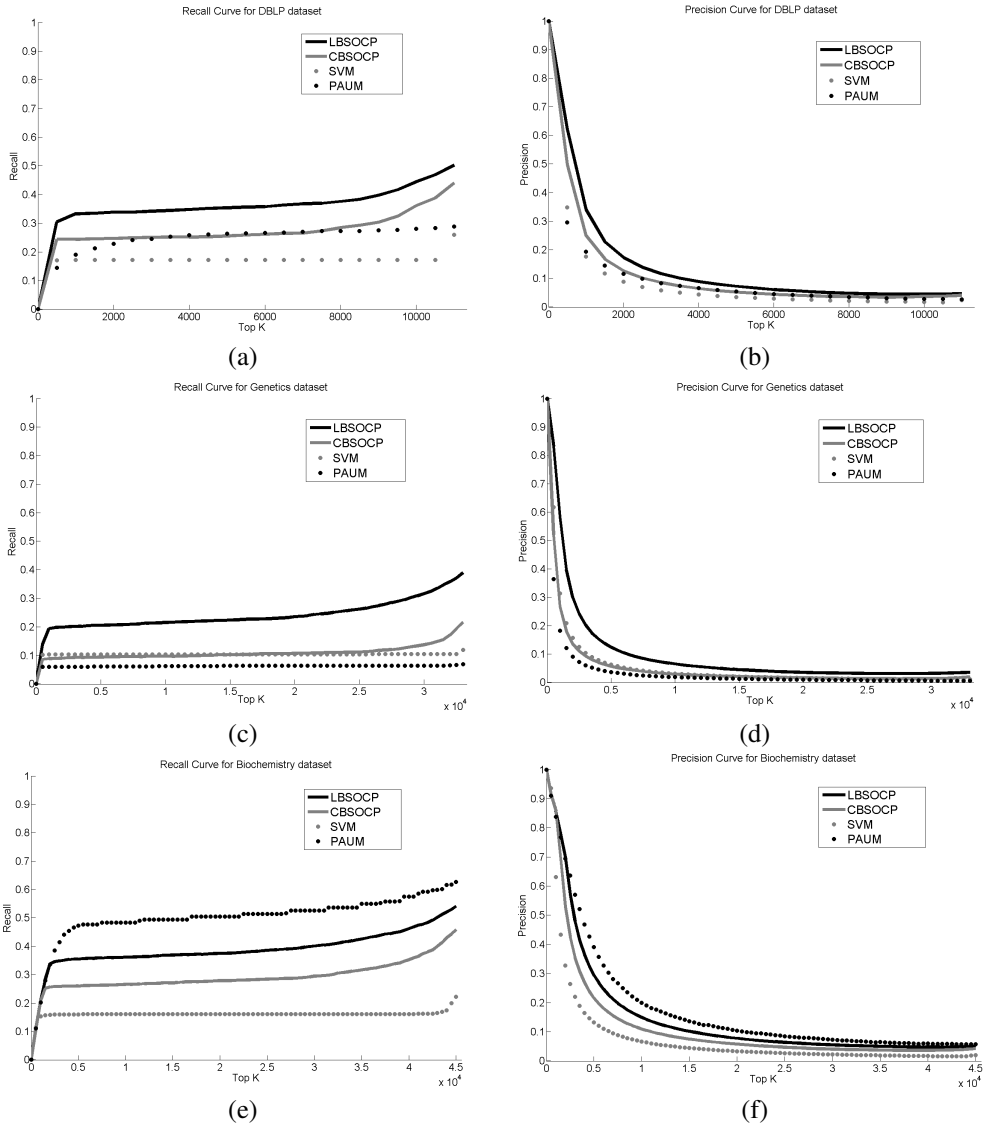

Figure 2: Recall and Precision curves for the three datasets - DBLP, Genetics and Biochemistry

6 Acknowledgements

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. FA8750-09-C-0179. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the DARPA, or the Air Force Research Laboratory (AFRL). This work is also partially funded by NSF Grant No. 0746930. We would like to thank Saketha Nath for useful discussions which helped our understanding on chance-constrained programs.

References

[1] Lada Adamic and Eytan Adar. Friends and neighbors on the web. *Social Networks*, 25:211–230, 2001.

[2] Chiranjib Bhattacharyya. Second order cone programming formulations for feature selection. *Journal of Machine Learning Research (JMLR)*, 5:1417–1433, 2004.

[3] Mustafa Bilgic, Galileo Mark Namata, and Lise Getoor. Combining collective classification and link prediction. In *Workshop on Mining Graphs and Complex Structures at the IEEE International Conference on Data Mining (ICDM-2007)*, 2007.

[4] Lise Getoor and Christopher P. Diehl. Link mining: a survey. *SIGKDD Explorations*, 7(2):3–12, 2005.

[5] Lise Getoor, Nir Friedman, Daphne Koller, and Benjamin Taskar. Learning probabilistic models of relational structure. In *Proceedings of International Conference on Machine Learning (ICML)*, 2001.

[6] Lise Getoor, Nir Friedman, Daphne Koller, and Benjamin Taskar. Learning probabilistic models of link structure. *Journal of Machine Learning Research*, 3:679– –707, 2002.

[7] Mohammad Hasan, Vineet Chaoji, Saeed Salem, and Mohammed Zaki. Link prediction using supervised learning. In *Proceedings of SDM workshop on Link Analysis Counterterrorism and Security*, 2006.

[8] Hisashi Kashima and Naoki Abe. A parameterized probabilistic model of network evolution for supervised link prediction. In *Proceedings of International Conference on Data Mining (ICDM)*, 2006.

[9] Yaoyong Li, Hugo Zaragoza, Ralf Herbrich, John Shawe-Taylor, and Jaz S. Kandola. The perceptron algorithm with uneven margins. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 379–386, 2002.

[10] David Liben-nowell and Jon Kleinberg. The link prediction problem for social networks. In *Proceedings of International Conference on Knowledge Management (CIKM)*, 2003.

[11] Miguel Sousa Lobo, Lobo I, Lieyen Vandenberghe, Herv Lebret, and Stephen Boyd. Applications of second-order cone programming. *Linear Algebra and its Applications*, 238:193–228, 1998.

[12] A.W. Marshall and I. Olkin. Multivariate chebyshev inequalities. *Annals of Mathematical Statistics*, 31:1001–1014, 1960.

[13] J. Saketha Nath. *Learning Algorithms using Chance-Constrained Programming*. Ph.d dissertation, Computer Science and Automation, IISc Bangalore, 2007.

[14] J. Saketha Nath and Chiranjib Bhattacharyya. Maximum margin classifiers with specified false positive and false negative error rates. In *Proceedings of SIAM International Conference on Data Mining (SDM)*, 2007.

[15] J. Saketha Nath, Chiranjib Bhattacharyya, and M. Narasimha Murty. Clustering based large margin classification: a scalable approach using socp formulation. In *Proceedings of International Conference on Knowledge Discovery and Data Mining (KDD)*, 2006.

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

[17] Alexandrin Popescul, Rin Popescul, and Lyle H. Ungar. Statistical relational learning for link prediction. In *Proceedings of IJCAI workshop on Learning Statistical Models for Relational Data*, 2003.

[18] Pannagadatta K. Shivaswamy, Chiranjib Bhattacharyya, and Alexander J. Smola. Second order cone programming approaches for handling missing and uncertain data. *Journal of Machine Learning Research (JMLR)*, 7:1283–1314, 2006.

[19] Ben Taskar, Ming fai Wong, Pieter Abbeel, and Daphne Koller. Link prediction in relational data. In *Proceedings of Annual Conference on Neural Information Processing Systems (NIPS)*, 2003.

[20] Chao Wang, Venu Satuluri, and Srinivasan Parthasarathy. Local probabilistic models for link prediction. In *Proceedings of International Conference on Data Mining (ICDM)*, 2007.

[21] Elena Zheleva, Lise Getoor, Jennifer Golbeck, and Ugur Kuter. Using friendship ties and family circles for link prediction. In *2nd ACM SIGKDD Workshop on Social Network Mining and Analysis (SNA-KDD)*, 2008.