2012 IEEE 24th International Conference on Tools with Artificial Intelligence

Link Prediction in Complex Networks by Supervised Rank Aggregation

Manisha Pujari
LIPN CNRS UMR-7030
Université Paris Nord
93430-Villetaneuse, France
manisha.pujari@lipn.univ-paris13.fr

Rushed Kanawati
LIPN CNRS UMR-7030
Université Paris Nord
93430-Villetaneuse, France
rushed.kanawati@lipn.univ-paris13.fr

***Abstract*—Link prediction is a central task in the field of dynamic complex network analysis. A major trend in this area consists of applying a dyadic topological approach. Most of existing approaches apply machine learning algorithms where the link prediction problem is converted into a binary classification task. In this work, we propose a new dyadic topological link prediction approach applying *supervised social choice* algorithm. Given a training graph observed over a period $[t_0, t'_0]$, this interval is divided into two sub-intervals: the *learning* interval and the *labeling* one. For each unlinked couple of vertices in the learning interval, a topological feature vector is computed. The labeling interval is used to fix the class of each example (e.g. linking, not-linking). Instead of learning a classification model as it is the case when applying machine learning approaches, we use these data to learn weights to associate to each computed feature based on the ability of each attribute to predict observed links. These weights are then used within *weighted/supervised computational social choice algorithms* to predict new links at time $t > t'_0$. Two weighting schemes are experimented. We introduce weighted social choice rules by modifying classical voting approaches, namely: the Borda rule and the Kemeny aggregation rule. We also introduce our own concept of finding weights. We have implemented our approach on an academic co-authoring dataset (DBLP dataset). The preliminary results have been quite good, so we are working further on experimentation.**

I. Introduction

Analyzing dynamic large-scale networks is a major emerging topic in different research areas. Actually, many real-world systems can be modeled as an evolving network of interacting *actors*. This is namely the case of on-line social networks, collaboration networks (such as academic co-authoring networks, product co-purchasing, etc), biological systems (such as protein interaction networks) and computer science networks as the Internet and peer-to-peer networks. One of the major problems in studying dynamic evolution of complex networks, is the problem of *link prediction* [1], [2]. This refers to the problem of finding new associations (edges) in a network at a given point of time *t* when provided with the information about the network's temporal history before time *t*. The problem has a wide range of applications: recommender systems, identification of probable professional or academic associations in scientific collaboration networks, identification of structures of criminal networks and structural analysis in the field of microbiology or biomedicine, etc. A variety of approaches have been proposed in the scientific literature. Recent surveys on the topic can be found in [2], [3].

A major trend is composed of *dyadic topological approaches*. In an seminal work proposed by Liben-Nowell and al. in [1], authors have shown that simple topological measures representing relationships between pairs of unlinked nodes in a complex network, can be used for predicting formation of new links. Let's consider the case of applying *common neighbors* as a topological measure. Let $\mathcal{L}$ by the list of pairs of unlinked nodes (belonging to a same connected component). We have $\mathcal{L} = \{(x, y)\}$. Let $\Gamma(x)$ be the function returning the set of direct neighbors of node x in the graph. The common neighbors function of two nodes x, y is then simply defined by:

$$CN(x, y) = |\Gamma(x) \cap \Gamma(y)| \tag{1}$$

The list $\mathcal{L}$ is sorted according to the value obtained by applying the common neighbors function to each couple of unlinked nodes. The *top* k pairs of nodes are then returned as the output of the prediction task. The assumption here is that, the more a couple of unlinked nodes share common neighbors, the more they are likely to have a link in future. In [1] k is equal to the number of really appearing links. Other types of topological measures can be applied for the same purpose.

Following the work of Liben-Nowell et al.[1], many works have been published focusing on how to combine different topological metrics in order to enhance prediction performances. One widely applied approach is based on expressing the problem of link prediction as a problem of binary classification. The idea is compute a set of topological measures for each unlinked couple of nodes in $\mathcal{L}$. Then for each element in $\mathcal{L}$ associate one of the labels: `Linking` or `Not-linking` based on observing the status of the graph at a future step. The dataset hence computed (topological features with classes), can then be used to learn a model for discriminating the *linking* class from the *not-linking* one using classical supervised machine learning approaches [4], [5].

Surprisingly, none of the previous works, attempted to combine the prediction power of individual topological measures by applying computational social choice algorithms (or what is also known as rank aggregation methods) [6]. *Rank*

1082-3409/12 $26.00 © 2012 IEEE
DOI 10.1109/ICTAI.2012.111

aggregation can be defined as a process of combining a number of ranked lists or rankings of candidates or elements to get a single list and with least possible disagreement with the all the experts or voters who provide these lists. These methods were a part of social choice theory and were mostly applied to political and election related problems [7], [8], [9]. It was not until recent years that these methods found an application in outside world especially in web metasearch engines [10], [11], [12], multiple search, similarity search [13] etc. These techniques were designed to ensure fairness amongst experts while combining their rankings and hence all experts are given equal weights. Expressing the link prediction problem in terms of a vote is straightforward: candidates are examples (pairs of unconnected nodes), while voters are topological measures computed for these pairs of unlinked nodes. Then, we have a voting problem with a quite huge set of candidates and rather a reduced set of voters. These settings are very similar to those encountered when considering the problem of ranking documents in a meta-search engines where voting schemes has also been applied with success [12], [10], [14].

While in democratic elections it is definitely preferable to consider all voters equally, in our settings, prediction performances can be boosted by weighting differently the different applied topological measures (voters) in function of their individual performances in predicting new links. We propose here two different weighting scheme. Weights are used in two different weighted rank aggregation methods: The first one is based on the classical Borda count approach [7], while the second is based on the Kemeny aggregation rule. The later is known to compute the *Condorcet* winner of an election (if it exists): the candidate that wins each duel with all other candidates. We contribute in three ways: first we provide a way to generate weights for the topological measures; second, we propose a new way of introducing weights to approximate Kemeny aggregation; and third, we use supervised or weighted rank aggregation to link prediction task in complex networks. Our approach is evaluated in the context of a link prediction task applied to academic co-authorship networks. Experiments are conducted on real networks extracted from the now well known DBLP bibliographical server.

The reminder of this paper is organized as follows. Next in section II we describe briefly some related work applying rank aggregation in similar settings. Our approach is detailed in section III. Experiments are presented in section IV. Results are provided and commented in the same section. And in the end, a conclusion is given in section V.

II. Related Work

Following the work of Liben-Nowell and al. [1] many attempts were made to combine the effects of individual topological metrics in order to enhance the overall prediction performance of the approach. Most of these works involve the application of *Machine Learning* algorithms.

The first approach is the one proposed by Mohammad Al Hasan and al. in [4]. The authors propose to use supervised machine learning algorithms for the purpose of link prediction. They convert the problem of link prediction in graphs into a binary classification problem where examples are unconnected node pairs and are characterized by a vector of topological attribute values. They make a comparative analysis on the suitability of many learning algorithms to be used in link prediction based on their prediction performance. Another interesting study that they make is to rank the attributes based on various factors in order to compare and judge their relative strength in a prediction task.

The work presented by N. Benchettara and al. in [5] is a temporal approach for link prediction based on supervised machine learning where link prediction is done by using Decision tree algorithm with boosting. This is a dynamic approach where the evolution of the network is taken into account. Their work is mostly based on bipartite graphs and they introduce the concept of indirect topological measures computed using the projected graphs. They show how the use of indirect attributes greatly enhances the final prediction result.

Before describing the approaches based on *supervised rank aggregation* which refers to the same process of combining rankings but giving different weights to experts, we would like to describe in brief, two of the well known classical rank aggregation methods.

- **Borda's method** [7] is a truly positional method as it is based on the absolute positioning of the ranked elements rather than their relative rankings. A Borda's score is calculated for each element in the lists and based on these scores the elements are ranked in the aggregated list. For a set of full lists $L = [L_1, L_2, L_3, ..., L_n]$, the Borda's score for a element x and a list L_k is given by:

  $$B_{L_i}(x) = \{count(y) \mid L_i(y) < L_i(x) \;\&\; y \in L_i\} \quad (2)$$

  The total Borda's score for an element is given as:

  $$B(x) = \sum_{t=1}^{n} B_{L_i}(x) \quad (3)$$

  Borda's method is mostly applicable to full lists and is not very suitable for partial lists. For clarification, full lists refer to the input lists that have exactly the same elements but may be in different order whereas partial lists may have some but not all elements in common. In our case, we are only dealing with full lists.
- **Kemeny optimal aggregation** proposed in [11], makes use of Kendall Tau distance to find the optimal aggregation. Kendall Tau distance counts the number of pairs of elements that have opposite rankings in the two input lists i.e. it calculates the pairwise disagreements.

  $$K(L_1, L_2) = \mid (x, y) \; s.t. \; L_1(x) < L_2(y) \;\&\; L_1(x) > L_2(y) \mid \quad (4)$$

  The first step is to find a initial aggregation of input lists using any standard method. The second step is to find all possible permutations of the elements in the initial aggregation. For each permutation, a score is computed which is equal to the sum of distances between this permutation and the input lists. The permutation having the lowest score is considered as optimal solution. For example, for a collection of input rankings $\tau_1, \tau_2, \tau_3, ..., \tau_n$ and an aggregation π, the score is given by:

$$SK(\pi, \tau_1, \tau_2, \tau_3, ..., \tau_n) = \sum_{i \in n} K(\pi, \tau_i) \qquad (5)$$

The speciality of Kemeny optimal aggregation is that it complies with *Condorcet principle* [9] which is not the case with positional methods like Borda's algorithm. *Condorcet principle* states that if there exists an item that defeats every other item in simple pairwise majority voting, then it should be ranked above all other. Sometimes, it may happen that we do not have such a winner and sometimes, we may also have more than one winner. This is another research issue which is beyond the scope of our present work.

In spite of all advantages Kemeny optimal aggregation is computationally hard to implement. So while looking for an alternative solution that gives similar kind of aggregation but is computationally feasible, we are led to another approach named *Local kemenization* [11]. A full list π is locally Kemeny optimal aggregation of partial lists $\tau_1, \tau_2, \tau_3, ..., \tau_n$, if there is no full list π' that can be obtained from π by performing a single transposition of a single pair of adjacent elements and for which

$$SK(\pi', \tau_1, \tau_2, \tau_3, ..., \tau_n) < SK(\pi, \tau_1, \tau_2, \tau_3, ..., \tau_n)$$

In other words, it is impossible to reduce the total distance of an aggregation by flipping any adjacent pair of elements in the aggregation.

Looking into the work based on rank aggregation techniques, we can say that not much have been explored when it comes to application of rank aggregation in link prediction. Moreover these works apply mostly unsupervised rank aggregation algorithms giving equal weight to all the experts who provide the ranked lists.

One of the well known work is weighted majority algorithm proposed in [15] where the authors have proposed to use weights for predictors, all having equal weights in the beginning. There is a master predictor which makes the final prediction based on the class which corresponds to a maximum total weight of predictors. If the final prediction is wrong then weights of all predictors who disagreed with that label, is increased by a factor β such that $0 \leq \beta < 1$ and thus reducing the effect of unworthy predictors at each iteration. This approach has a limitation that the performance of the master predictor can be at most equal to the best performing predictor. On the contrary, the use of rank aggregation can provide even better prediction at times. This may be due the fact that, in these algorithms, the *"likes"* of majority of the predictors is given higher preference. At the same time, the *"dislikes"* are given least preference. So these algorithms are much more spam/noise resistant.

A significant work on supervised rank aggregation has been done in [16] where authors propose supervised aggregation by Markov chain to enhance the ranking result on meta-searches. However, it has been shown that Local Kemenization improves on Markov chain-based approaches [11].

Another very recent work is in [17] where the authors use supervised rank aggregation to find influential nodes and future links. Authors have proposed their own supervised Kemeny aggregation method based on quick sort and applied it to Twitter and citation networks. However, their method is mostly based on the topological features of nodes. Whereas our work is based on the features of a couple of nodes(edges) with a use of merge sort algorithm to find supervised local Kemeny aggregation. The reason why we use merge sort is that it is seemingly more stable than quick sort. Stability of a sorting algorithm is important when two candidates have equal importance. A stable sorting algorithm is the one which never affects the relative order of two candidates who are equal in ranks. This feature may be important when we have ties.

III. Supervised Rank Aggregation for Link Prediction

Let $G = < G_1, \ldots, G_n >$ be a temporal sequence of an evolving graph. The whole sequence is divided into three parts: *training*, *labeling* and *testing* or *validation*. Three graphs namely G_{learn}, G_{label} and G_{test} are generated by making union of the temporal sequences of the graphs for corresponding time slots. The training data is constructed as follows. An example will be generated for each couple of nodes (x, y) that are not linked in G_{learn} but both belonging to the same connected component. The class labeling is obtained by checking whether the couple of nodes is indeed connected in G_{label}. If such a connection exists then it will be a positive example in the supervised learning task and if no connection exists, it will be a negative example [5]. Thus, examples are generated from these graphs for both training and validation. These examples are also characterized by a given number of topological attributes.

Each attribute of an example has the capacity to provide some unique information about the data when considered individually. The training examples are ranked based on the attribute values. So, for each attribute we will get a ranked list of all examples. Considering only the *top* k ranked examples and with an assumption that when we rank the examples according to their attribute values, the positive examples should be ranked on the top, we compute the performance of each attribute. This performance is measured in terms of either *precision* (maximization of identification of positive examples) or *false positive rate* (minimization of identification of negative examples) or a combination of both. Based on the individual performances, a weight is assigned to each attribute.

For validation, we use examples obtained from the validation graph characterized by same attributes and try to rank all examples based on their attribute values. So for n different attributes we shall have n different rankings of the test examples. These ranked lists are then merged using a *supervised rank aggregation* method and the *weights of the attributes* obtained during learning process. The *top* k ranked examples in the aggregation are taken to be the predicted list of positive examples. Using this predicted list, we calculate the performance of our approach. k in this case, is equal to the number of positive examples in the validation graph.

A. Weight computation

We propose to compute voter's (topological measures) weights based on their capability to identify correct elements in top k positions of their rankings. Weights associated to applied topological measures are computed based on the following criteria :

- **Maximization of positive precision:** Based on maximization of identification of positive examples the attribute weight is calculated as

$$w_i = n * Precision_i \tag{6}$$

  where n is the total number of attributes and $Precision_i$ is the $precision$ of attribute a_i based on identification of positive examples. Just to remind, precision is defined as the fraction of retrieved instances that are relevant.

- **Minimization of false positive rate:** By minimizing the identification of negative examples we get a weight as below

$$w_i = n * (1 - FPR_i) \tag{7}$$

  where n is the total number of attributes and FPR_i is the $false\,positive\,rate$ of attribute a_i based on identification of negative examples. False positive rate is defined as the fraction of retrieved instances that are not relevant.

B. Supervised rank aggregation

First let's define some basic functions used later in defining weighted aggregation functions. Let L_i be a ranked list of n candidates (a vote). $L_i(x)$ denotes the rank of element x in the list L_i. The top ranked element has the rank 0. The basic individual Borda score of an element x for a voter i is then given by:

$$B_i(x) = n - L_i(x)$$

Let x and y be two candidates. We define the local preference function as follows :

$$Pref_i(x, y) = \begin{cases} 1 & \text{if } B_i(x) > B_i(y) \\ 0 & \text{if } B_i(x) < B_i(y) \end{cases} \tag{8}$$

Introducing weights in Borda aggregation rule is rather straightforward: Let $(w_1, w_2, \dots, w_r)$ be the weights for r voters providing r ranked lists on n candidates. The weighted Borda score for a candidate x is then given by:

$$B(x) = \sum_{i=1}^{r} w_i * B_i(x) \tag{9}$$

For approximate Kemeny aggregation [11], we introduce weights into the definition of the non-transitive preference relationships between candidates. This is modified as follows. Let w_T be the sum of all computed weights i.e. $w_T = \sum_{i=1}^{r} w_i$. For each couple of candidates x, y we compute a score function as follows:

$$score(x, y) = \sum_{i=1}^{r} w_i * Pref_i(x, y)$$

The weighted preference relation($\succ_w$) is then defined as follows :

$$x \succ_w y : score(x, y) > \frac{w_T}{2}$$

This new preference relation is used to sort an initial aggregation of candidates in order to obtain a supervised Kemeny aggregation. The initial aggregation can be any of the input lists or an aggregation obtained by applying any other classical aggregation method like Borda. In our algorithm, we have applied merge-sort for the time being.

IV. Experiment

We evaluated our approach using data obtained from DBLP [1] databases. DBLP is a scientific bibliography website containing a large database of articles mostly related to computer science. Our network consists of authors and their publications providing us a bipartite structure of graph. The data used corresponds to a time span of 1970 to 1976. This data is divided into two datasets containing information for different years, each having a training set and a test or validation set. We generate examples for each dataset. Each example is a pair of unconnected nodes, represented by a vector of topological measures and a class label. Tables I and II provide information about the training graphs while table III summarizes information about the examples generated.

Datasets	**Training Time**	**Graph**		
		Authors	**Publications**	**Edges**
Dataset1	[1970,1973]	2661	1487	6634
Dataset2	[1972,1975]	4536	2542	10855

TABLE I
AUTHOR-PUBLICATION BIPARTITE GRAPHS

Datasets	**Author Graph**		**Publication Graph**	
	Nodes	**Edges**	**Nodes**	**Edges**
Dataset1	2661	2575	1487	1520
Dataset2	4536	4510	2542	2813

TABLE II
PROJECTED GRAPHS

The attributes characterizing each example in each dataset are:

- **Neighborhood-based attributes:**
  - Common neighbors: $CN(x, y)$)=$\parallel \Gamma(x) \cap \Gamma(y) \parallel$
  - Jaccard's coefficient: $JC(x, y)$)=$\frac{\|\Gamma(x) \cap \Gamma(y)\|}{\|\Gamma(x) \cup \Gamma(y)\|}$
  - Adamic Adar: $AD(x, y)$=$\sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{log\|\Gamma(z)\|}$ [18]

[1] http://www.dblp.org

Datasets	**Training Time**	**Validation Time**	**Training examples**		**Test examples**	
			Pos	**Neg**	**Pos**	**Neg**
Dataset1	[1970,1973]	[1971,1974]	30	1663	41	3430
Dataset2	[1972,1975]	[1973,1976]	87	19245	82	18675

TABLE III
EXAMPLES GENERATED FROM AUTHOR GRAPHS

  - Preferential attachment: $PA(x,y)$=$\parallel \Gamma(x) \times \Gamma(y) \parallel$ [19]
- **Distance-based attributes:**
  - Shortest path distance(Dis)
  - Katz: $Katz(x,y) = \Sigma_{l=1}^{\infty} \beta^{\ell} \times \parallel path_{x,y}^{(\ell)} \parallel$, where $path_{x,y}^{(\ell)}$ is the number of paths between x and y of length ℓ and β is a positive parameter which favours shortest paths [20]
  - Maximum forest algorithm (MFA): It makes use of *Laplacian* matrix and identity matrices of the graph to find distance between all the nodes of a graph. A matrix is obtained as $M = (I + M_L)^{-1}$ where I is an identity matrix and M_L is the *Laplacian* matrix of the graph being studied. So, $MFA(x,y) = M(x,y)$ [21].
- **Centrality-based attributes:**
  - Product of PageRank (PPR)[22]
  - Product of clustering coefficient (PCF)

These attributes can be computed directly from the bipartite graph and/or also from the projected graphs. Projected graph refers to the unimodal graphs obtained by projecting the bipartite graph over one of its node sets [5]. The attributes computed from projected graphs are called indirect attributes.

Before using the rank aggregation method, we ranked all the test examples by their attribute values. Considering only the *top* k examples and taking the number of positive examples in each dataset as the corresponding value of k, we compute the performance of each attribute in identifying the real positive links in *top* k positions. Table IV summarizes this information.

In the first part of experimentation, we applied our approach to the complete datasets. For rank aggregation, we have used supervised Borda and supervised Kemeny methods. We also try to compare our approach with link prediction approaches using basic machine learning algorithms like Decision tree, Naive bayes etc. We name our approaches as Supervised Borda 1 and Supervised Borda 2 based on how the attribute weights are computed. 1 represents weights computed based on maximization of positive precision and 2 represents weights being computed based on minimization of false positive rates. We will follow the same convention to represent supervised Kemeny. Figure.1 shows the results obtained on the complete datasets in terms of precision and recall.

For both datasets, our approach of supervised Kemeny based on maximization of positive precision, gives a comparatively better performance in terms of precision as compared to most of the other methods.

In the second part of experimentation, we try to evaluate the performance of our approach for the task of recommendation. So we vary the value of k used during validation to observe the performance on dataset-1. We vary k from 5 to total number of positive examples in the test set of dataset-1 i.e. 41. Figures.2 and 3 show the performances of our rank aggregation based approaches in terms of precision and recall.

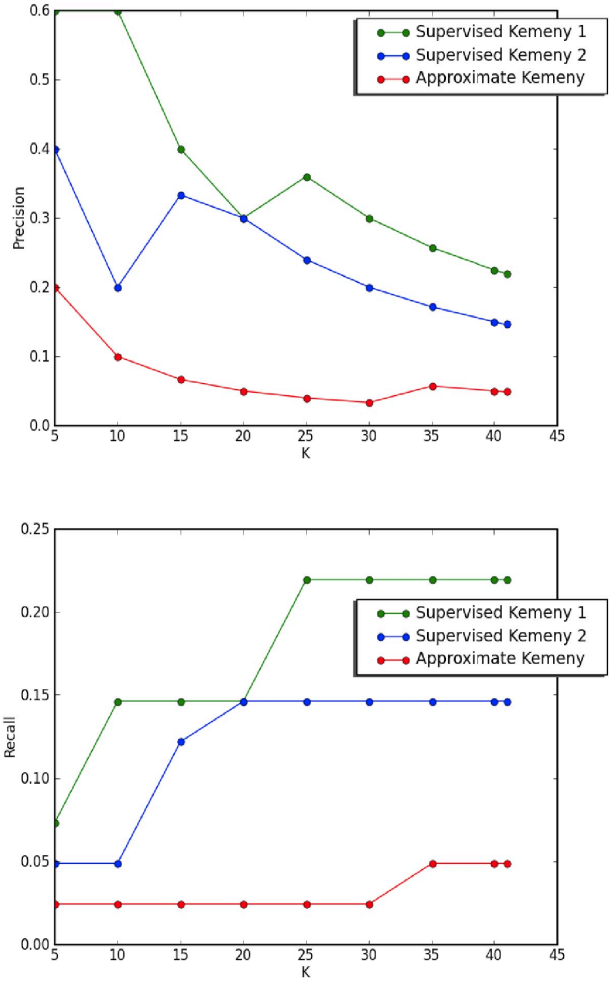

Fig. 2. Effect on performance of supervised Kemeny by varying k on Dataset-1

Attributes	Dataset1	Dataset2
Katz	0	0
MFA	0.0244	0.0732
PPR	0.0244	0.0244
PCF	0.0732	0.0244
PCD	0	0
VC	0.5122	0.4268
JC	0.2195	0.1707
AD	0.1463	0.1463
AP	0.0488	0
Dis	0	0.0122
Indirect Katz	0.1220	0.1098
Indirect MFA	0.0488	0.0732
Indirect PPR	0.0488	0
Indirect PCF	0.4878	0.4756
Indirect PCD	0.0244	0.0122
Indirect VC	0.0488	0.0488
Indirect JC	0.0488	0.1098
Indirect AD	0.0976	0.0488
Indirect AP	0.0244	0
Indirect Dis	0.6098	0.5366

TABLE IV
RESULTS (PRECISION) OBTAINED BY RANKING THE TEST EXAMPLES BY ATTRIBUTE VALUES

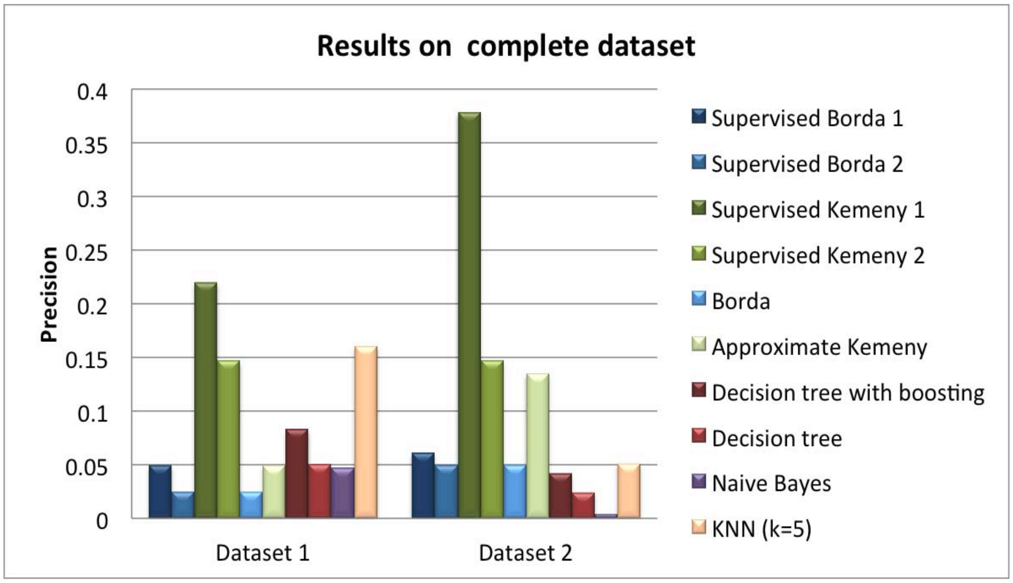

Fig. 1. Precision for complete test set by learning on complete datasets

As observed from figures 2 and 3, the precision decreases while the recall increases with increasing k. The performance of supervised Kemeny is much better than that of supervised Borda, which is quite evident due to the fact that Kemeny based approaches are quite efficient in dealing with noise or spam in data. The precision of supervised and approximate Kemeny approaches at lower values of k are good enough to make them suitable for recommendation task.

V. CONCLUSION

This paper presents our research work on finding a new solution to link prediction problem in dynamic large graphs representing complex networks. We propose a novel approach based on supervised rank aggregation and is motivated by the belief that each attribute can provide us with some unique information which can be aggregated in the end to make a better prediction of association between two unconnected entities in a network. First we have come up with a new way of introducing weights in a well known rank aggregation method. And secondly, we have proposed to apply this approach for the purpose of link prediction in complex networks. We evaluated our approach on a co-authorship network obtained from DBLP database. The experimental results are quite encouraging as our method seems to perform better than the approaches using classical machine learning algorithms for link prediction. We also tested performance of the approach by varying the value

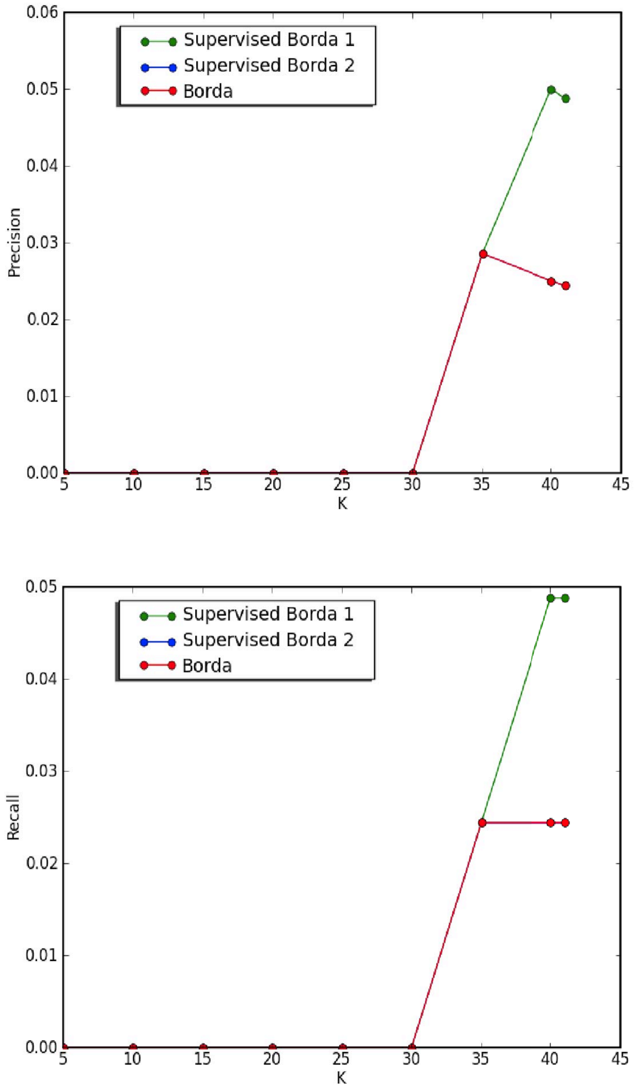

Fig. 3. Effect on performance of supervised Borda by varying k on Dataset-1

of k for validation. This experiment was done in order to see if the method can perform well with very small values of k which is a primary requirement for recommendation task. In future we intend to apply our method on other scientific collaboration networks to futher verify the prerformance of our approach. While working in complex networks, it is extremely important to take care of the complexity of the algorithms one is working on. In our case we have made a big attempt of trying to apply rank aggregation algorithms which have their own limitations in terms of time complexity. In order to avoid the limitations of Kemeny optimal aggregation(which is NP-hard), we have focused on the concept of approximate Kemeny aggregation. And the use of merge sort have reduced the complexity to $O(rn \log n)$ where r is the number of experts and n is the number of candidates in each input ranked list. In order to enhance the performance of our approach and reduce the complexity further, we are looking forward to use the concept of $top\ k$ aggregation, where instead of aggregating the complete input lists, only the $top\ k$ from each are considered for aggregation.

References

[1] D. Liben-Nowell and J. Kleinberg, "The link prediction problem for social networks," in *Proceedings of the twelfth international conference on Information and knowledge management*, ser. CIKM '03. New York, NY, USA: ACM, 2003, pp. 556–559. [Online]. Available: http://doi.acm.org/10.1145/956863.956972

[2] L. Lü and T. Zhou, "Link prediction in complex networks: A survey," *Physica A: Statistical Mechanics and its Applications*, vol. 390, no. 6, pp. 1150–1170, Mar. 2011. [Online]. Available: http://dx.doi.org/10.1016/j.physa.2010.11.027

[3] A. H. Mohammad and Z. M. J., "A survey of link prediction in social networks," in *Social network Data Analysis*, C. C. Aggarwal, Ed. Springer, 2010, ch. 9, pp. 243–275.

[4] M. A. Hasan, V. Chaoji, S. Salem, and M. Zaki, "Link prediction using supervised learning," in *In Proc. of SDM 06 workshop on Link Analysis, Counterterrorism and Security*, 2006.

[5] N. Benchettara, R. Kanawati, and C. Rouveirol, "Apprentissage supervisé pour la prédiction de nouveaux liens dans des réseaux sociaux bipartie," in *Actes da la 17iéme Rencontre de la société francophone de classification (SFC'2010)*, St. Denis, La réunion, Jun. 2010, pp. 63–66.

[6] Y. Chevaleyre, U. Endriss, J. Lang, and N. Maudet, "A short introduction to computational social choice," *SOFSEM 2007: Theory and Practice of Computer Science*, pp. 51–69, 2007. [Online]. Available: http://www.springerlink.com/index/768446470RPLJ120.pdf

[7] J. de Borda, "Memoire sur les elections au scrutin," 1781.

[8] D. Black, R. Newing, I. McLean, A. McMillan, and B. Monroe, *The Theory of Committees and Elections by Duncan Black, and Revised Second Editions Committee Decisions with Complementary Valuation by Duncan Black*, 2nd ed. Kluwer Academic Publishing.

[9] H. Young and A. Levenglick, "A consistent extension of condorcet's election principle," *SIAM Journal on Applied Mathematics*, vol. 35, no. 2, 1978.

[10] J. A. Aslam and M. Montague, "Models for metasearch," in *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, ser. SIGIR '01. New York, NY, USA: ACM, 2001, pp. 276–284. [Online]. Available: http://doi.acm.org/10.1145/383952.384007

[11] C.Dwork, R.Kumar, M.Naor, and D.Sivakumar, "Rank aggregation methods for web," in *Proceedings of the 10th international conference on World Wide Web*, ser. WWW '01. Hong Kong: ACM, 2001, pp. 613–622. [Online]. Available: http://doi.acm.org/10.1145/371920.372165

[12] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar, "Rank aggregation, spam resistance, and social choice," in *WWW '01: Proceedings of 10th international conference on World Wide Web*, 2001, pp. 613–622.

[13] R. Fagin, R. Kumar, and D. Sivakumar, "Efficient similarity search and classification via rank aggregation," in *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, ser. SIGMOD '03. New York, NY, USA: ACM, 2003, pp. 301–312. [Online]. Available: http://doi.acm.org/10.1145/872757.872795

[14] M. Montague and J. A. Aslam, "Condorcet fusion for improved retrieval," in *Proceedings of the eleventh international conference on Information and knowledge management*, ser. CIKM '02. New York, NY, USA: ACM, 2002, pp. 538–548. [Online]. Available: http://doi.acm.org/10.1145/584792.584881

[15] N. Littlestone and M. K. Warmuth, "Weighted majority algorithm," *IEEE Symposium on Foundations of Computer Science*, 1989.

[16] Y.-T. Liu, T.-Y. Liu, T. Qin, Z.-M. Ma, and H. Li, "Supervised rank aggregation," in *Proceedings of the 16th international conference on World Wide Web*, ser. WWW '07. New York, NY, USA: ACM, 2007, pp. 481–490. [Online]. Available: http://doi.acm.org/10.1145/1242572.1242638

[17] K. Subbian and P. Melville, "Supervised rank aggregation for predicting influence in networks," in *In the proceedings of the IEEE Conference on Social Computing (SocialCom-2011).*, Boston, October 2011.

[18] L. A. Adamic, O. Buyukkokten, and E. Adar, “A social network caught in web,” in *First Monday*, 8, Ed., no. 6, June 2003.

[19] Z. Huang, X. Li, and H. Chen, “Link prediction approach to collaborative filtering,” in *JCDL*, M. Marlino, T. Sumner, and F. M. S. III, Eds. ACM, 2005, pp. 141–142.

[20] L. Katz., “A new status index derived from socimetric analysis.” *Psychmetrika, 18(1)*, vol. 18, no. 1, pp. 39–43, 1953.

[21] F. Fouss, A. Pirotte, J.-M. Renders, and M. Sarens, “Random-Walk Computation of Similarities between Nodes of a Graph with Application to Collaborative Recommandation,” *IEEE Transactions on knowledge and data engineering*, vol. 19, no. 3, pp. 355–369, 2007.

[22] S. Brin and L. Page, “The anatomy of a large scale hypertextual web search,” in *proceedings of seventh International conference on the world wide web*, 1998.