NETWORKS AND HETEROGENEOUS MEDIA
©American Institute of Mathematical Sciences
Volume **10**, Number **1**, March **2015**
doi:10.3934/nhm.2015.10.17
pp. **17–35**

LINK PREDICTION IN MULTIPLEX NETWORKS

Manisha Pujari and Rushed Kanawati

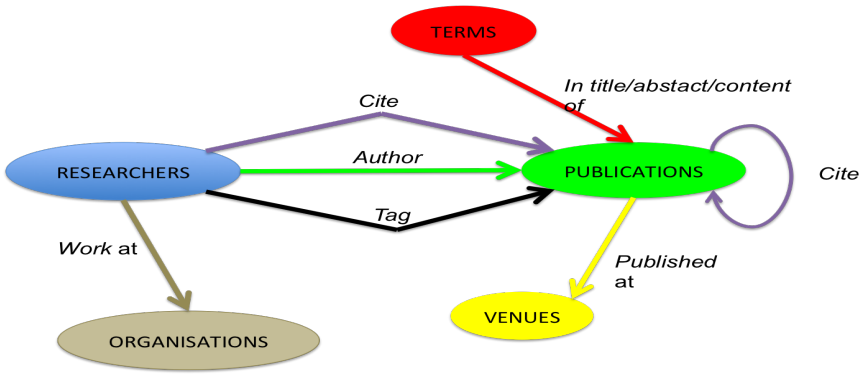
SPC, UP13, LIPN, CNRS UMR 7030
99 Av. J.B. Clément, 93430 Villetaneuse, France

Abstract. In this work we present a new approach for co-authorship link prediction based on leveraging information contained in general bibliographical multiplex networks. A multiplex network is a graph defined over a set of nodes linked by different types of relations. For instance, the multiplex network we are studying here is defined as follows : nodes represent authors and links can be one of the following types: co-authorship links, co-venue attending links and co-citing links. A supervised-machine learning based link prediction approach is applied. A link formation model is learned based on a set of topological attributes describing both positive and negative examples. While such an approach has been successfully applied in the context on simple networks, different options can be applied to extend it to multiplex networks. One option is to compute topological attributes in each layer of the multiplex. Another one is to compute directly new multiplex-based attributes quantifying the multiplex nature of dyads (potential links). These different approaches are studied and compared through experiments on real datasets extracted from the bibliographical database DBLP.

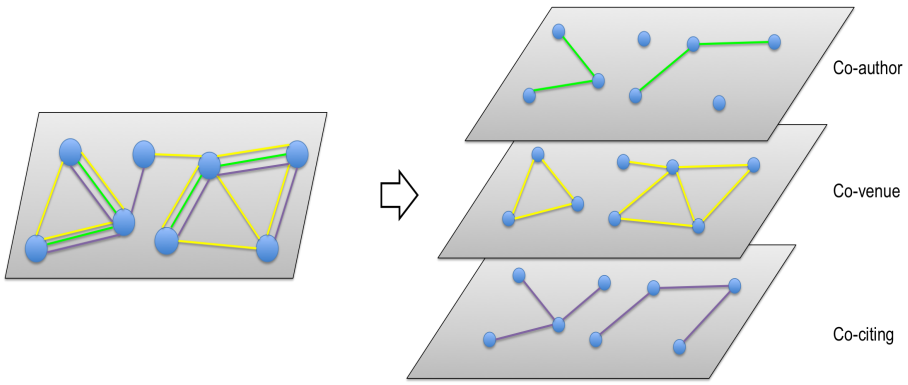
1. **Introduction.** Analyzing dynamic large-scale networks is a major emerging topic in different research areas. Actually, many real-world systems can be modeled as an evolving network of interacting *actors*. This is namely the case of on-line social networks, collaboration networks (such as academic 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* [28, 34]. 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 [34, 36]. Most of existing works consider only simple networks where all links are of the same type. However, real networks are often heterogeneous. They involve different types of links and nodes. For example, Figure 1(a) shows a diagrammatic representation of a scientific collaboration network. Different types of *actors* can be distinguished: researchers, publications, venues, terms ..., etc. Focusing on interactions among

2010 *Mathematics Subject Classification.* Primary: 58F15, 58F17; Secondary: 53C35.
Key words and phrases. Complex network analysis, multiplex network, link prediction, rank aggregation, bibliographical networks.

researchers, different types of links can be defined: researchers can be linked if they have co-published some articles or if they have published their articles in the same conferences or the domain of their research are the same. They can also be linked if they have referred to same works in their articles. Authors or researchers network can be better modeled by a multiplex network [9, 11] (see figure 1 b). We define a multiplex as a multi-layer network, where each layer contains the same set of nodes but a different type of links.



(a) Scientific collaboration networks



(b) Multiplex layers in author (researchers) network

FIGURE 1. Multiplex structure in a scientific collaboration network

In this work, we explore approaches for leveraging information mined from different layers in a multiplex in order to inform the link formation in one specific layer. For instance, in a bibliographical network, we search to inform the formation of co-authorship links by mining different types of interactions among researchers (co-venue, co-citing, etc.).

Various link prediction approaches have been proposed in recent years [34, 36, 40]. Few has dealt with heterogeneous network [3, 16, 27, 14, 26]. Link prediction approaches can be classified as *node-features based approaches* or *topological approaches* based on whether they use node-features or only structural information of the graph for prediction. In node-features based approaches, apart from the structure of graph we also have some extra information regarding the properties or

characteristics of nodes. These extra information can be helpful in predicting links when the nodes are very sparsely connected in the graph. One such approach is local probabilistic model proposed by C. Wang and al. [43]. Topological approaches refer to those which involve only exploitation of graph structure of the network. They compute scores for pairs of unconnected nodes based on only the graphical features of the network structure and without any extra information about the features of nodes. They observe mainly how the connections have been established between of nodes and how they change over time. Based on former they try to predict a missing link or based on the later they predict a new link.

Node features are very useful when the network graph is very sparsely connected and not much can be learned from graph topology. Whereas topological approaches are very efficient in the absence of content of feature information. Both have their own utility and at times a combination of both can come out to give a very good predictor. These kind of approaches can be termed as *hybrid* approaches. The topological (graph based) link prediction approaches can be further categorized as *temporal* or *non-temporal / static* based on the fact that whether they take into account the dynamic aspect of the network or not. Another way to classify them is as *dyadic* or *structural* approaches, based on the way of score computations. They can also be classified as *supervised* or *unsupervised*: Supervised approaches generate a model using many topological scores for unlinked node pairs to predict links whereas unsupervised approaches use a single type of score for the node pairs and mostly use ranking to predict new links.

Next in section 2 we introduce basic notations and definitions used in this paper. In section 3, we give brief account of the classical link prediction methods, focusing mainly on dyadic topological approaches. In section 4 we introduce our approach based on supervised rank aggregation. In section 5 we present our new approach of link prediction in multiplex networks using multiplex link information. Finally, we conclude in section 6.

2. Formal definition and notations.

A network $\mathcal{N}$ can be modeled as a graph $G =< V, E >$ where V is the set of nodes or vertices and E is the set of edges present in the graph. A multiplex network is defined as a multi-layer graph:

$$G =< V, E_1, \ldots, E_\alpha : E_k \subseteq V \times V \ \forall k \in \{1, \ldots, \alpha\} >$$

where V is a set of nodes and E_k is a set of edges of type k. α denotes the number of slices in the multiplex. We introduce some new notations that will be used next in this paper.

- $A^{[k]}$ is the adjacency matrix of slice k
- $n = |V|$ is the number of nodes in the multiplex.
- $m_k = |E_k|$ is the number of edges in slice k
- $\Gamma(v)^{[k]} = \{x \in V : (x, v) \in E_k\}$ denotes the neighbors of v in slice k
- $d_v^k =\| \Gamma(v)^{[k]} \|$ is the degree of node v in slice k

In the case of a simple network (one layer) we simply omit the superscript k. Let $G =< G_0, G_1, ..., G_n >$ be a temporal sequence of networks. The goal of a link prediction approach is to predict the structure of graph G_{n+1}. In other words, here we try to find pairs (u, v) such that $u, v \in V$ and $(u, v) \notin E$ where $V = \bigcup_{i=0}^{n} V_i$ and $E = \bigcup_{i=0}^{n} E_i$.

In machine learning terms, the unlinked pairs of nodes are called *examples* or *instances*. If the time aspect of the network are to be considered also, then the

examples can be generated as follows. Let $G =< G_1, \ldots, G_n >$ be a temporal sequence of an evolving graphs. The whole sequence is divided into two parts: *training* and *testing*. Each part is then again divided into two phases one for generation of examples and another for labeling those examples. Thus, for example, in training we shall have a *learning* and *labeling* phases resulting in graphs namely G_{learn} and G_{label} generated by making union of the temporal sequences of the graphs for three corresponding time slots. The training data is constructed as follows. An example will be a couple of nodes (x, y) that are not linked in G_{learn} but both belonging to the same connected component. The class 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 [7]. Thus, examples are generated from these graphs for both training and testing. These examples are also characterized by a given number of topological attributes computed on learning (or test) graphs. Figure 2 and 3 illustrate the process diagrammatically.

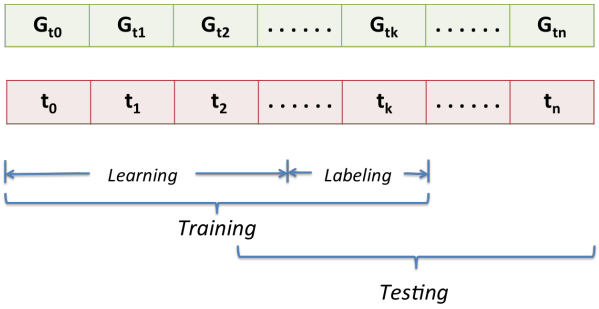


FIGURE 2. Generation of examples

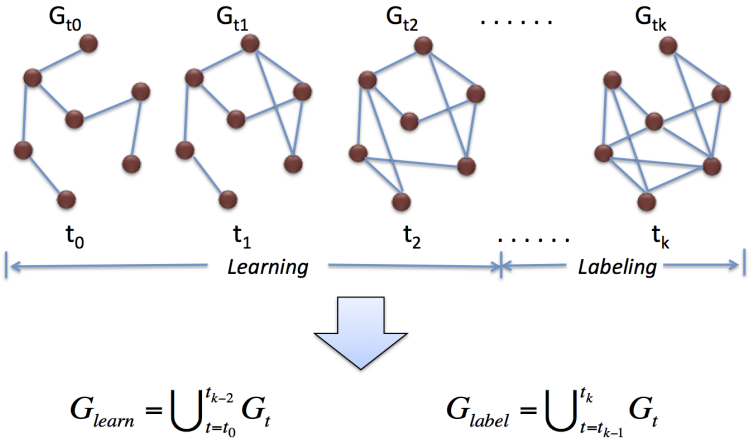


FIGURE 3. Construction of learning and labeling graphs

3. **Link prediction approaches.** The basic and most simple approach for predicting links using network graphs is to compute similarity scores for the unlinked node pairs and based on this score decide the presence or appearance of a link between them. In scientific literature we find many ways of computing this score. They can be *neighborhood-based*, *distance-based* or *an aggregation of node properties*. These approaches are mostly unsupervised. Below we list few of the important methods that have been used for link prediction. For sake of simplicity, we define hereafter basic topological scores for the case of simple (one layer) network. Extensions of these scores to multiplex networks is discussed in section 5.

Neighborhood based features.

***Common neighbors*:** Common neighbors counts the number of nodes (i.e. neighbors) that are connected to both the nodes under observation. Newman used this quantity for studying collaboration networks [38], while Kossinets used it while analysing large-scale social networks [25].

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

***Jaccard coefficient*:** Jaccard coefficient calculates the ratio of number of common neighbors to that of the total number of neighbors of the two nodes [23].

$$JC(x,y) = \frac{\mid \Gamma(x) \cap \Gamma(y) \mid}{\mid \Gamma(x) \cup \Gamma(y) \mid} \tag{2}$$

***Adamic Adar coefficient*:** This metric proposes to weight the common neighbours based on their connectivity while computing the score. It gives more weight to less connected neighbours increasing their contribution in the score [29].

$$AA(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{log \mid \Gamma(z) \mid} \tag{3}$$

It is based on the coefficient proposed by L. Adamic and E. Adar to find similarity between two web pages [1]. For two web pages x and y, sharing a set of features z, this coefficient is computed as $\sum_z \frac{1}{log(frequency(z))}$. Where z is shared feature between x and y.

***Resource allocation*:** This metric is based on resource allocation dynamics on complex networks [39]. Like Adamic Adar coefficient, this index also depresses the contribution of high-degree common neighbours.

$$RA(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\mid \Gamma(z) \mid} \tag{4}$$

Path based features.

***Shortest path length*:** Number of edges in the shortest path between x and y in G. It is also known as the distance between nodes. More is the distance, lesser is the similarity between the nodes and lesser is the chance of having an link between them. This metric captures the fact that that the path between two nodes in a social network can effect the formation of a link between them following the fact that friend of a friend can be a friend in a social network.

***Katz's index*:** This index has been proposed initial in [24]. It is based on paths between nodes in a graph. It sums over a collection of paths and is exponentially damped by length to give shorter paths more weights. Mathematically

it is defined as,

$$Katz(x,y) = \sum_{l=1}^{\infty} \beta^{\ell} \times \mid path_{x,y}^{(\ell)} \mid \tag{5}$$

where $path_{x,y}^{(\ell)}$ is the number of paths between x and y of length ℓ and β is a positive parameter (i.e. damping factor) which favours shortest paths. The same can be presented using adjacency matrix

$$Katz(x,y) = \beta A_{xy} + \beta^2 (A^2)_{xy} + \beta^3 (A^3)_{xy} + \dots \tag{6}$$

A_{xy} is the adjacency matrix where the values are either 1 or 0 based on whether x and y are directly connected. $A^2)_{xy}$ is the matrix showing numbers of paths of length 2 between x and$y$ and so on. A very small β leads to a score close to number of common neighbors because long paths contribute very little. So the matrix showing Katz score between all pairs of nodes can be found as

$$K = (I - \beta A)^{-1} - I \tag{7}$$

β must be lower than the reciprocal of the largest eigenvalue of matrix A to ensure the convergence of above given equation [34].

***Matrix forest index*:** Matrix forest index computes the similarity between two nodes as the ratio of number of spanning rooted forests such that the two nodes belong to the same tree rooted at one of the nodes all the spanning rooted forests of the network. It can be computed as $M = (I - L)^{-1}$, I being the identity matrix and $L = D - A$ is the Laplacian matrix of the network where D is the degree matrix and A is the adjacency matrix [12]. This index was used for collaborative recommendation task in the work of F. Fouss et al. [19].

***Hitting time and commute time*:** Hitting time is a random walks based feature that counts the time required by a random walker to go from node x to node y in a graph. It is defined as the expected number of steps required for a random walker to walk from one node to the other. Shorter hitting time may denote the nodes are similar and can have higher chance of linking in future. As this metric is not symmetrical, often for undirected graphs, average commute time is used instead. If $HT(x,y)$ is the hitting time to reach node y from node x, average commute time is given by

$$CT(x,y) = HT(x,y) + HT(y,x) \tag{8}$$

A negated value of hitting or commute time can be used as a score for predicting links.

Rooted Pagerank: Pagerank denotes the importance of a node x by summing up the importance of all other nodes linked to x. This importance can also be represented by stationary distribution weight of a node. This feature can be altered to find a similarity score between two nodes and is termed as *rooted pagerank* in [29]. The similarity between two nodes x and$y$ is measured as the stationary probability of y in a random walk that returns to x with probability $1 - \alpha$ in each step, moving to a random neighbor with probability α. Rooted pagerank for all node pairs can be computed as follows.

$$RPR = (1 - \alpha)(I - \alpha N)^{-1} \tag{9}$$

where D is the diagonal degree matrix and $N = DA^{-1}$ is ajacency matrix with row sums normalized to 1.

***PropFlow*:** PropFlow captures the probability that a restricted random walk starting from one node x ends at another node y in l or less step using link weights as the transition probabilities. The restriction is that a walk terminates on reaching y or on revisiting any node including x. The walk selects links based on their weights which produces a score to estimate likelihood of new links. This measure is a more localized measure of propagation and is insensitive to topological noise far from the source node [30].

Aggregation of node features.

***Preferential attachment*:** Preferential attachment combines the degrees of the two concerned nodes and can be used as a score for predicting links. Here the probability of appearance of a new link is directly proportional to the degree of the observed nodes [5].

$$PA(x, y) =\mid k_x \times k_y \mid \tag{10}$$

For a simple un-directed and un-weighted graph the degree of a node is equal to the number of neighbors i.e. $k_x = \Gamma(x)$.

***Sum of neighbors*:** In the work of [21], the authors have used sum of neighbors as a topological feature for characterizing an unlinked node pairs. Formally, it can be defined as $\Gamma(x) + \Gamma(y)$

***Aggregation of clustering coefficients*:** Clustering coefficient of a node quantifies the probability of the neighbors of the node to get connected to each other.

$$cf(x) = \frac{3 \times \#Triangles\ adjacent\ to\ x}{\#Possible\ triples\ adjacent\ to\ x} \tag{11}$$

This property can also be used for link prediction by taking an aggregation (sum or product) of the clustering coefficients of two unconnected nodes. So the similarity score for any two nodes x and y will be

$$CC(x, y) = cf(x) \times cf(y) \quad or \quad CC(x, y) = cf(x) + cf(y) \tag{12}$$

In a seminal work proposed in [28], 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 same connected component). We have $\mathcal{L} = \{(x, y)\}$. Let $\Gamma(x)$ be the function returning a set of direct neighbors of node x in the graph. The common neighbors function of two nodes x, y is then defined by:

$$CN(x, y) = |\Gamma(x) \cap \Gamma(y)| \tag{13}$$

The list $\mathcal{L}$ is sorted according to the values obtained by applying the common neighbors function to couples of unlinked nodes. The top k couples 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 [28] k is equal to the number of really appearing links. Other types of topological measures can be applied for the same purpose.

Many other 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 to compute for each unlinked couple of nodes in $\mathcal{L}$, a set of topological measures. Then with each element in $\mathcal{L}$, associate one of

the labels: `Linking` (positive) or `Not-linking` (negative) based on 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 [21, 7].

Another main category is matrix based approches. In the work presented by A.K. Menon and al. [35], the authors use supervised matrix factorization approach for link prediction. The model learns latent features from the structure of a graph. The authors show that combining these latent features with explicit node features and also with outputs of other models to make better prediction. They propose a new approach to deal with class imbalance problem by directly optimizing a ranking loss. The model is optimized with stochastic gradient descent and also scales to large graphs. Another work on temporal link prediction given in [20] is a model based on matrix factorization. Authors exploit multiple information sources in the network to predict link occurrence probabilities as a function of time. They propose a unique model combining global network structure, content information of nodes and local proximity information. For combining the temporal information of the network, they use a weighted exponentially decaying model to build an aggregate weighted link matrix over a set of T time slices.

Other approaches include Probabilistic models, Stochastic block models, Heirarchical models etc. A more detailed survey on link prediction and approches can be found in [36] and [34].

4. **Supervised rank aggregation based link prediction.** None of the previous work, attempt to combine the prediction power of individual topological measures by applying computational social choice algorithms (or what is also known as rank aggregation methods) [13]. *Rank 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 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 [15, 10, 44]. These techniques were designed to ensure fairness among 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 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 [17, 4, 37].

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 [15], 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.

Before describing the approaches based on *supervised rank aggregation* which refers to the same process of combining rankings but giving different weights to

experts and these weights are learned in a due process of training, here is a brief description about two of the well known classical rank aggregation methods.

- **Borda's method**[15] is a truly positional method as it is based on the absolute positioning of the ranked elements rather than their relative rankings. A Borda score is calculated for each element in the lists and based on this score 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:

$$B_{L_i}(x) = \{count(y)|L_i(y) < L_i(x) \& y \in L_i\} \tag{14}$$

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

$$B(x) = \sum_{t=1}^{n} B_{L_i}(x) \tag{15}$$

  Borda's method is mostly applicable to full lists and is not very suitable for partial lists.
- **Kemeny optimal aggregation** proposed in [17], 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.

$$K(L_1, L_2) =| \ (x, y) \ s.t. \ L_1(x) < L_2(y) \ \& \ L_1(x) > L_2(y) \ | \tag{16}$$

  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:

$$SK(\pi, \tau_1, \tau_2, \tau_3, ..., \tau_n) = \sum_{i \in n} K(\pi, \tau_i) \tag{17}$$

  The speciality of Kemeny optimal aggregation is that it complies with *Condorcet principle* which is not the case with positional methods like Borda's algorithm. *Condorcet principle* [44] 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.

  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* [17]. 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 work 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 [32] 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 weights 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 [33] 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 [17].

Another very recent work is in [41] where the authors use supervised rank aggregation to find influential nodes and future links. Authors propose 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. Where as 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.

In the next part (subsection 4.1), is the description about our work on link prediction using rank aggregation. 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.

4.1. **Link prediction by supervised rank aggregation .** 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.

4.1.1. *Weights computation.* We propose to compute voters (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

$$w_i = n * Precision_i \tag{18}$$

  where n is the total number of attributes and $Precision_i$ is the *precision* of attribute 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

$$w_i = n * (1 - FPR_i) \tag{19}$$

  where n is the total number of attributes and FPR_{a_i} is the $false\ positive\ rate$ of attribute a_i based on identification of negative examples examples. False positive rate is defined as the fraction of non-relevant instances that are retrieved as relevant.

4.1.2. *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 :

$$Pref_i(x, y) = \begin{cases} 1 & \text{if } B_i(x) > B_i(y) \\ 0 & \text{if } B_i(x) < B_i(y) \end{cases} \tag{20}$$

Introducing weights in Borda aggregation rule is rather straightforward: Let $(w_1, w_2, \ldots, w_r)$ 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:

$$B(x) = \sum_{i=1}^{r} w_i * B_i(x) \tag{21}$$

For approximate Kemeny aggregation[17] 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:

$$score(x, y) = \sum_{i=1}^{r} w_i * Pref_i(x, y) \tag{22}$$

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

$$x \succ_w y : score(x, y) > \frac{w_T}{2} \tag{23}$$

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.

4.2. **Experimentation.** 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 as nodes and they are linked if they have co-published at least one paper during the observed period of time. The data corresponds to year between 1970-1979. We create three datasets out of that. Following the procedure described in the previously, we generate examples for each dataset. Table 1 provides information about the training graphs while table 2 summarizes information about the examples generated.

Years	**Properties**	**Co-Author**
1970-1973	*Nodes*	91
	Edges	116
	Density	0.028327
1972-1975	*Nodes*	221
	Edges	319
	Density	0.013122
1974-1977	*Nodes*	323
	Edges	451
	Density	0.008673

TABLE 1. Basic statistics about the co-authorship networks extracted from DBLP database

Years		**# Positive**	**# Negatives**
Train/Test	**Labeling**		
1970-1973	1974-1975	16	1810
1972-1975	1976-1977	49	12141
1974-1977	1978-1979	93	26223

TABLE 2. Examples from co-authorship graph

We applied our approach to the complete datasets. For rank aggregation, we have used supervised Borda and supervised Kemeny methods. We compare our approach with link prediction approaches using basic machine learning algorithms like Decision tree, Naive bayes and k-Nearest neighbors algorithm. We name our approaches

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

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. We selected the following topological attributes: Number of common neighbors (CN), Jaccard coefficient (JC), Preferential attachment (PA)[22], Adamic Adar coefficient (AA)[2], Resource allocation (RA)[45] and Shortest path length (SPL).

	Learning:1970-1973 Test:1972-1975	**Learning:1972-1975 Test:1974-1977**
Decision tree	0.0357	0.0168
Naive Bayes	0.1032	0.0070
Kemeny	0.2449	0.0860
Supervised Kemeny 1	0.4286	0.2581
Supervised Kemeny 2	0.4286	0.2258

TABLE 3. Link prediction results in terms of F1-measure, using different machine learning and supervised rank aggregation approaches

Table 3 summarizes the results obtained in terms of F1-measure. While K-nearest neighbors and our method based on Borda and supervised Borda failed to provide any substantial results (due to which we have not listed them in the table), our approximate Kemeny and supervised Kemeny based methods outperform the decision tree and naive bayes algorithms for both datasets. This shows the validity of our approach.

Although it is still early to say that rank aggregation based methods are better performing than the other approaches of link prediction, the preliminary results do show that rank aggregation especially with Kemeny method indeed adds some new information which may enhance the result of prediction task. This is quite encouraging for us to continue this work further. Still fact remains that rank aggregation methods especially Kemeny method has a high computational complexity which questions its applicability for link prediction in large scale networks. To cope with this we will be working on application of top-k rank aggregation. Much work needs to be done in this regards.

5. **Link prediction using multiplex links.** All these work that we saw till now, address the link prediction in only simple networks having homogeneous links. In this section we explain how prediction of links can be done in a multiplex scenario and how prediction performances can be enhanced using multiplex information.

To our knowledge, not much have been explored in this aspect. Some recent has tackled the problem of link prediction in heterogeneous networks [42]. There have also been few work on extending simple structural features like degree, path, to the context of multiplex networks [6, 8] but none have attempted to use them for link prediction.

We propose a new approach for exploring the multiplex relations to predict future collaboration (co-authorship links) among authors. The applied approach is supervised machine learning based, where we attempt to learn a model for link formation based on a set of topological attributes describing both positive and negative examples. While such an approach has been successfully applied in the context on simple

networks, different options can be used to extend it to the multiplex network context. One option is to compute topological attributes in each layer of the multiplex. Another one is to compute directly new multiplex-based attributes quantifying the multiplex nature of dyads (potential links). Both approaches will be discussed in the next section.

5.1. **Our approach.** Our approach includes computing simple topological scores for unconnected node pairs in a graph. We extend these attributes to include information from other layers of the network. This can be done in three ways:

- Compute simple topological measures in all layers.
- Compute simple aggregation values of these scores across all layers
- Compute an entropy-aggregation of values across all layers. This gives importance to the presence of a non-zero score of the node pair in each layer.

All above mentioned attributes can be combined in various ways to form different sets of vectors of attribute values characterizing each example or unconnected node pair. Formally, if we have a multiplex network $G =< V, E_1, \ldots, E_m >$ which in fact is a set of graphs $< G_1, G_2, \ldots, G_m >$ and a topological attribute X. For any two unconnected nodes u and v in graph G_i (where we want to make a prediction), $X(u, v)$ computed on G_i will be *direct* attribute and the same computed on all other dimension graphs will be *indirect* attributes. The second category computes an average of the attribute over all the dimension i.e. $X_{average} = \frac{\sum_{\alpha=1}^{m} X(u,v)^{[\alpha]}}{m}$ for $u, v \in V$ and $(u, v) \notin E_i$. where m is the number of types of relations in the graph (dimension or layer). In the third category we propose a new attribute called *product of node degree entropy (PNE)* which is based on *degree entropy*, a multiplex property proposed by F. Battistion et al. [6]. If degree of node u is $k(u)$, the degree entropy is given by: $E(u) = -\sum_{\alpha=1}^{m} \frac{k(u)^{[\alpha]}}{k_{total}} \log(\frac{k(u)^{[\alpha]}}{k_{total}})$ where $k_{total} = \sum_{\alpha=1}^{m} k(u)^{[\alpha]}$ and we define *product of node degree entropy* as

$$PNE(u, v) = E(u) * E(v) \tag{24}$$

We also extend the same concept to define entropy of a simple topological attribute, say X_{ent}

$$X_{ent}(u, v) = -\sum_{\alpha=1}^{m} \frac{X(u, v)^{[\alpha]}}{X_{total}} \log(\frac{X(u, v)^{[\alpha]}}{X_{total}}) \tag{25}$$

where $X_{total} = \sum_{\alpha=1}^{m} X(u, v)^{[\alpha]}$. The entropy based attributes are more suitable to capture the distribution of the attribute value over all dimensions. A higher value indicates uniform distribution attribute value across the multiplex layers. We address average and entropy based attributes as *multiplex attributes*.

5.2. **Experiments.** We evaluated our approach using data obtained from DBLP databases of which we created three datasets, each corresponding to a different period of time. Table.4 summarizes the information about the graphs of each dataset. Each graph has four years for learning or training and next two years are used to label the examples generated from the learning graphs. Examples are unconnected node pairs and they are labeled as *positive* or *negative* based on whether they are connected during the labeling period or not. Table.5 shows the number of examples obtained for each dataset.

We use the same attributes that were used in the previous section for supervised rank aggregation based approach i.e. Number of common neighbors (CN), Jaccard

Years	Properties	Co-Author	Co-Venue	Co-Citation
1970-1973	*Nodes*	91	91	91
	Edges	116	1256	171
1972-1975	*Nodes*	221	221	221
	Edges	319	5098	706
1974-1977	*Nodes*	323	323	323
	Edges	451	9831	993

TABLE 4. Basic statistics about the 3-layer multiplex networks extracted from te DBLP database

Years		# Positive	# Negatives
Train/Test	Labeling		
1970-1973	1974-1975	16	1810
1972-1975	1976-1977	49	12141
1974-1977	1978-1979	93	26223

TABLE 5. Number of examples extracted from co-authorship layer

coefficient (JC), Preferential attachment (PA)[22], Adamic Adar coefficient (AA)[2], Resource allocation (RA)[45] and Shortest path length (SPL). For any attribute XX

- XX_{aut} : Value of attribute was computed on co-authorship graph during learning period
- XX_{ven} : Value of attribute was computed on co-venue graph
- XX_{cit} : Value of attribute was computed on co-citation graph
- $AvgXX$: Average of the attribute value over the different relation graphs n our case $m = 3$ as we are using co-authorship, co-venue and co-citation graphs.
- PNE : Product of node degree entropy. If degree of node i is $k(i)$, the entropy for node i is calculated as

$$E_i = -\sum_{\alpha=1}^{m} \frac{k(i)^{[\alpha]}}{k_{total}} \log(\frac{k(i)^{[\alpha]}}{k_{total}}) \tag{26}$$

  where $k_{total} = \sum_{\alpha=1}^{m} k(i)^{[\alpha]}$ and

$$PNE(i,j) = E_i * E_j \tag{27}$$

- $XXent$: Entropy value of the corresponding attribute (based on the entropy equation proposed for node degree in the work of F. Battiston and al.)

$$XXent(i,j) = -\sum_{\alpha=1}^{m} \frac{XX(i,j)^{[\alpha]}}{XX_{total}} \log(\frac{XX(i,j)^{[\alpha]}}{XX_{total}}) \tag{28}$$

We apply decision tree algorithm on one dataset to generate a model and then tested it on another dataset. We are using data mining tool Orange[2] for that. We use four types of combinations of the attributes creating five different sets namely: Set_{direct}(attributes computed only in the co-authorship graph); $Set_{direct+indirect}$ (attributes computed in co-authorship, co-venue and co-citation graphs);

[2] http://orange.biolab.si

$Set_{direct+multiplex}$(attributes computed from co-authorship graph with average attributes obtained from three dimension graphs, and also entropy based attributes); Set_{all}(attributes computed in co-authorship, co-venue and co-citation graphs, with average of the attributes, and also entropy based attributes) and $Set_{multiplex}$(average attributes and entropy based attributes). Table.6 shows the result obtained in terms of F1-measure and area under the ROC curve (AUC). We can see that there is improvement in the F1-measure when we use multiplex attributes. AUC is better for all the sets that include multiplex and indirect attributes for both datasets.

Attributes	**Learning:1970-1973 Test:1972-1975**		**Learning:1972-1975 Test:1974-1977**	
	F-measure	**AUC**	**F-measure**	**AUC**
Set_{direct}	0.0357	0.5263	0.0168	0.4955
$Set_{direct+indirect}$	0.0256	0.5372	0.0150	0.5132
$Set_{direct+multiplex}$	0.0592	0.5374	0.0122	0.5108
Set_{all}	0.0153	0.5361	0.0171	0.5555
$Set_{multiplex}$	0.0374	0.5181	0.0185	0.5485

TABLE 6. Comparative link prediction results applying decision tree algorithm using different types of attributs

6. **Conclusion.** In this paper we present the problem of link prediction in complex networks and multiplex networks. We present here a brief state of art of various link prediction approaches focusing mainly on dyadic topological approaches. The unsupervised methods involve computation of scores for unlinked pairs of nodes. While neighborhood based scores are easy to compute, some path based measures like Katz, commute time, rooted pagerank can be really be time consuming. Same is the case of other matrix based approaches which have issues of computation time and memory when applied on real large scale networks. This makes them difficult to be employed for evolving real networks. So some approximate solutions for these measures such as truncated Katz and more can be a good choice.

In supervised approaches, especially machine learning based methods attempt is made to combine the effect of various topological attributes to generate a model which is then used to predict links on a test graph. The same is done in our proposed approach based on supervised rank aggregation. While machine learning methods have been in use since a long time and have given reliable performances in various contexts, supervised rank aggregation method is quite new and requires much work to establish its applicability in real applications. Also the fact remains that as they involve use of aggregation methods like approximate Kemeny aggregation, they have a computational complexity of $O(rn\log(n))$ where r is the number of attributes used and n is the number of examples in each input ranked list provided. But the preliminary results we get on the DBLP datasets validate the approach and encourage us to explore the method further.

A major challenge faced while using these types of supervised approaches, is the well known *extreme class skewness or class imbalance problem.* The number of actual new links is very small as compared to the number of possible links. As we can see, in the DBLP datasets we have used, the ratios of positive vs negatives links are 1:113 ,1:248 and 1:282 in the three datasets respectively. Also note that this imbalance increases with the size of graphs used for experimentation. This

makes it more difficult for an algorithm to generate a good model and give a good inference on the test data. Although very few of the negative examples have actual predictor value as positive examples, the model ends up giving a large number of raw false positives. Also in presence of large class skew, the information carried by the positive examples gets diluted in the vast negative class. Moreover unlike classical machine learning context, in link prediction, correct classification of positive examples are more important. Most common solution to this problem, as suggested by the existing research, is sampling of negative examples. This can be done by random methods or by using some filters by distance, node degree etc. Another way on which we are working is to use a filter based on community detection algorithm. The assumption here is that, two nodes that do not belong to the same community, tend to remain unlinked for a longer period than compared to those belonging to the same community. Thus they can have more meaning as negative examples during the learning of model. Each method has its own advantages and disadvantages, but some some can be fairer than others. The sampling of data is mostly done on the learning data. Sometimes it is required to sample test data like the case when extremely large number of test examples causes unreasonable demands on processing resources and storage. If for any reason this has to be done, proper care should be taken based on the context where link prediction is to be done. More details about class imbalance problem can be found in [36, 31]. In [31], there is a detailed description about how the predictor performance changes with sampling of test data. They also provide valuable information about which performance measure is to be used for evaluating different link prediction techniques.

Last but not the least, we have presented in the end of this paper, how to extend the traditional supervised machine learning based link prediction approach to predict links in multiplex networks. We propose new attributes that capture multiplex information. By applying them for the prediction of co-authorship links, we show that the use of multiplex attributes improves the prediction result. The same method can be used to predict links in any of the multiplex layers. With the preliminary results, we are really excited and hopeful that the multiplex information can prove to be very useful for different tasks in the analysis of the network.

REFERENCES

[1] L. Adamic and E. Adar, Friends and neighbors on the Web, *Social Networks*, **25** (2003), 211–230.

[2] L. A. Adamic, O. Buyukkokten and E. Adar, A social network caught in the Web, *First Monday*, **8** (2003), 1995–2015.

[3] C. C. Aggarwal, Y. Xie and P. S. Yu, A framework for dynamic link prediction in heterogeneous networks, *Statistical Analysis and Data Mining*, **7** (2014), 14–33.

[4] 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*, SIGIR '01, ACM, New York, NY, USA, 2001, 276–284.

[5] A.-L. Barabási and R. Albert, Emergence of scaling in random networks, *Science*, **286** (1999), 509–512.

[6] F. Battiston, V. Nicosia and V. Latora, Metrics for the analysis of multiplex networks, arXiv:1308.3182, (2013).

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

[8] M. Berlingerio, M. Coscia, F. Giannotti, A. Monreale and D. Pedreschi, Foundations of Multidimensional Network Analysis, in *Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on*, IEEE, 2011, 485–489.

[9] M. Berlingerio, M. Coscia, F. Giannotti, A. Monreale and D. Pedreschi, Multidimensional networks: Foundations of structural analysis, *World Wide Web*, **16** (2013), 567–593.
[10] 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 edition, Kluwer Academic Publishing, 1998.
[11] P. Brodka and P. Kazienko, *Encyclopedia of Social Network Analysis and Mining*, chapter Multi-Layered Social Networks, Springer, 2014.
[12] P. Chebotarev and E. Shamis, The matrix-Forest theorem and measuring relations in small social groups, *Automation and Remote Control*, **58** (1997), 1505–1514.
[13] Y. Chevaleyre, U. Endriss, J. Lang and N. Maudet, A short introduction to computational social choice, in *SOFSEM 2007: Theory and Practice of Computer Science*, Lecture Notes in Computer Science, 4362, Springer-Verlag, Berlin-Heidelberg, 2007, 51–69.
[14] D. A. Davis, R. Lichtenwalter and N. V. Chawla, Multi-relational link prediction in heterogeneous information networks, in *2011 International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, IEEE Computer Society, 2011, 281–288.
[15] J.-C. de Borda, *Memoire sur les Elections au Scrutin*, 1781.
[16] Y. Dong, J. Tang, S. Wu, J. Tian, N. V. Chawla, J. Rao and H. Cao, Link prediction and recommendation across heterogeneous social networks, in *2012 IEEE 12th International Conference on Data Mining (ICDM)* (eds. M. J. Zaki, A. Siebes, J. X. Yu, B. Goethals, G. I. Webb and X. Wu), IEEE Computer Society, 2012, 181–190.
[17] 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*, WWW '01, ACM, Hong Kong, 2001, 613–622.
[18] 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, 613–622.
[19] F. Fouss, L. Yen, A. Pirotte and M. Saerens, An experimental investigation of graph kernels on a collaborative recommendation task, in *Sixth International Conference on Data Mining (ICDM'06)*, IEEE, 2006, 863–868.
[20] S. Gao, L. Denoyer and P. Gallinari, Temporal link prediction by integrating content and structure information, in *Proceedings of the 20th ACM International Conference on Information and Knowledge Management - CIKM '11*, ACM Press, New York, New York, USA, 2011, 1169–1174. Available from: http://dblp.uni-trier.de/db/conf/cikm/cikm2011.html#GaoDG11.
[21] M. A. Hasan, V. Chaoji, S. Salem and M. Zaki, Link prediction using supervised learning, in *Proc. of SDM 06 workshop on Link Analysis, Counterterrorism and Security*, 2006.
[22] Z. Huang, X. Li and H. Chen, Link prediction approach to collaborative filtering, in *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries* (eds. M. Marlino, T. Sumner and F. M. S. III), ACM, 2005, 141–142.
[23] P. Jaccard, Étude comparative de la distribution florale dans une portion des alpes et des jura, *Bulletin de la Société Vaudoise des Sciences Naturelles*, **37** (1901), 547–579.
[24] L. Katz, A new status index derived from socimetric analysis, *Psychmetrika*, **18** (1953), 39–43.
[25] G. Kossinets, Effects of missing data in social networks, *Social Networks*, **28** (2006), 247–268.
[26] T.-T. Kuo, R. Yan, Y.-Y. Huang, P.-H. Kung and S.-D. Lin, Unsupervised link prediction using aggregative statistics on heterogeneous social networks, in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2013, 775–783.
[27] J. B. Lee and H. Adorna, Link prediction in a modified heterogeneous bibliographic network, in *ASONAM*, IEEE Computer Society, 2012, 442–449.
[28] 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*, CIKM '03, ACM, New York, NY, USA, 2003, 556–559.
[29] D. Liben-Nowell and J. M. Kleinberg, The link-prediction problem for social networks, *JASIST*, **58** (2007), 1019–1031.
[30] R. N. Lichtenwalter, J. T. Lussier and N. V. Chawla, New perspectives and methods in link prediction, in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '10*, ACM Press, New York, New York, USA, 2010, 243–252. Available from: http://dblp.uni-trier.de/db/conf/kdd/kdd2010.html#LichtenwalterLC10.

[31] R. Lichtnwalter and N. Chawla, Link prediction: Fair and effective evaluation, in *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 2012, 376–383.
[32] N. Littlestone and M. K. Warmuth, Weighted majority algorithm, *Information and Computation*, **108** (1994), 212–261.
[33] 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*, WWW '07, ACM, New York, NY, USA, 2007, 481–490.
[34] L. Lü and T. Zhou, Link prediction in complex networks: A survey, *Physica A: Statistical Mechanics and its Applications*, **390** (2011), 1150–1170.
[35] A. K. Menon and C. Eklan, Link prediction via matrix factorization, in *Machine Learning and Knowledge Discovery in Databases* (eds. D. Gunopulos, T. Hofmann, D. Malerba and M. Vazirgiannis), Lecture Notes in Computer Science, 6912, Springer Berlin Heidelberg, 2011, 437–452.
[36] A. H. Mohammad and Z. M. J., A survey of link prediction in social networks, in *Social Network Data Analysis* (ed. C. C. Aggarwal), Chapter 9, Springer, 2010, 243–275.
[37] M. Montague and J. A. Aslam, Condorcet fusion for improved retrieval, in *Proceedings of the Eleventh International Conference on Information and Knowledge Management*, CIKM '02, ACM, New York, NY, USA, 2002, 538–548.
[38] M. E. J. Newman, Coauthorship networks and patterns of scientific collaboration, *Proceedings of the National Academy of Science of the United States (PNAS)*, **101** (2004), 5200–5205.
[39] Q. Ou, Y. D. Jin, T. Zhou, B. H. Wang and B. Q. Yin, Power-law strength-degree correlation from resource-allocation dynamics on weighted networks, *Phys. Rev. E*, **75** (2007), 021102.
[40] M. Pujari and R. Kanawati, Supervised rank aggregation approach for link prediction in complex networks, in *WWW (Companion Volume)* (eds. A. Mille, F. L. Gandon, J. Misselis, M. Rabinovich and S. Staab), ACM, 2012, 1189–1196.
[41] K. Subbian and P. Melville, Supervised rank aggregation for predicting influence in networks, in *Proceedings of the IEEE Conference on Social Computing (SocialCom-2011)*, Boston, 2011.
[42] Y. Sun, R. Barber, M. Gupta, C. C. Aggarwa and J. Han, Co-Author Relationship Prediction in Heterogeneous Bibliographic Networks, in *Advances on Social Network Analysis and Mining (ASONAM)*, Kaohsiung, Taiwan, 2011, 121–128.
[43] C. Wang, V. Satuluri and S. Parthasarathy, Local Probabilistic Models for Link Prediction, in *IEEE International Conference on Data Mining (ICDM)* (eds. Y. Shi and C. W. Clifton), IEEE, 2007, 322–331.
[44] H. Young and A. Levenglick, A consistent extension of condorcet's election principle, *SIAM Journal on Applied Mathematics*, **35** (1978), 285–300.
[45] T. Zhou, L. Lu and Y.-C. Zhang, Predicting missing links via local information, *The European Physical Journal B,* **71** (2009), 623–630.

Received July 2014; revised December 2014.

E-mail address: manisha.pujari@lipn.univ-paris13.fr
E-mail address: rushed.kanawati@lipn.univ-paris13.fr