HALLP: A Hybrid Active Learning Approach to Link Prediction Task

Ke-Jia Chen, Jingyu Han, Yun Li
School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing, P. R. China
Email: chenkj@njupt.edu.cn, hjysky@gmail.com, liyun@njupt.edu.cn

***Abstract*—A new link prediction method using active learning technique, named HALLP, is proposed in this paper. The method provides the user with most useful examples from the large number of unlabeled examples (i.e. unlinked node pairs in the network) for query. Once labeled by users, these examples will be fed to the learner for the improvement of the link predictor in next round. The utility of an example is decided by its uncertainty measure calculated simultaneously by its local structure and its hierarchical structure in networks. Experiments indicate link prediction method can be improved with the use of active learning techniques and both the local structure and global structure are beneficial for selecting examples with high utility.**

***Index Terms*—link prediction, active learning, link mining, social network analysis**

I. Introduction

The network (for example, online social networks, email networks, communication networks, and biological networks, etc.) and its applications have penetrated into every aspect of human life. The needs to develop new applications based on potential information from the massive network data are also increased. For example, in bioinformatics area, the prediction of relation and regulation of proteins can guide the experiment designers to discover new biological factors [1]. As a critical research sub-field of link mining, link prediction is concerned with the problem of predicting unknown portion (or the future structure) of the network from the known portion of the network. Different from traditional data mining tasks, link prediction investigates not only the attributes of individual nodes, but also the relationships among nodes. Link prediction has a wide range of application scenarios, such as bibliographic domains, recommendation systems and criminal investigations. In citation networks, for example, scholars can find the papers potentially useful to cite and managers can identify future core papers with the help of link prediction [2].

The classic link prediction framework is under the (semi-)supervised learning setup where a model is trained with an existing dataset composed of examples (i.e. node pairs) annotated with certain attributes. However, link prediction in sparse networks presents a significant challenge due to the inherent disproportion of links that can form to links that do form [3]. Further exploration on a large number of unlinked node pairs, most of which are unlabeled examples, may improve the performance of link prediction.

The paper proposes to use active learning technique to select the most possible useful unlabeled examples from all unlabeled examples. These examples will first be labeled with the assistance of the users. The predictor will then be retrained with the updated set of examples and could be further improved. One of the advantages of active learning is that it can minimize the amount of labeled data without sacrificing much quality of the learned models [4]. Thus the users in the networks will not be bothered too much in the labeling process. The link prediction method proposed in this paper is named HALLP - Hybrid Active Learning approach for Link Prediction. HALLP identifies the examples with higher utility from unlabeled examples in networks to query. The utility of the example in networks is decided by both local structure and global structure of the example in networks, which will be described in details later.

The remainder of the paper is organized as follows. Section II introduces background and notation of our work. The proposed method is described in Section III. Section IV reports an empirical evaluation within two datasets. The paper is concluded in Section V.

II. Background and Notation

A. Link Prediction

Generally, the network data can be visualized and represented as a graph $G=<V,E>$. Each node $v\in V$ corresponds to an object (a person, a paper, a product, etc.) in a group and each edge $e=<v^{\alpha},v^{\beta}>_t\in E$ represents an interaction between node v^{α} and node v^{β} at a particular time *t*. Take the co-authorship network as an example, the nodes represent authors, and two authors are connected by an edge if they wrote one or more papers together.

Link prediction is usually described as the task to predict the likelihood of a link existence between an arbitrary pair of nodes. Therefore, it can be viewed as the

Manuscript received January 1, 2013; revised August 2, 2013; accepted September 2, 2013.

Corresponding author: Ke-Jia CHEN, Tel.: +86-18912963366, E-mail address: chenkj@njupt.edu.cn

doi:10.4304/jcp.9.3.551-556

problem of refreshing an adjacency matrix representing the structure of a network. However, the graph or matrix representing the real-world network is sparse, that is, the number of edges known to be present is often significantly less than the number of edges known to be absent. The extremely imbalanced distribution presents a challenge for link prediction methods based on supervised learning. Moreover, the data in real-world networks has a large scale. Developing efficient link prediction methods applicable to large-scale data is another challenge.

A variety of works on link prediction have been developed. The early works include probabilistic relational models [5], structural logistic regression models [6], and stochastic relational models [7]. Recently, machine learning techniques have been used in link prediction. Al Hasan et al. [8] used several supervised learning models for link prediction and concluded that SVM outperformed other models in all performance measures. Clauset et al. [9] investigated the hierarchical structure of social networks to predict missing connections in partially known networks with high accuracy. Kashima et al. [10] proposed a method named Link Propagation (inspired from label propagation algorithm) as a new semi-supervised learning method for link prediction problems. Parimi and Caragea [11] used a clustering approach in a social network service to predict potential friendships. Brouard and Szafranski [12] addressed link prediction as an output kernel learning task through semi-supervised output kernel regression.

In the paper, link prediction is regarded as a binary classification problem for the node pairs in the network. The linked node pairs are marked as positive examples and part of the unlinked node pairs are marked as negative examples. A binary classifier is trained to determine the likelihood of the link existence between each two unlinked nodes, based on the predetermined features. Let $\hat{x}^{<\alpha,\beta>} := \{x_1, x_2, ..., x_n\}$ describe an arbitrary pair of nodes $< v^\alpha, v^\beta >$. Each x_i represents the attribute regarding an interrelationship between node v^α and node v^β in the given network. The domain of x_i can be either discrete or continuous. Let $L := \{l\}$ denotes the label set of node pairs. The label represents the existence of a link.

$$l^{<\alpha,\beta>} = \begin{cases} 1, \ if < v^\alpha, v^\beta > \in E \\ 0, \ if < v^\alpha, v^\beta > \notin E \end{cases} \tag{1}$$

In a prediction model, the hypothesis h is established by $score^{<\alpha',\beta'>} := h(\hat{x}^{<\alpha',\beta'>})$, where $score^{<\alpha',\beta'>}$ denotes a connection weight or possibility to a new pair of nodes $< v^{\alpha'}, v^{\beta'} >$. Several classification models (decision tree, *k*-NN, SVM, RBF network, etc.) have been developed and been compared in supervised learning framework to solve the link prediction problem [8]. In this paper, SVM is employed as the prediction model as it shows the best performance among these models.

B. Active Learning

In many real-world applications, some pairs of existing nodes that are not connected may connect in the future. Therefore, these examples are treated as unlabeled instead of negative in this paper. The paper aims to employ active learning techniques to better use the abundant unlabeled examples in networks. Active learning attempts to achieve better accuracy by posing queries in the form of unlabeled examples to be labeled by users. Actually, queries can be obtained in a natural way with the activity of users in networks. For example, in online social network, a user can be recommended with some possible friends, and a feedback can be given to the system once the user gives the confirmation. It is a reasonable way to predict links in an active learning process.

In the pool-based active learning cycle, the learner is initially provided with a training set *L* composed of labeled examples and a pool of unlabeled examples *U*. At each step, a batch of *k* examples in *U* are selected and labeled, and then added to the labeled corpus *L* and removed from *U*. One of the general techniques in active learning is uncertainty sampling, which is to select examples that are most uncertain for the current model [4]. In this paper, uncertain sampling is used in the pool-based active learning cycle.

To date, most of the active learning methods focused on data assumed to be independent and identically distributed (IID) [13]. Recently, some researchers [13][14] began to use active learning in network data by exploiting the relationships of objects. Their methods are used to solve node classification problem. So far, active learning had not been introduced into link prediction task. In this paper, we propose a new active learning approach for link prediction using the utility of *links* instead of nodes.

III. The Proposed Method - Hallp

The section describes HALLP, which is a novel hybrid active learning approach for link prediction problem. The method exploits the information of unlabeled examples based on both the local structure and the global structure of the network. The pseudo code for HALLP is described in Algorithm 1.

In HALLP, the linked node pairs in the observed social network are seen as labeled positive examples. Unlike the usual way, the unlinked node pairs here are treated as unlabeled examples instead of negative examples since some of them may be linked in future. Let *U* denote the unlabeled data set, and *L* denotes the labeled data set, $L = P \bigcup N$. *P* and *N* denote the set of positive examples and negative examples, respectively. Negative examples can be sampled from *U* because the majority of examples in *U* are labeled negative. In this paper, training examples are sampled from *L* and *U* in a simple random manner considering the large scale of network data. But certainly, other sampling methods can be tried to select more possible negative examples.

Algorithm 1. The pseudo code of HALLP

```
Input:
$G=<V,E>$: a network;
$L$: a set of labeled examples;
$U$: a set of unlabeled examples;
$T$: the training set;
$M$: the prediction model;
Output:
$G'=<V,E'>$: the updated network;
Process:
Choose $n/2$ positive examples from $L$ and $n/2$ negative examples from $U$ respectively, and add them into $T$
Train $M$ on $T$
While stopping criterion not met do
    for each $<v^\alpha, v^\beta> \in U$ do
        $Form_1 \leftarrow$ UtilityLocal($<v^\alpha, v^\beta>$);
        $Form_2 \leftarrow$ UtilityGlobal($<v^\alpha, v^\beta>$);
        Utility($<v^\alpha, v^\beta>$) $\leftarrow c_1 \bullet Form_1 + c_2 \bullet Form_2$;
    end
    for $i \in 1..k$ do
        $u \leftarrow \arg\max$ Utility($<v^\alpha, v^\beta>$);
        Label $u$ by the user
        $U \leftarrow U - \{u\}$;
        $T \leftarrow T \bigcup \{u\}$;
    end
    $M \leftarrow$ retrain $M$ on $T$;
end
Predict $E'$ in the network $G'$ based on $M$;
Return $G'$.
```

The pool-based active learning cycle is used, as the queries can be naturally obtained in most of applications. Each time, the selected examples will be labeled by the user. In experimentation, they are labeled according to whether they are linked or not in a later period of time. The newly labeled examples will be added to T as additional examples either positive or negative to retrain the learner in the next round. Fed with more training examples, the learner could be more accurate than before. The above process can be repeated several times. The updated M will finally be used to predict links. As a result, new links can be predicted and E is updated to E'.

Uncertainty sampling [15] strategy is employed to measure the utility of the examples, where the learner will query the examples with the least certainty to label. Adding these examples is expected to contribute more for training a model. In HALLP, the utility of an individual example is calculated based on both its local structure and its global structure in the network. The local structure is described by local features (for example, the number of the common neighbors of a pair of nodes or the shortest distance between a pair of nodes), while the global structure is described by the hierarchical random graph of the network [9].

The definition of the local utility is shown in Eq. (2), where the *score* is decided by a learner. The value is normalized to interval [-1, +1].

$$\text{UtilityLocal}(<v^\alpha, v^\beta>) = \text{Norm}(1/\text{Abs}(score^{<\alpha,\beta>})) \quad (2)$$

The learner M used by HALLP may be implemented in different ways. The paper employs SVM as the predictive model. An uncertainty sampling strategy for SVM involves querying the unlabeled examples which are close to the linear decision boundary. Therefore, the score of an example denotes the distance of the example to the hyperplane. The score of an example is defined by Eq.(3), where $\hat{w} = (w_1, w_2, ..., w_n)$ is the parameter vector that specifies the model.

$$\begin{aligned} score^{<\alpha,\beta>} &= \hat{w}^T \hat{x}^{<\alpha,\beta>} \\ &= w_1 x_1 + w_2 x_2 + ::: + w_n x_n \end{aligned} \quad (3)$$

Given the training data, the optimal parameter w is found by minimizing the objective function in Eq.(4).

$$\sum Max\{1 - y^{<\alpha,\beta>}, 0\} + c\|\hat{w}\|_2^2 \quad (4)$$

The definition of the global utility is shown in Eq. (5) inspired by the work of Clauset et al [9].

$$\text{UtilityGlobal}(<v^\alpha, v^\beta>) = \text{Norm}(1/\text{Abs}(P^{<\alpha,\beta>}\text{-}0.5)) \quad (5)$$

The idea is based on the fact that in the same community nodes are more possible to be connected, while in the communities with high distance, the possibility to be connected could be low. Hierarchical structure includes organizations (communities) at all scales in a network. The more closely related pairs of nodes, the lower their lowest common ancestors. The probability of a connection between two nodes can be calculated according to their degree of relatedness in a hierarchical random graph. The node pairs with connection probability close to 0.5 are considered to have more utility to the system. One of the advantages to use global features is to avoid the outliers in the network, which could have high uncertainty but might be not useful to query.

IV. Experiments

A. Data and Features

In experimentation, two data sets [1] were used to validate the proposed approach: Enron email corpus and DBLP database.

Enron dataset contains 252,759 emails from 151 Enron employees, mainly its senior managers. In experiments, we focused on emails sent from and to these 151 people. The link in Enron is defined as an email communication from node v^α to node v^β. The data in DBLP dataset were derived from a snapshot of the Computer Science bibliography from 1995 to 2002, including 29,459 authors and 419,751 publications. The link in DBLP is defined as a co-authorship between node v^α and node v^β. Comparing to Enron, the adjacency matrix is much sparser and the data scale is much larger in DBLP dataset.

[1] Enron email corpus is available at
http://www.cs.cmu.edu/~enron/;
DBLP database is available at
http://dblp.uni-trier.de/xml/.

The paper aims to verify the effectiveness of the proposed method both in a small-scale network and in a large-scale network.

Each dataset is partitioned into two non-overlapping sub-ranges according to the range of time (i.e. the time-stamp of emails or papers). The former is selected as the training dataset and the latter as the testing dataset. The linked node pairs appear in the testing dataset but not in the training dataset should be predicted as positive examples, and the unlinked node pairs in the whole dataset should be predicted as negative examples.

Six local topological features are extracted from the directed graph of Enron and five from the undirected graph of DBLP (TABLE I), where "√" denotes the feature is used in the dataset while "×" denotes the feature is not used in the dataset.

TABLE I.

TOPOLOGICAL FEATURES IN TWO NETWORKS

Features	Enron	DBLP
Number of common neighbors	√	√
Shortest distance	√	√
Jaccard coefficient	√	√
Difference in betweenness centrality	√	√
Difference in the number of in-links	√	×
Clustering coefficient	√	√

All the topological features are described as follows:

- *Number of common neighbors*. Newman [16] verified a correlation between the number of common neighbors of node v^{α} and node v^{β} and the possibility they will collaborate in the future. In Enron, the number of common neighbors is defined as the number of nodes that is connected to both the node *From* and the node *To*.
- *Shortest distance*. The shortest distance is chosen as an important feature because Kleinberg [17] discovered that in social network most of the nodes are connected with a very short distance.
- *Jaccard coefficient*. The Jaccard coefficient represents the relative value of the number of common neighbors. Let $\Gamma(v)$ denote the set of immediate neighbors of node v in G. The Jaccard coefficient is defined in Eq.(6):

$$J(v^{\alpha}, v^{\beta}) = \frac{\left|\Gamma(v^{\alpha}) \cap \Gamma(v^{\beta})\right|}{\left|\Gamma(v^{\alpha}) \cup \Gamma(v^{\beta})\right|} \tag{6}$$

- *Difference in betweenness centrality*. Betweenness centrality [18] represents the extent to which a node lies on the paths between other nodes and it can also be interpreted as measuring the influence a node has over the spread of information through the network.
- *Difference in the number of in-links*. The feature is defined as the difference in the number of in-links of node v^{α} and node v^{β}. The feature is extracted in Enron but not extracted in DBLP.
- *Clustering coefficient*. A node located in dense is likely to grow more edges than the one that is located in a sparser neighborhood. The clustering coefficient [16] of a node pair measures the localized density and is defined as follows:

$$C(v^{\alpha}, v^{\beta}) = \frac{3 \times \text{number of triangles with } v^{\alpha} \text{ and } v^{\beta}}{\text{number of connected triples with } v^{\alpha} \text{ and } v^{\beta}} \tag{7}$$

B. Methodology and Configurations

In the experiments, HALLP is benchmarked with two baseline methods (referred to Random-SVM and ALLP-SVM) and two graph-based methods named LP-HS (referring to the link prediction method based on hierarchical structure) [9] and LinkPro (referring to *Link Propagation*) [10], respectively. In Random-SVM method, the k examples to be labeled are randomly selected from the unlabeled dataset and the utility is calculated with Eq.(2). In ALLP-SVM method, active learning is used and the utility measure is calculated with Eq.(2).

For SVM, the tool LIBSVM Version 3.11 [19] is used, and the C-SVC algorithm and the RBF kernel are selected. The ratio of coefficients c_1 and c_2 in Algorithm 1 is set according to the AUC values for the initial results of Random-SVM and LP-HS.

Two sets of experiments are conducted. In the first set, the accuracy of ALLP-SVM and Random-SVM in Enron are compared in two settings. In the first setting, the number of initial training examples n changes in the range from 20 to 2000 with the step size of 20 while the number of examples selected to be queried is fixed (k=50, 100, and 500). In the second setting, the k value changes in the range from 10 to 500 with the step size of 20 while the n value is fixed (n=100, 500 and 1000). In the training set, the number of positive examples and negative examples is the same. For both ALLP-SVM and Random-SVM, the whole process is repeated five times with different training sets. The average result is recorded. Only one round active learning process is performed to observe the results. More rounds can certainly be performed if required.

Another set of experiments compares the ROC curves of the methods. In Enron dataset, HALLP, Random-SVM, ALLP-SVM, LP-HS and LinkPro were compared. For HALLP, Random-SVM and ALLP-SVM, the value k=500 and the value n=2,000. In DBLP dataset, HALLP, Random-SVM, ALLP-SVM and LP-HS were compared. LinkPro is not verified in DBLP dataset due to the extremely high complexity of time and space. For HALLP, Random-SVM and ALLP-SVM, the value k=2,000 and the value n=20,000.

C. Results

The geometrical accuracy graphs for Enron dataset with the increase of the value n when k=50, 100, and 500 are shown in Figure 1. The geometrical accuracy graphs for Enron dataset with the increase of the value k when *n*=100, 500, and 1000 are shown in Figure 2. The geometrical ROC graphs of five methods for Enron dataset are shown in Figure 3 (a) and the geometrical ROC graphs of four methods for DBLP dataset are shown in Figure 3 (b).

Figure 1 shows that for Enron dataset the accuracy of ALLP-SVM is better than Random-SVM in all settings of *k*. It is noticed that the accuracy values of both ALLP-SVM and Random-SVM are quite small when *n* is small in the setting of *k*=50 and 100. It is because the overall number of training data is too small to get an efficient predictor. But when *k* is 500, the number of training data is big enough to get a predictor with a better accuracy.

Figure 2 shows that for Enron dataset the accuracy of ALLP-SVM is evidently better than Random-SVM when *n*=500 and 1000. When *n* is 100, the accuracy of ALLP-SVM has no significant improvement compared with Random-SVM. From the shape of the curves, the performance of ALLP-SVM is more stable than Random-SVM. With increasing of *k*, the accuracy of both methods did not significantly increase. It is inconsistent with our expect that the more examples to query the users the higher the accuracy of the learner.

Figure 3 shows that for Enron dataset the ROC curve of HALLP is overall higher than Random-SVM, ALLP-SVM, LP-HS and LinkPro. ALLP-SVM is slightly worse than HALLP but much better than LP-HS and LinkPro. Notice that LP-HS does not perform as well as we expected. One possible reason is that the link prediction method based on hierarchical clustering is not applicable for all types of networks and especially for small-scale networks. For DBLP dataset, the ROC curve of HALLP is lightly better than the curve of ALLP-SVM, but much better than the curve of Random-SVM and LP-HS. The AUC values were also calculated for Figure 3, which also show that HALLP performs better than other methods in most situations. For example, in Enron, the AUC value of HALLP is 0.09 higher than Random-SVM.

Overall, the results verify that ALLP-SVM performs better than Random-SVM in the small-scale network as well as in the large-scale network. Using active learning does help to improve the performance of link prediction. Using the hierarchical structure, HALLP performs better than ALLP-SVM, but the degree of superiority depends on the performance of LP-HS in corresponding networks. The performance of HALLP is better than LP-HS since the former uses local structures in addition to global structures.

V. Conclusion

In this paper, the link prediction problem is converted to binary classification problem under machine learning framework.

The main contributions of the paper are:

- Active learning process is introduced into link prediction task and a link prediction method based on active learning is proposed.
- Both the local structure and the global structure of networks are considered for the selection of useful examples.

Experiments show the proposed method HALLP is superior to the baseline methods. Employing active learning is beneficial to link prediction task. Moreover, the active learning based link prediction method using both local and global structures is beneficial to the one using only local structures.

It is noticed that the links in the practical social network are very sparse, and the numbers of positive and negative examples can be greatly imbalanced. In this paper, the negative examples in the training set are only sampled randomly from the whole unlabeled dataset, which might produce some false negative examples and deteriorate the accuracy of the learner. In future work, the method can be improved by selectively sampling the unlabeled examples according to some semi-supervised methods. Besides, the real world networks are dynamic and the data in networks is better viewed as a sequence of snapshots of an evolving graph or as a continuous time process [20]. Whether or not the active learning process can be effectively merged into the link prediction in dynamic networks is another problem to be considered.

Acknowledgment

This work was supported by National Natural Science Foundation of China under grants No. 61100135, 61003040 and 61073114.

References

[1] D. Liben-Nowell and J. Kleinberg, “The link-prediction problem for social networks”, *Journal of the American Society for Information Science and Technology*, vol. 58, issue 7, pp. 1019-1031, 2007.

[2] N. Shibata, Y. Kajikawa, and I. Sakata, “Link prediction in citation networks”, *Journal of the American Society for Information Science and Technology*, vol. 63, issue 1, pp. 78-85, 2012.

[3] R. N. Lichtenwalter, J. T. Lussier, and N. V. Chawla, “New perspectives and methods in link prediction”. In *Proceedings of the 16th ACM SIGKDD*, 2010, pp. 243-252.

[4] B. Settles, “Active learning literature survey”, Computer Sciences Technical Report 1648, University of Wisconsin-Madison, 2009.

[5] L. Getoor and M. Sahami, “Using probabilistic relational models for collaborative filtering”, In: *Proceedings of the KDD Workshop on Web Usage Analysis and User Profiling*, 1999.

[6] A. Popescul and L. H. Ungar, “Structural logistic regression for link analysis”, In *Proceedings of the KDD Workshop on Multi-Relational Data Mining*, 2003.

[7] K. Yu, W. Chu, S. Yu, V. Tresp, and Z. Xu, “Stochastic relational models for discriminative link prediction”, In *Proceedings of the 20th NIPS*, 2006, pp. 1553-1560.

[8] M. Al Hasan, V. Chaoji, S. Salem, M. Zaki. Link prediction using supervised learning. In: Proceedings of the SDM workshop on Link Analysis, Counterterrorism and Security, 2006.

[9] A. Clauset, C. Moore, and M. E. J. Newman, “Hierarchical structure and the prediction of missing links in networks”. *Nature*, vol. 453, pp. 98-101, 2008.

[10] H. Kashima, T. Kato, Y. Yamanishi, M. Sugiyama, and K. Tsuda, “Link propagation: A fast semi-supervised learning algorithm for link prediction”, In *Proceedings of 9th SIAM on Data Mining*, 2009, pp. 1099-1110.

[11] R. Parimi and D. Caragea, “Predicting friendship links in social networks using a topic modeling approach”, In *Proceedings of the 15th PAKDD*, 2011, vol. 2, pp. 75-86.

[12] C. Brouard, F. d'Alché-Buc, and M. Szafranski, “Semi - supervised penalized output kernel regression for link prediction” In *Proceedings of the 28th ICML*, 2011, pp. 593-600.

[13] M. Bilgic and L. Getoor, “Link-based active learning” In *Proceedings of the NIPS Workshop on Analyzing Networks and Learning with Graphs*, 2009.

[14] M. Bilgic, L. Mihalkova, and L. Getoor, “Active learning for networked data”, In *Proceedings of the 27th ICML*, 2010, pp. 79-86.

[15] D. D. Lewis and J. Catlett, “Heterogeneous uncertainty sampling for supervised learning”, In *Proceedings of the 11th ICML*, 1994, pp. 148-156.

[16] M. E. J. Newman, “The structure and function of complex networks”, *SIAM Review*, vol. 45, issue 2, pp. 167-256, 2003.

[17] J. M. Kleinberg, “Navigation in a small world”, *Nature*, vol. 406, pp. 845, 2000.

[18] U. Brandes, “A faster algorithm for betweenness centrality”, *Journal of Mathematical Sociology*, vol. 25, pp. 163-177, 2001.

[19] C. C. Chang and C.-J. Lin, “LIBSVM: a library for support vector machines”, *ACM Transactions on Intelligent Systems and Technology*, vol. 2, issue 3, pp. 1-27, 2011.

[20] D. Vu, A. Asuncion, D. Hunter, and P. Smyth, “Continuous-time regression models for longitudinal networks”, In *Proceedings of the 25th NIPS*, 2011, pp. 2492-2500.

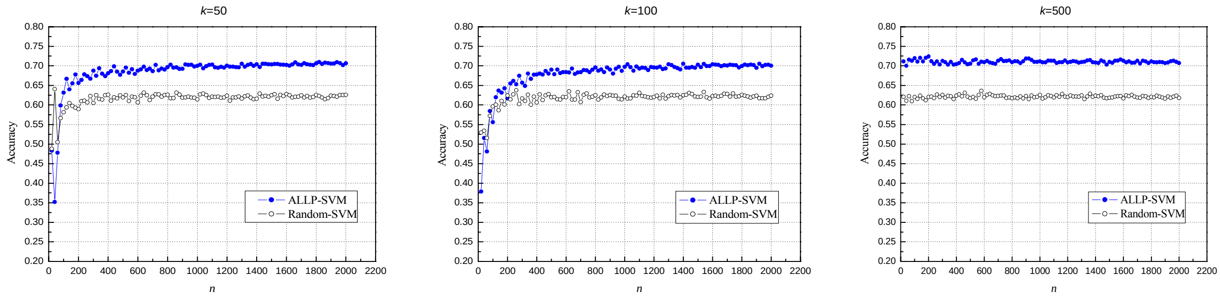


Figure 1. Geometrical accuracy graphs of ALLP-SVM and Random-SVM for Enron dataset when the *k* value is fixed

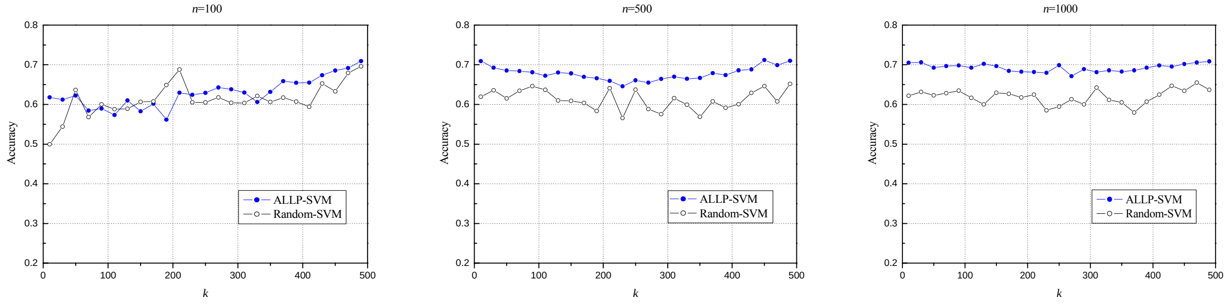
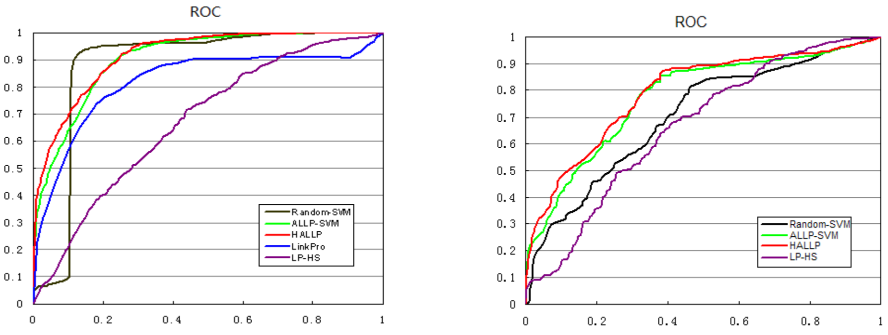


Figure 2. Geometrical accuracy graphs of ALLP-SVM and Random-SVM for Enron dataset when the *n* value is fixed



a. Enron dataset

b. DBLP dataset

Figure 3. Geometrical ROC graphs of HALLP, Random-SVM, ALLP-SVM, LP-HS and LinkPro