

April 27, 2010

RT0903

Mathematics 21 pages

Research Report

Proximity in skewed bipartite graphs with unsupervised
auxiliary information

Rudy Raymond and Hisashi Kashima

IBM Research - Tokyo
IBM Japan, Ltd.
1623-14 Shimotsuruma, Yamato
Kanagawa 242-8502, Japan

Limited Distribution Notice

This report has been submitted for publication outside of IBM and will be probably copyrighted if accepted. It has been issued as a Research Report for early dissemination of its contents. In view of the expected transfer of copyright to an outside publisher, its distribution outside of IBM prior to publication should be limited to peer communications and specific requests. After outside publication, requests should be filled only by reprints or copies of the article legally obtained (for example, by payment of royalties).



Proximity in Bipartite Graphs with Unsupervised Auxiliary Information

Rudy Raymond¹ and Hisashi Kashima²

¹ IBM Research, Tokyo Research Laboratory
1623-14 Shimo-tsuruma, Yamato
Kanagawa 242-8502, Japan
raymond@jp.ibm.com

² Department of Mathematical Information
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
kashima@mist.i.u-tokyo.ac.jp

Abstract. Measuring proximity between nodes in graphs, particularly bipartite graphs, is an important topic in information retrieval and knowledge management because it is fundamental to many applications, such as centrality tracking and link prediction. One of the most successful methods for measuring proximity is called the *Random Walk with Restart* (RWR), which is very efficient if the underlying bipartite graphs are highly unbalanced. However, although there is a recent work that uses supervised side information to modify the graph’s link structure, there is no known efficient method to exploit unsupervised auxiliary information. In this paper, we propose a new approach for combining the proximity scores of RWR with unsupervised auxiliary information on the relationships between the nodes (such as the similarities between contents of Web spam hosts in the host-host bipartite graphs, or those of pairs of users in the track-user bipartite graphs). Our approach is based on a minimization problem that guides the RWR to assign similar scores to similar nodes (and diverging scores to dissimilar nodes), without explicitly changing the structure of the underlying graph. On the technical side, we show that the proximity scores of RWR with the auxiliary information can be obtained efficiently by modifying the scores of the original RWR. As an interesting and practical real-world application, we show how to apply such proximity scores for labeling Web spam hosts and for recommendations by predicting links in large and skewed bipartite graphs.

1 Introduction

Many interactions in the real world can be expressed as bipartite graphs whose nodes can be partitioned into two parts (called *left* and *right* nodes) such that all edges of the graph link nodes from different parts. There are many natural examples of such graphs. For example, an author-conference bipartite graph whose nodes represent scientists or conferences, and whose edges link nodes corresponding to scientists to those of conferences, thus representing the contributed-to

relationship. Moreover, a general graph can be turned into a bipartite one by copying all of its nodes into left and right nodes, and adding corresponding links between left and right nodes appropriately. For this reason, we can also consider a host-host bipartite graph whose nodes represent Web hosts, and whose edges represent hyperlinks of Web pages on the hosts. Proximity scores of nodes on such graphs have many important applications in recommendation, ranking, link prediction, etc.

In many such typical bipartite graphs although the total number of nodes is large (for example, the number of scientific paper authors registered in DBLP from 1990 to 2008 is more than 490,000), they are often *skewed*, so the number of nodes in one part is relatively small compared to the other (there are only about 4,000 conferences registered in DBLP for the same period). Skewed bipartite graphs have special properties that can be exploited for designing efficient algorithms to compute proximity scores from their topological structures. One of them is the so-called Random Walk with Restart (or, RWR for short)[16], that calculates a proximity score of node v to node u from the steady-state probability of reaching v from u by a random walk. The principle of RWR is similar to the well-known random-surfer model of PageRank [12] on general graphs, however, the scores of RWR on bipartite graphs are easier to compute, especially, when the number of left and right nodes is highly unbalanced. This has sparked widespread interest on measuring proximities with RWR, even for dynamic bipartite graphs [17]. However, only little is known about how to take into account information other than the link structure for proximity measurements.

Quite recently, a fast algorithm for proximities that incorporates supervised auxiliary information for general graphs was proposed in [18], where the supervised auxiliary information was regarded as binary information and used to refine the link structure of the underlying graph. This was done with techniques taking into account the user's favourable preference by adding new links between the corresponding user's node and its marked positive nodes, and the unfavourable preference, by adding new links between the marked negative nodes and their neighbors with a special node without outlinks (a sink node). The techniques require careful selection of parameters for the weight of links between positive and negative nodes as well as for the selection of neighboring nodes.

At the same time, one can also obtain other types of information that take continuous values representing the degree of similarities between nodes in the graph. For example, the similarities of Web hosts in the host-host bipartite graphs can be calculated from the inner product of their keyword features, or, similarities of users can be measured from the overlaps of their tracks, friends or social tags, and so on. Those scores of similarities are obviously not discrete, and thus, present us with a challenge as to how to incorporate them for better proximity scores.

In this paper, we propose a new approach for refining the proximity scores of RWR with such unsupervised auxiliary information. Our approach is based on the *graph label propagation* for deriving a minimization problem that guides the RWR to assign similar scores to similar nodes (and diverging scores to dissimilar

nodes), without explicitly changing the structure of the underlying graph. The auxiliary information only gives the (dis)similarity scores of partial nodes in the graph and does not give their preferred order explicitly, and hence the term unsupervised. We designed our approach so that it still retains the advantages of RWR on large and skewed bipartite graphs. Its computational complexity is at most the same as that of the original RWR. Therefore, we believe that our approach will be useful for enriching the applicability and the effectiveness of the RWR. For this reason, we also include some interesting experimental results on applying proximity scores, from both the RWR and our new approach, in labeling Web spam hosts and in link prediction for bipartite graphs using real-world Web-spam host graphs and social network datasets.

To summarize, our contributions in this paper are:

- We present a novel approach of using unsupervised auxiliary information to adjust the proximity scores of RWR on large and skewed bipartite graphs.
- We describe an efficient procedure to obtain the adjusted scores incorporating the auxiliary information from the original scores of RWR.
- We present experimental results using proximity scores of the RWR and the adjusted scores for labeling Web spam hosts and for predicting links in bipartite graphs. We confirmed that the auxiliary information is helpful in refining the effectiveness of proximity scores of RWR.

The rest of the paper is organized as follows: We give definitions of the problems and symbols used in this paper in Section 2. We then present the RWR and our proposed method of RWR with unsupervised auxiliary information in Section 3. Section 4 summarizes our preliminary experimental results. Finally, we provide a brief summary of related work in Section 5 and concluding remarks in Section 7. The derivation of the new scores incorporating the auxiliary information is summarized in Appendix.

2 Definitions

We first explain notation, and then give the definitions of the problems we consider in this paper.

A bipartite graph $G(V, E)$ consists of a set of nodes V and a set of edges E . Its nodes can be partitioned into two disjoint sets: the left-node set L and the right-node set R such that $V = L \cup R$, and any edge $e \in E$ links a node in L with a node in R . Without loss of generality, we assume that the number of nodes in R , denoted as r , is at most that of nodes in L , denoted as l .

Bipartite graphs are represented by their adjacency matrices, and we use \mathbf{G} for denoting the adjacency matrix of G . A non-negative element of \mathbf{G} at row i and column j is denoted by G_{ij} and represents the link weight between nodes $i \in L$ and $j \in R$, and therefore \mathbf{G} is a $l \times r$ matrix. In this paper, matrices are always denoted by bold capital letters, such as, \mathbf{G} , and \mathbf{G}^T as its transpose, where its i -th row is denoted by $\mathbf{G}(i, \cdot)$, and its j -th column by $\mathbf{G}(\cdot, j)$. Following [17], we list all math symbols related to the original RWR in Table 1.

Table 1. Symbols in the original RWR

Symbol	Description
\mathbf{G}	the $l \times r$ adjacency matrix of bipartite graph G
L	the set of left nodes of G (of size l)
R	the set of right nodes of G (of size $r \ll l$)
\mathbf{D}_L	the $l \times l$ diagonal matrix whose element at row i and column i is $\sum_j G_{ij}$
\mathbf{D}_R	the $r \times r$ diagonal matrix whose element at row j and column j is $\sum_i G_{ij}$
\mathbf{I}	an identity matrix
$\mathbf{0}$	a zero matrix
c	fly-away probability (fap) of RWR
\mathbf{Q}	the proximity score matrix of dimension $(l+r) \times (l+r)$ from the original RWR, which is partitioned into 4 parts: $\mathbf{Q}_1, \mathbf{Q}_2, \mathbf{Q}_3$ and \mathbf{Q}_4 , each of dimension, $l \times l, l \times r, r \times l$, and $r \times r$, respectively, such that $\mathbf{Q} = \begin{pmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \\ \mathbf{Q}_3 & \mathbf{Q}_4 \end{pmatrix}$.

Unique to our approach is the unsupervised auxiliary information, which is given as a matrix \mathbf{A} whose real-valued element A_{ij} denotes the (dis)similarity scores between nodes i and j . Our approach outputs new proximity scores based on those of RWR on G and \mathbf{A} . In this paper, we assume that \mathbf{A} is symmetric, and the auxiliary information is available for nodes in R (i.e., the right nodes) whose size is much smaller and therefore easier to compute. Notice that the values of A_{ij} can be negative for denoting dissimilarity scores.

The assumption about this type of unsupervised auxiliary information is not unnatural in the following sense: We might have proximity scores between nodes in the graph from information other than the link structure. For example, in a host-host bipartite graph, the proximity scores due to the similarity of the contents of pages are likely to differ from those due to the link structure, and combining them to obtain better proximity scores is not trivial. Moreover, in an author-conference bipartite graph, it is relatively easier to infer similarities between conferences (e.g., from the research tracks) than those between authors.

We list additional symbols for matrices that appear in our new formulation of adjusting the proximity scores with the auxiliary information in Table 2. Notice that all matrices in the table are computable from \mathbf{G} and \mathbf{A} .

Given the above notation, the input and output to compute proximity scores in bipartite graphs with unsupervised auxiliary information in this paper is formalized as follows.

[Proximity Scores with Auxiliary Information]

Input: A bipartite graph $G(L \cup R, E)$ and an auxiliary information matrix \mathbf{A} , and a list of pairs of nodes $(u_1, r_1), (u_2, r_2), \dots$ such that $u_i \in L \cup R$ and $r_i \in R$ as queries.

Table 2. Symbols in RWR w/ Auxiliary Information

Symbol	Description
\mathbf{A}	the $r \times r$ auxiliary information matrix,
\mathbf{L}	the Laplacian matrix of \mathbf{A} , i.e., $= \mathbf{D}(\mathbf{A}) - \mathbf{A}$, where $\mathbf{D}(\mathbf{A})$ is the diagonal matrix whose element D_{ii} is the sum of row i of \mathbf{A} . $\mathbf{L} = \mathbf{L}_1 \mathbf{L}_2$ holds for some $l \times d$ and $d \times l$ matrices \mathbf{L}_1 , and \mathbf{L}_2
\mathbf{L}'	the $(l+r) \times (l+r)$ matrix $\begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{L} \end{pmatrix}$
\mathbf{Q}^*	the $(l+r) \times (l+r)$ new proximity score matrix incorporating \mathbf{L}
\mathbf{Q}_2^*	the new $l \times r$ proximity score matrix, corresponding to \mathbf{Q}_2
\mathbf{Q}_4^*	the new $r \times r$ proximity score matrix, corresponding to \mathbf{Q}_4
λ	a regularization parameter
$\mathbf{\Gamma}$	the $r \times r$ matrix defined as $\mathbf{I} + c^2 \mathbf{G}^T \mathbf{D}_L^{-1} \mathbf{D}_L^{-1} \mathbf{G}$
$\mathbf{\Lambda}$	the $r \times r$ matrix defined as $(\mathbf{I} + \lambda \mathbf{L}_2 \mathbf{Q}_4^T \mathbf{\Gamma} \mathbf{Q}_4 \mathbf{L}_1)^{-1}$
$\mathbf{\Delta}$	the $r \times r$ matrix defined as $\mathbf{Q}_4 \mathbf{L}_1 \mathbf{\Lambda} \mathbf{L}_2 \mathbf{Q}_4^T$

Output: The proximity scores of node r_i from node u_i for each query.

In above, we restrict the query to contain a node from the right set, with $r_i \in R$ as the second element of the query because we will mainly use the proximity scores for adjusting the proximity scores between the right nodes, and use the scores for predicting links between left and right nodes, as discussed below. However, as in the original RWR, proximity queries with regards to left nodes can be considered and the adjusted scores can be computed by our methods. We omit the details due to the space restriction.

We shall evaluate the effectiveness of proximity scores in handling two different tasks. The first task is the labelling of Web spam hosts in the host-host bipartite graph, whose data is available from the Web Spam Challenge Archive. This task is a preliminary step required for generating labeled training data for many state-of-the-art machine learning algorithms such as those described in [1]. As mentioned in [3], the task of labelling Web spam hosts is time consuming (10 hours on average to classify 200 hosts), and therefore a quick and reliable method to list all spam hosts is important. Good proximity scores can be used for this task by: (1) listing all hosts close to a known set of spam hosts, (2) labelling each of the host in the lists and adding newly found spam hosts to the known set, and repeat until sufficient number of spam hosts are found.

The second task is link prediction by applying the proximity scores in a similar setting to the ones discussed in [10, 11, 22]. For this purpose, for each $l \in L$ we randomly removed 20% or 50% links $(l, r) \in L \times R$ in a bipartite graph G to obtain graph G' , and perform RWR with auxiliary information to obtain the proximity scores between any two nodes in G' . For each node $l \in L$, we then computed the ranked list of nodes in R in the decreasing order of their proximity

scores to l , evaluated the precisions and recalls of the list against the removed links and calculated their averages.

The baseline method in this paper is the plain RWR which was showed to be superior than a collaborative filtering method [10], and have many novel applications [15, 17, 18].

3 Proximity Scores with RWR and Unsupervised Auxiliary Information

In this section, we first introduce the original RWR [6, 13, 15, 16] and then present our new approach.

3.1 The RWR on Bipartite Graphs

For any graph (not limited to a bipartite one), whose adjacency matrix is \mathbf{M} , the proximity score of node i to node j by the RWR is defined as the steady-state probability of being in node j when performing RWR started at node i [16]. The value of the steady-state probability of RWR from node i to node j can be computed recursively from the equation

$$Q_{ij} = c \sum_k Q_{ik} \frac{M_{kj}}{\sum_l M_{kl}} + (1 - c)\delta_{ij},$$

which means that the probability of reaching node j is the sum of probabilities of reaching nodes connected to j multiplied by the probabilities of moving from those nodes to node j . We can rewrite this equation to obtain the linear matrix equality

$$\mathbf{Q} = c\mathbf{Q}\mathbf{M} + (1 - c)\mathbf{I},$$

where \mathbf{M} is now the row-normalized adjacency matrix. This implies that the proximity scores can be obtained by solving the equation $\mathbf{Q} = (\mathbf{I} - c\mathbf{M})^{-1}$ (omitting the $1 - c$ factor since we are only interested in the rankings of the scores). However, there exists a more efficient method to compute scores for a bipartite graph G , because the sparse row-normalized adjacency matrix \mathbf{M} is

$$\mathbf{M} = \begin{pmatrix} \mathbf{0} & \mathbf{D}_L^{-1}\mathbf{G} \\ \mathbf{D}_R^{-1}\mathbf{G}^T & \mathbf{0} \end{pmatrix}.$$

From this, we can derive that the proximity scores of RWR on bipartite graphs are linear equations of scores of the right nodes. That is,

$$\begin{pmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \\ \mathbf{Q}_3 & \mathbf{Q}_4 \end{pmatrix} = (\mathbf{I} - c\mathbf{M})^{-1},$$

where $\mathbf{Q}_1, \mathbf{Q}_2$ and \mathbf{Q}_3 are linear in \mathbf{Q}_4 , as in the equations

$$\mathbf{Q}_1 = \mathbf{I} + c^2\mathbf{D}_L^{-1}\mathbf{G}\mathbf{Q}_4\mathbf{D}_R^{-1}\mathbf{G}^T, \quad (1)$$

$$\mathbf{Q}_2 = c\mathbf{D}_L^{-1}\mathbf{G}\mathbf{Q}_4, \quad (2)$$

$$\mathbf{Q}_3 = c\mathbf{Q}_4\mathbf{D}_R^{-1}\mathbf{G}^T. \quad (3)$$

The $r \times r$ matrix \mathbf{Q}_4 is obtained from the equation

$$\mathbf{Q}_4 = (I - c^2 \mathbf{D}_R^{-1} \mathbf{G}^T \mathbf{D}_L^{-1} \mathbf{G})^{-1}, \quad (4)$$

which is relatively easy to compute when r is small. The fact that all of the proximity scores of RWR on bipartite graphs are related by Eqs. (1)–(4) results in computational advantages. We can instead compute the inverse of the smaller $r \times r$ matrix in Eq. (4) to obtain the inverse of the larger $(l+r) \times (l+r)$ matrix $\mathbf{I} - c\mathbf{M}$.

As an example, let us consider the bipartite graph in Fig. 1. There are 9 left nodes (labeled with numbers from 1 to 9), 6 right nodes (labeled with letters from a to f), and 11 edges in the graph whose adjacency matrix is given in the same figure. For each node $u \in \{1, \dots, 9\}$ of the bipartite graph, we can compute the ordered proximity scores between u with $v \in \{a, \dots, f\}$ using Eqs. (2) and (4) with the fly-away probability (fap) $c = 0.3$, and obtain the ordered list of right-node neighbors for each node u as shown in the figure. For example, nodes that are close to node 1 are nodes a and d (in that order), while for each right node, there is only at most one right node close to it.

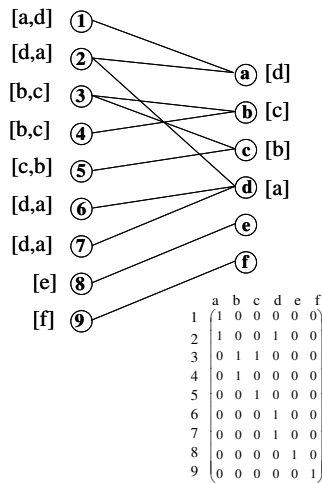


Fig. 1. An example of a bipartite graph and its adjacency matrix. A list of right nodes close to each of its nodes, as computed from the scores of the original RWR, is shown enclosed in parathenses next to the corresponding node.

3.2 The New RWR with Unsupervised Auxiliary Information

We present the procedure to compute the new RWR scores incorporating auxiliary information while describing the principle behind their derivation. It is easy

to see that the RWR scores are also the solution of the minimization problem

$$\min_{\mathbf{Q}} \|\mathbf{Q} - (c\mathbf{Q}\mathbf{M} + (1-c)\mathbf{I})\|_F^2, \quad (5)$$

where $\|\cdot\|_F$ is the Frobenius matrix norm. Now given scores Q_{ij} , Q_{ik} and auxiliary information A_{jk} , we add $\beta_i = \sum_{j,k} (Q_{ij} - Q_{ik})^2 A_{jk}$ terms for all $i \in V$. The meaning of the minimization of these terms is that if node j is similar to (divergent from) node k , then the proximity score of node i to node j should be close to (far from) that of node i to node k . Combined with the restriction that the new scores should not be too far from those of RWR, i.e., Eq. (5), the new scores are generated by *propagating* proximity scores of the original RWR in accordance to the auxiliary information matrix \mathbf{A} . This is the principle of label propagation on graphs which has received a lot of attention in the link-mining community [5, 9, 20, 22].

This means, in addition to Eq. (5), we also want to minimize the sum of the scalar β_i .

$$\begin{aligned} \beta_i &= \sum_{j,k} (Q_{ij} - Q_{ik})^2 A_{jk}, \\ &= 2 \sum_j Q_{ij}^2 \sum_k A_{jk} - 2 \sum_{j,k} Q_{ij} Q_{ik} A_{jk}, \\ &= 2 (\mathbf{Q}(i, \cdot) \mathbf{D}(\mathbf{A}) \mathbf{Q}(i, \cdot)^T - \mathbf{Q}(i, \cdot) \mathbf{A} \mathbf{Q}(i, \cdot)^T), \end{aligned} \quad (6)$$

where $\mathbf{D}(\mathbf{A})$ is the diagonal matrix whose element D_{ii} is defined as the sum of the row i of \mathbf{A} . Summing Eq. (6) for all i , we have $\sum_i \beta_i = \text{Tr}(\mathbf{Q}\mathbf{L}\mathbf{Q}^T)$, where $\mathbf{L} = \mathbf{D}(\mathbf{A}) - \mathbf{A}$ is the Laplacian matrix of the auxiliary information. In this paper we do not require \mathbf{A} to be a non-negative matrix, and therefore \mathbf{L} is not the strict graph Laplacian. Thus, our new objective function for obtaining proximity scores with unsupervised auxiliary information, is

$$\min_{\mathbf{Q}} \|\mathbf{Q} - (c\mathbf{Q}\mathbf{M} + (1-c)\mathbf{I})\|_F^2 + \lambda \text{Tr}(\mathbf{Q}\mathbf{L}'\mathbf{Q}^T), \quad (7)$$

where λ is a real positive regularization parameter.

The solution \mathbf{Q}^* minimizing this new objective function if exists can be obtained by differentiating it, which results in (see Appendix for matrix derivatives used)

$$\mathbf{Q} \left(2(\mathbf{I} - c\mathbf{M})(\mathbf{I} - c\mathbf{M})^T + 2\lambda\mathbf{L}' \right) - 2(1-c)(\mathbf{I} - c\mathbf{M})^T. \quad (8)$$

Setting the above equation to 0, we obtain the new proximity score matrix \mathbf{Q}^* from the linear equation (again, omitting $(1-c)$ for brevity)

$$\mathbf{Q}^* = (\mathbf{I} - c\mathbf{M})^T \left((\mathbf{I} - c\mathbf{M})(\mathbf{I} - c\mathbf{M})^T + \lambda\mathbf{L}' \right)^{-1}. \quad (9)$$

After some calculations involving the use of the Sherman-Morrison-Woodbury Lemma [14] to obtain the inverse of the sum of matrices on the right-hand side

Algorithm 1 Proximity scores by RWR on bipartite graphs and auxiliary information. Input: $(\mathbf{G}, \mathbf{A}, c, \lambda)$.

- 1: Compute RWR scores \mathbf{Q}_2 and \mathbf{Q}_4 as in Eqs. (2) and (4).
 - 2: Compute matrices $\mathbf{\Delta}$ and $\mathbf{\Gamma}$ as in Table 2.
 - 3: Compute the adjusted scores \mathbf{Q}_4^* and \mathbf{Q}_2^* as in Eqs. (10) and (11).
-

of the Eq. (9), we can write \mathbf{Q}_2^* and \mathbf{Q}_4^* , the proximity scores we need for the output, in terms of the original RWR score matrix \mathbf{Q}_4 ,

$$\mathbf{Q}_4^* = (\mathbf{I} - \lambda \mathbf{\Delta} \mathbf{\Gamma}) \mathbf{Q}_4 \quad (10)$$

$$\mathbf{Q}_2^* = c \mathbf{D}_L^{-1} \mathbf{G} \mathbf{Q}_4^*, \quad (11)$$

where the $r \times r$ matrices $\mathbf{\Gamma}$ and $\mathbf{\Delta}$ are defined as in Table 2. It can be seen that when $\lambda = 0$ or $\mathbf{L}' = \mathbf{0}$, $\mathbf{Q}_4^* = \mathbf{Q}_4$ and $\mathbf{Q}_2^* = \mathbf{Q}_2$, as expected. The details of derivation of \mathbf{Q}_2^* and \mathbf{Q}_4^* are listed in Appendix.

From the above equations, assuming that we already have computed the original RWR and obtained \mathbf{Q}_4 , computing \mathbf{Q}_2^* and \mathbf{Q}_4^* can be done within $O(r^3 + mr)$ computation time, where m is the number of nonzero elements of \mathbf{G} . This is mainly due to the matrix inversion and multiplication whose computational complexities are asymptotically the same with that of the original RWR. The procedure to compute the adjusted RWR scores using auxiliary information is summarized in Algorithm 1

To see how the proximity scores with auxiliary information are different from those of the original RWR, let us again consider the example in Fig. 1 with auxiliary information given indicating that node a is similar to nodes c and d , and node e to node f with $\lambda = 1$. We can see from Fig. 2 that in the proximity scores with auxiliary information there are more right nodes close to each of the nodes in the graph. For example, in addition to node d (the original neighbor), nodes c, b are now close to node a because of the auxiliary info A_{ac} and the path between node b and node c in the graph. Notice that the inclusion of new right-node neighbors obeys both the conditions implied by the auxiliary information and those by the connectivity of the graph.

4 Experiments

We first explain the datasets used for comparing the proximity scores of the original RWR with those of RWR with auxiliary information, and present the experimental results.

4.1 Datasets and Setups

[Web spam host datasets] This dataset is available from the Web Spam Challenge archive³. It was created from the network of 9,072 Web hosts (called

³ See <http://webspam.lip6.fr/wiki/pmwiki.php?n=Main.PhaseIITrainingCorpora>

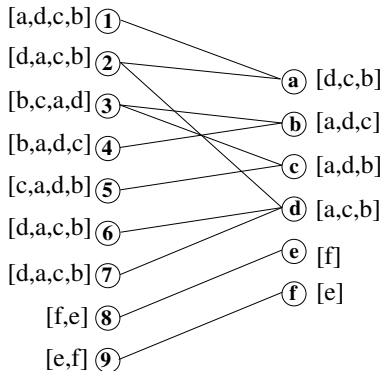


Fig. 2. An example of a bipartite graph and the neighbors of its nodes, as computed from the scores of the RWR with the auxiliary information matrix whose non-zero elements are $A_{ac} = A_{ad} = A_{ef} = 1$.

Table 3. Datasets used in the experiments.

Name	Size ($l \times r$)	Edges	Description
Host1K	1,000 \times 1,000	6,222	host-host.
Host9K	9,072 \times 9,072	514,700	host-host.
LastFM-TrU	30,520 \times 3,148	580,879	track-user.
AC1990-2008	494,752 \times 4,058	1,337,230	author-conf.

Host9K dataset) whose links correspond to the existences of hyperlinks between pages in the hosts. There are 1,728 hosts marked as “spam”. For each host, there is a sparse feature vector that represents a normalized tf-idf vector over the contents of its pages. We used the feature vectors for unsupervised auxiliary information. For each host h in the hostgraph, we computed the closest host h' (note that $h' \neq h$), based on the value of the cosine similarities of their feature vectors, and set $A_{hh'}$ equal to the value. All other A_{hj} values for $j \neq h'$ are set to zero. This resulted in an auxiliary matrix with a density of $\leq 3.0 \times 10^{-4}$. We also created a small subgraph from Host9K that consists of the first 1,000 nodes of Host9K (and contains 204 spam hosts). We call this dataset *Host1K* and used it to set parameters of RWR and our methods. We found that the cases with $\text{fap}=0.2$ and $\text{fap}=0.95$ have typical properties. Experimenting with λ equal to 0.5, 0.05, 0.005, 0.0005, we found that $\lambda = 0.005$ gave the best MAP result. We used this value for all experiments in this paper.

The effectiveness of labelling web spam hosts is computed in three measures against the set of Web spam hosts. The first one is the precision-recall of a list of hosts that are close to a known Web spam host (or, the precision-recall of a *blacklist*). A good labelling algorithm will produce a blacklist that contains larger number of spam hosts on the top of the list (or, high in precision and recall rate). The second one is the ratio of Web spam hosts contained in a list of

hosts that are close to a given non Web spam host (or, the false-positive rate of a *whitelist*). A good labelling algorithm will give a whitelist that contains smaller number of spam hosts on the top of the list (or, low false-positive rate). The third one is the *number of relevant* spam hosts that are retrievable from several blacklists of length k . The number is denoted by $nr(k)$ and is computed in these steps: For each Web spam host h , let us denote the set of all Web spam hosts contained in its top- k whitelist by $S(k, h)$ (h is not contained in $S(k, h)$). Then, $nr(k)$ is the size of the union of $S(k, h)$ of all h in the Web spam hosts. In this paper, all precision-recall curves are from the average of precisions and recalls at lists of length 1,3,5,10,30,50 and 100.

[LastFM datasets] This dataset is part of the last.fm social network described in [10]. LastFM-TrU is a bipartite graph whose right nodes represent users and left nodes represent tracks they have listened to. The weighted links between left and right nodes in those graphs represent listened-by relations with weight denoting the playcount number. There are two track-user bipartite graphs available: a small dataset which is used in the experiments in [10] and a big one (528,235 tracks \times 3,398 users) which is the superset of the small one. We constructed two bipartite graphs $B1$ and $B2$ from the small dataset by removing 50% and 20% links, respectively, for evaluation purposes as described in Section 2, and the auxiliary information from the big dataset. The auxiliary information was computed for each user from the top-10 closest users in terms of cosine similarities of their feature vectors whose elements represent the tracks he/she has listened to (excluding those removed for evaluations). This resulted in an auxiliary information matrix with a density of $\leq 6.0 \times 10^{-3}$. We used $fap=0.2$.

[DBLP datasets] The DBLP dataset is the author-conference bipartite graphs whose left nodes are authors, and right nodes are the conferences that took place between 1990 and 2008, and the link weights are the number of papers that the authors contributed to the conferences⁴. We constructed the author-conference bipartite graph (called, AC1990-2008) to compare our proximity scores with the *Proximity with Side Information* (ProSIN) method in [18], in particular, to show how to take into account negative auxiliary information on a familiar set of scientific conferences.

4.2 Results

Web spam datasets In this paper, the Web spam datasets are the only ones with ground truth obtained from extensive human judgement. We used these datasets to apply RWR and our methods to efficiently find as many spam hosts as possible. Given a (non) spam host, we can utilize the link structure of hostgraph to obtain a blacklist (whitelist) of hosts to be examined. Those lists can be created by ordering hosts from their proximity scores.

We experimented on RWR with fly-away probability (fap) varied between 0.2 to 0.95 on the Host1K dataset and found that RWR with $fap=0.95$ gives the best precision-recall curves, while that with $fap=0.2$ gives the lowest ones.

⁴ Available from <http://www.informatik.uni-trier.de/~ley/db>

However, as we shall see later in Table 4, the number of relevant spam hosts of RWR with $\text{fap}=0.95$ is small, suggesting that the lists are almost similar and not good for searching spam hosts. We presented experimental results with $\text{fap}=0.2$ and $\text{fap}=0.95$ since all other results lie between their curves.

Fig. 3 shows the precision-recall curves of blacklists on the Host1K dataset. It is shown that our proposed methods (**Aux** $\text{fap}=0.2$ and $\text{fap}=0.95$) are better than the baseline RWR since the curves of our methods dominate those of the baseline. The differences are significant at shorter prediction length (up to 50% precision increase compared to the baseline RWR for $\text{fap}=0.2$), which implies the higher probability of encountering spam hosts on top of the blacklists. They pass a sign test with $p < 0.0001$ except **Aux** $\text{fap}=0.95$ at the prediction of length 1, 5 and 10 with $p < 0.05$, and **Aux** $\text{fap}=0.2$ at the prediction of length 5 with $p < 0.005$. Moreover, Fig. 4 also shows that **Auxs** outperform the baseline RWR in the quality of their whitelists on the Host1K dataset (statistically significant with $p < 0.0001$ at the prediction of length ≤ 10). The RWRs on Host1K produce low-quality whitelists but **Auxs** can generate much better whitelists.

From Fig. 5, on the Host9K dataset we can also see that **Auxs** significantly outperform the baseline RWRs. The differences pass a sign test with $p < 0.0001$. The precisions at the prediction length 1 are increased up to 6 times by our methods. On this dataset, the RWR with $\text{fap}=0.2$ is better than that with $\text{fap}=0.95$, but it generates low-quality blacklists. **Auxs** can boost the quality of blacklists without sacrificing the quality of the whitelists much as seen from Fig. 6; the false-positive rate of **Aux** $\text{fap}=0.2$ is better at the prediction length of ≤ 5 (statistical significance at $p < 0.05$). The false positive rates of **Aux** $\text{fap}=0.95$ are worse than the corresponding RWR, but the absolute values are small. The figures also show that at short prediction length, **Auxs** can decrease the variations in the quality of blacklists and whitelists due to the variation of fap . These were achieved by adding very sparse auxiliary information matrices. Notice that the recalls of **Aux** and RWR with $\text{fap}=0.95$ in Fig. 5 were very low (less than 0.005) compared to those of **Aux** and RWR with $\text{fap} = 0.2$. We suspect that this is due to the locality of the link structure in the host graphs. In addition to links to normal hosts, spam hosts tend to have more direct links to other spam hosts and therefore the blacklist obtained by a random walk from a spam host with low fap contains more neighboring spam hosts within a few links from the origin of the walk. On the other hand, the blacklist obtained by a random walk from a spam host with high fap contains more normal hosts (that are linked by many other hosts) and hence the low recalls in Fig. 5.

Finally, Table 4 shows another aspect of effectiveness of the RWR with auxiliary information. We can see from the table that there are only a few spam hosts on the blacklists of the baseline RWRs. For example, $\text{nr}(1)$ for the RWR with $\text{fap} = 0.2$ and $\text{fap}=0.95$ on the Host1K dataset are, respectively, 13 and 6. In contrast, by our methods (**Auxs**) the corresponding numbers are 103 and 101, respectively. The same tendency can be observed on the Host9K dataset.

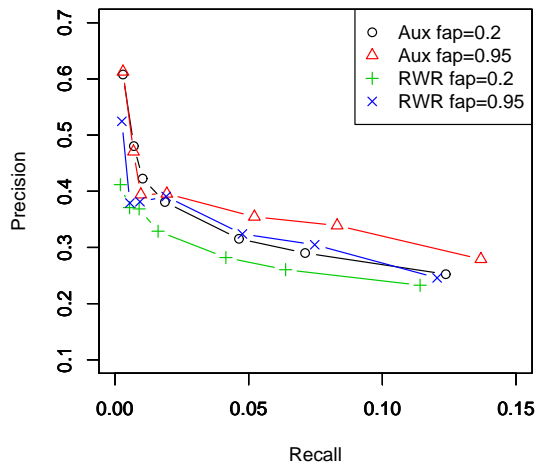


Fig. 3. Interpolated Precision-Recall curves of blacklists on the Host1K dataset

Table 4. Number of Relevant Spam Hosts Retrieved from the Web Spam Hosts

Dataset Method / nr(k)	k=1	k=3	k=5	k=10	k=30	k=50	k=100
Host1K RWR fap=0.2	13	28	41	66	112	126	159
Host1K Aux fap=0.2	103	166	188	199	204	204	204
Host1K RWR fap=0.95	6	10	14	27	55	71	100
Host1K Aux fap=0.95	101	149	172	182	200	202	202
Host9K RWR fap=0.2	204	414	530	741	1047	1223	1406
Host9K Aux fap=0.2	762	1118	1286	1463	1647	1693	1713
Host9K RWR fap=0.95	55	137	177	247	333	369	458
Host9K Aux fap=0.95	718	1017	1181	1360	1563	1606	1656

LastFM datasets In Fig. 7 it is shown that our methods also outperform the baseline RWR method on the LastFM datasets since the precision-recall curves of our methods dominate those of the baseline (In the figure, RWR B1 and B2 denote the results of the baseline RWR on the graphs B1 and B2, respectively, while **Aux** B1 and B2 denote results of our methods on the corresponding graph). The comparisons are statistically significant at $p < 0.0001$. The results indicated that parts of the big dataset that were not in the small dataset can be used to increase the effectiveness of link predictions. We also tried to use tag-user and user-user (friendship) networks as auxiliary information but could not obtain improvement in precisions, perhaps, because of the sparsity of the networks ([10] also reported the decrease in precisions, albeit an increase in recalls, due to the introduction of social networks).

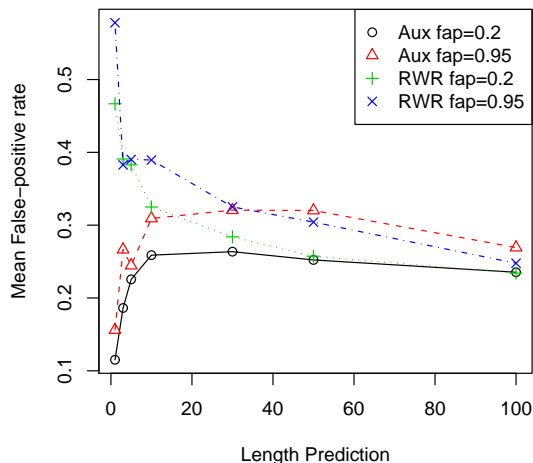


Fig. 4. Interpolated false-positive rate curves of whitelists on the Host1K dataset

DBLP datasets We show the comparison of our methods with the ProSIN of [18] using the same *neighborhood search* example. This means, we want to find a list of the conferences which is close to the KDD conference from the AC1990-2008 dataset. Table 5 shows the lists according to the proximity scores of the baseline RWR, and according to our methods with two types of auxiliary information; the first one, whose list (**Aux1**) is shown in the middle of the table, shows the result when the auxiliary information is $A_{\text{KDD},\text{SIGIR}} = 1$ and $A_{\text{KDD},\text{ICML}} = -1$ (or, close to SIGIR but far from ICML), while the second one, whose list (**Aux2**) is shown on the right, is when the auxiliary information is $A_{\text{KDD},\text{SIGIR}} = -1$ and $A_{\text{KDD},\text{ICML}} = 1$ (or, far from SIGIR but close to ICML). In ProSIN [18], the first auxiliary information corresponds to setting SIGIR and ICML as a *positive* and *negative* node, respectively, while the second, vice versa. We obtained similar results with those in ProSIN, that is, the results make sense, but unlike ProSIN, our method requires less parameters and is simpler. For example, in **Aux1**, Information Retrieval (IR) related conferences, such as CIKM and TREC in the top 5, and WWW, CLEF, JCDL in the top 20, become closer to KDD while major AI/statistics-related conferences, such as NIPS and ECML, disappear from the top 20. In contrast, in **Aux2** AI/statistics related conferences, such as NIPS and ECML, become closer to KDD while some IR-related conferences, such as, TREC, CIKM, WWW, and CLEF, are dropped from the top 20.

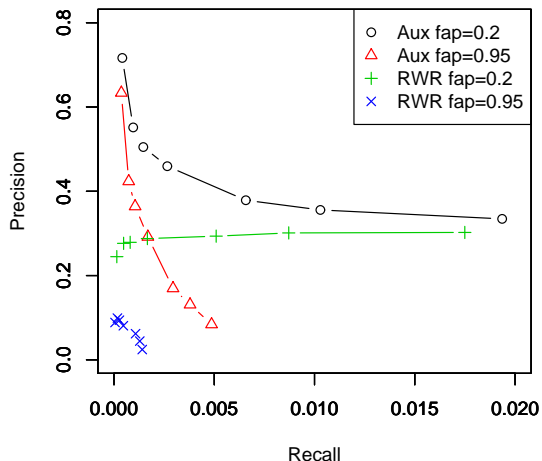


Fig. 5. Interpolated Precision-Recall curves of blacklists on the Host9K dataset

5 Related Work

The related work can be divided into two main types: proximity scores using RWR and link prediction. We give a brief review for each of them.

[Proximity Scores using RWR] RWR is quite prominent in the literature, such as [6, 13, 15, 16], which mostly dealt with static (unchanged) graphs. A clever observation on the possibility of exploiting the skewedness property of graphs appeared in [15]. This idea is further explored in [16], where the proximity scores of nodes of a general graph are approximated by the proximity scores of its corresponding k -partite graph, which can be efficiently computed by some parallel graph coloring algorithm techniques [8].

A fast and efficient method of updating proximity scores for RWR without recomputing from scratch was shown in [17]. The idea is based on an observation that the core matrix \mathbf{Q}_4 can be updated efficiently when only a small part of G changes. By the linearity relation of \mathbf{Q}_4 with \mathbf{Q}_4^* (the RWR scores with auxiliary information in this paper), similar techniques can be used to dynamically update \mathbf{Q}_4^* .

Our method was inspired by the recent work in [18], where the auxiliary information (called the side information) is of supervised type (like/dislike between any pair of nodes), and the RWR scores are computed by adjusting the underlying graph structure. In contrast, our methods only require auxiliary information among the right nodes in the bipartite graphs, and it works with unsupervised auxiliary information (the degrees of similarities in real numbers suffice) without modifying the underlying graph.

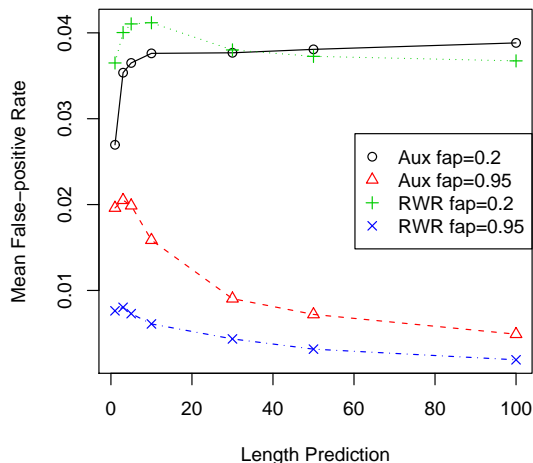


Fig. 6. Interpolated false-positive rate curves of whitelists on the Host9K dataset

[Link prediction] The problem of predicting the structure of networks is called the *link prediction problem*, which is one of the important tasks of link mining [4] in the data mining community. A typical link prediction problem is to predict the unknown parts of the structure of a network (or the future structure of the network) from the known parts of the network, which results in a completion problem of adjacency matrices.

Another type of link prediction is in using proximity scores as detailed in [11] with extensive comparison results using various scoring methods. RWR is closely related to the weighted Katz measure in [11] as can be seen from the Taylor expansion of $(\mathbf{I} - c\mathbf{M})^{-1}$. Interestingly, [11] reported that the Katz measure and its variants achieved the best performances for link predictors in their experiments.

Quite recently, [22] proposed a label-propagation-based method to compute proximity scores from multiple graphs. The idea can be regarded as adjusting proximity scores of a *symmetric* network of items in a low-dimensional semantic space by using several *bipartite* graphs (that contain the set of items) as auxiliary information. In contrast, our methods adjust proximity scores of RWR in a *bipartite* graph that contains the set of items by using a *symmetric* network of items as auxiliary information. In addition, because our methods are specially tuned for bipartite graphs, we can compute proximity scores of nodes on much larger graphs.

Link prediction methods can fall into two categories in accord with the information used for prediction: (i) topology-based methods (e.g. [7, 11, 21]) and

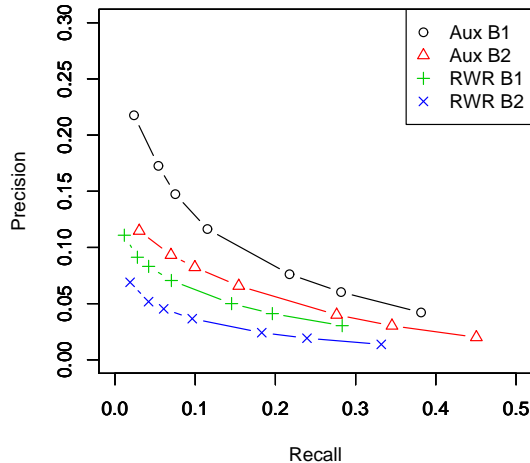


Fig. 7. Interpolated Precision-Recall curves on predicting track-user links on LastFM datasets when RWR with $\text{fap}=0.2$

(ii) feature-based methods (e.g. [2, 9, 19]). The former methods use only information in adjacency matrices, using similarities among nodes called link metrics computed from observed parts of the adjacency matrices. In contrast, the latter methods exploit node information such as feature vectors of nodes or similarity values among nodes. Our methods in this paper can be regarded as a proposal combining the two methods.

6 Discussion

We have showed how to efficiently incorporate auxiliary information to proximity scores of RWR in large and skewed bipartite graphs. We should also mention that the auxiliary information used in the experiment is very sparse (only the top-1 host for Web spam datasets and top-10 closest users for LastFM datasets) which also implies that we only need a small amount of node information to boost proximity scores from the link information. The advance use of proximity scores by our method and the comparison with other state-of-the-art methods using this sparse auxiliary information are left as the future work. Here we mention several limitations of our method and ideas on how to mitigate them.

First, to apply our method to a general graph not limited to a bipartite one, we can consider the corresponding bipartite graph as mentioned in Sec. 1 and compute the proximity scores from the corresponding bipartite graph. Notice that it is exactly what we have done with the Web spam host datasets whose resulting bipartite graphs are not skewed.

Table 5. The Top-20 Neighbors of KDD Conferences

By RWR	By Aux1 (close to sigir but far from icml)	By Aux2 (close to icml but far from sigir)
icdm, icml, icde sdm, nips, sigmod, vldb, cikm, pkdd, pakdd, aaai, www, sigir, ijcai, uai, ecml,aaai/iaai, ssdbm, ictai, icdm-workshops	sigir, cikm, icdm, trec,icde, sigmod, vldb, sdm, www, pakdd, pkdd, ecir, ssdbm, edbt, dasfaa, sac, clef, jcdl,pods, acm-multimedia	icml,nips, icdm, sdm, pkdd, ecml, aaai, pakdd, uai, sigmod, icde, ijcai, aaai/iaai, vldb, colt, ictai, ssdbm, ilp, icdm-workshops,cvpr

Second, when the size of the right node set R is big (say, in the order of millions), then our method cannot be applied directly. This is also the difficulty faced by the baseline RWR method. However, notice that in many cases, such as user-item and user-blog bipartite graphs, R corresponds to the set of recommended and high-value items or blogs and therefore a simple preprocessing can be used to filter items or blogs to be included in R . For example, in our experience working with a blog site provider that have millions of bloggers, within a given period only less than 1% of bloggers whose blogs were accessed by more than 80% of active users in the blogspace. This empirical observation is also consistent with the “1% rule” in Internet culture⁵.

7 Concluding Remarks

In this paper, we have presented a novel approach of using unsupervised auxiliary information to compute proximity scores between nodes in large and skewed bipartite graphs. We showed that the new proximity scores incorporating the auxiliary information can be computed efficiently, and are better than the proximity scores of RWR for applications such as labelling Web spam hosts and link predictions. This leverages the possibility of using RWR with auxiliary information for link prediction and recommender systems. We plan to use the proximity scores in link prediction and recommender systems and perform the comparison with the state-of-the-art methods. Another interesting future research is on how to integrate multiple sources of auxiliary information, like the setting in [22], while maintaining the computational efficiency.

References

1. J. Abernethy, O. Chapelle, and C. Castillo. Web spam identification through content and hyperlinks. In *AirWeb'08*. ACM, 2008.

⁵ See [http://en.wikipedia.org/wiki/1%25_rule_\(Internet_culture\)](http://en.wikipedia.org/wiki/1%25_rule_(Internet_culture))

2. A. Ben-Hur and W. S. Noble. Kernel methods for predicting protein-protein interactions. *Bioinformatics*, 21(Suppl. 1):i38–i46, 2005.
3. C. Castillo, D. Donato, L. Becchetti, P. Boldi, S. Leonardi, M. Santini, and S. Vigna. A reference collection for web spam. *SIGIR Forum*, 40(2):11–24, 2006.
4. L. Getoor and C. P. Diehl. Link mining: a survey. *SIGKDD Explor. Newsl.*, 7(2):3–12, December 2005.
5. R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *WWW '04*, pages 403–412, New York, NY, USA, 2004. ACM.
6. J. He, M. Li, H.-J. Zhang, H. Tong, and C. Zhang. Manifold-ranking based image retrieval. *ACM Multimedia*, 2004.
7. G. Jeh and J. Widom. SimRank: A measure of structural-context similarity. In *KDD'02*, pages 538–543. ACM, 2002.
8. G. Karypis and V. Kumar. Paralell multilevel k-way partitioning for irregular graphs. *SIAM Review*, 41(2):278–300, 1999.
9. H. Kashima, T. Kato, Y. Yamanishi, M. Sugiyama, and K. Tsuda. Link propagation: A fast semi-supervised learning algorithm for link prediction. In *SDM'09*, pages 1099–1110. SIAM, 2009.
10. I. Konstas, V. Stathopoulos, and J. M. Jose. On social networks and collaborative recommendation. In *SIGIR'09*. ACM, 2009.
11. D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *CIKM '03*, pages 556–559. ACM Press, 2003.
12. L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.
13. J.-Y. Pan, H.-J. Yang, C. Faloutsos, and P. Duygulu. Automatic multimedia cross-modal correlation discovery. In *KDD '04*, pages 653–658. ACM, 2004.
14. W. Piegorsch and G. E. Casella. Inverting a sum of matrices. *SIAM Review*, 32:470–, 1990.
15. J. Sun, H. Qu, D. Chakrabarti, , and C. Faloutsos. Neighborhood formation and anomaly detection in bipartite graphs. In *ICDM '05*, pages 418–425, 2005.
16. H. Tong, C. Faloutsos, and J.-Y. Pan. Random walk with restart: Fast solutions and applications. *Knowledge and Information Systems: An International Journal (KAIS)*, 2008.
17. H. Tong, S. Papadimitriou, P. S. Yu, and C. Faloutsos. Proximity tracking on time-evolving bipartite graphs. In *SDM '08*, pages 704–715. SIAM, 2008.
18. H. Tong, H. Qu, and H. Jamjoon. Measuring proximity on graphs with side information. In *ICDM'08*. IEEE, 2008.
19. J.-P. Vert and Y. Yamanishi. Supervised graph inference. In *Advances in Neural Information Processing Systems 15*, pages 1433–1440, 2005.
20. F. Wang, S. Ma, L. Yang, and T. Li. Recommendation on item graphs. In *ICDM'06*, pages 1119–1123, Washington, DC, USA, 2006. IEEE.
21. H. C. Zan Huang, Xin Li. Link prediction approach to collaborative filtering. In *JCDL'05*, pages 141–142. ACM/IEEE, 2005.
22. D. Zhou, S. Zhu, K. Yu, X. Song, B. L. Tseng, H. Zha, and C. L. Giles. Learning multiple graphs for document recommendations. In *WWW '08*, pages 141–150, New York, NY, USA, 2008. ACM.

A Matrix Derivatives

To find a solution of Eq. 7, we use the following derivatives of traces.

$$\begin{aligned}\frac{\delta}{\delta \mathbf{X}} \text{Tr}(\mathbf{X} \mathbf{B} \mathbf{X}^T) &= \mathbf{X} (\mathbf{B}^T + \mathbf{B}), \\ \frac{\delta}{\delta \mathbf{X}} \text{Tr}((\mathbf{A} \mathbf{X} \mathbf{B} + \mathbf{C})(\mathbf{A} \mathbf{X} \mathbf{C} + \mathbf{C})^T) &= 2\mathbf{A}^T (\mathbf{A} \mathbf{X} \mathbf{B} + \mathbf{C}) \mathbf{B}^T.\end{aligned}$$

We can see that Eq. 7 is obtained from the above equations and since

$$\begin{aligned}\|\mathbf{Q} - (c\mathbf{Q}\mathbf{M} + (1-c)\mathbf{I})\|_F^2 &= \\ \text{Tr}((\mathbf{Q}(\mathbf{I} - c\mathbf{M}) - (1-c)\mathbf{I})(\mathbf{Q}(\mathbf{I} - c\mathbf{M}) - (1-c)\mathbf{I})^T) &.\end{aligned}$$

B Derivation of \mathbf{Q}_2^* and \mathbf{Q}_4^*

We summarize the derivation of proximity scores incorporating the auxiliary information in the following lemma.

Lemma 1. *Let $\mathbf{\Delta}$ and $\mathbf{\Gamma}$ be the $r \times r$ matrices as defined in Table 2, and \mathbf{Q}_2^* and \mathbf{Q}_4^* be, respectively, the $l \times r$ and $r \times r$ matrices which are the submatrices of \mathbf{Q}^* in Eq. (11). Then, these equations hold*

$$\begin{aligned}\mathbf{Q}_4^* &= (\mathbf{I} - \lambda \mathbf{\Delta} \mathbf{\Gamma}) \mathbf{Q}_4, \\ \mathbf{Q}_2^* &= c \mathbf{D}_L^{-1} \mathbf{G} \mathbf{Q}_4^*.\end{aligned}$$

Proof. Notice that because $\mathbf{Q} = (\mathbf{I} - c\mathbf{M})^{-1}$ and since transposition and inversion are commutative, we have $\mathbf{Q}^{-1} = (\mathbf{I} - c\mathbf{M})$ and $(\mathbf{Q}^T)^{-1} = (\mathbf{I} - c\mathbf{M})^T$. Thus, recalling that $\mathbf{L}' = \mathbf{L}'_1 \mathbf{L}'_2$, we can rewrite Eq. 9 into this form

$$\mathbf{Q}^* = (\mathbf{Q}^T)^{-1} \left(\mathbf{Q}^{-1} (\mathbf{Q}^T)^{-1} + \lambda \mathbf{L}'_1 \mathbf{L}'_2 \right)^{-1}. \quad (12)$$

By the Sherman-Morrison-Woodbury Lemma [14], the second term on the right-hand side of the above equation is equal to:

$$\mathbf{Q}^T \mathbf{Q} - \lambda \mathbf{Q}^T \mathbf{Q} \mathbf{L}'_1 \mathbf{X} \mathbf{L}'_2 \mathbf{Q}^T \mathbf{Q}, \quad (13)$$

where the matrix \mathbf{X} is defined as

$$\mathbf{X} = (\mathbf{I} + \lambda \mathbf{L}'_2 \mathbf{Q}^T \mathbf{Q} \mathbf{L}'_1)^{-1}. \quad (14)$$

This implies that Eq. (12) is equal to

$$\mathbf{Q}^* = \mathbf{Q} - \lambda \mathbf{Q} \mathbf{L}'_1 \mathbf{X} \mathbf{L}'_2 \mathbf{Q}^T \mathbf{Q} \quad (15)$$

Next, notice that by definition $\mathbf{L}' = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{L} \end{pmatrix}$, and therefore, its decomposition can be written as

$$\mathbf{L}'_1 = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{L}_1 \end{pmatrix}, \mathbf{L}'_2 = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{L}_2 \end{pmatrix}.$$

We see that

$$\mathbf{L}'_2 \mathbf{Q}^T = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{L}_2 \mathbf{Q}_2^T & \mathbf{L}_2 \mathbf{Q}_4^T \end{pmatrix}, \mathbf{Q} \mathbf{L}'_1 = \begin{pmatrix} \mathbf{0} & \mathbf{Q}_2 \mathbf{L}_1 \\ \mathbf{0} & \mathbf{Q}_4 \mathbf{L}_2 \end{pmatrix},$$

which implies that

$$\mathbf{L}'_2 \mathbf{Q}^T \mathbf{Q} \mathbf{L}'_1 = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{L}_2 \mathbf{Q}_2^T \mathbf{L}_1 + \mathbf{L}_2 \mathbf{Q}_2^T \mathbf{Q}_2 \mathbf{L}_1 \end{pmatrix}.$$

Therefore, we have found that \mathbf{X} is a block diagonal matrix in the form:

$$\mathbf{X} = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Lambda} \end{pmatrix}, \quad (16)$$

where $\mathbf{\Lambda}$ is defined as in Table 2. Notice that we used the matrix equality $\mathbf{Q}_2 = c\mathbf{D}_L^{-1} \mathbf{G} \mathbf{Q}_4$ in this derivation to obtain $\mathbf{\Lambda}$.

Returning to Eq. (15), we can see that the multiplication of the matrices in the second term of the right-hand side results in

$$\begin{aligned} \mathbf{Q} \mathbf{L}'_1 \mathbf{X} \mathbf{L}'_2 \mathbf{Q}^T \mathbf{Q} &= \begin{pmatrix} \mathbf{Q}_2 \mathbf{L}_1 \mathbf{\Lambda} \mathbf{L}_2 \mathbf{Q}_2^T & \mathbf{Q}_2 \mathbf{L}_1 \mathbf{\Lambda} \mathbf{L}_2 \mathbf{Q}_4^T \\ \mathbf{Q}_4 \mathbf{L}_1 \mathbf{\Lambda} \mathbf{L}_2 \mathbf{Q}_2^T & \mathbf{Q}_4 \mathbf{L}_1 \mathbf{\Lambda} \mathbf{L}_2 \mathbf{Q}_4^T \end{pmatrix} \mathbf{Q} \\ &= \begin{pmatrix} c\mathbf{D}_L^{-1} \mathbf{G} \mathbf{\Delta} \mathbf{G}^T \mathbf{D}_L^{-1} & c\mathbf{D}_L^{-1} \mathbf{G} \mathbf{\Delta} \\ c\mathbf{\Delta} \mathbf{G}^T \mathbf{D}_L^{-1} & \mathbf{\Delta} \end{pmatrix} \mathbf{Q}. \end{aligned}$$

Therefore, it follows that, with regards to \mathbf{Q}_4^* , we have

$$\begin{aligned} \mathbf{Q}_4^* &= \mathbf{Q}_4 - \lambda (c\mathbf{\Delta} \mathbf{G}^T \mathbf{D}_L^{-1} \mathbf{Q}_2 + \mathbf{\Delta} \mathbf{Q}_4) \\ &= \mathbf{Q}_4 - \lambda \mathbf{\Delta} \mathbf{\Gamma} \mathbf{Q}_4, \end{aligned}$$

since $c\mathbf{G}^T \mathbf{D}_L^{-1} \mathbf{Q}_2 + \mathbf{Q}_4 = \mathbf{\Gamma} \mathbf{Q}_4$. With a similar derivation, with regards to \mathbf{Q}_2^* we have,

$$\begin{aligned} \mathbf{Q}_2^* &= \mathbf{Q}_2 - \lambda c\mathbf{D}_L^{-1} \mathbf{G} \mathbf{\Delta} \mathbf{\Gamma} \mathbf{Q}_4 \\ &= c\mathbf{D}_L^{-1} \mathbf{G} (\mathbf{Q}_4 - \lambda \mathbf{\Delta} \mathbf{\Gamma} \mathbf{Q}_4) \\ &= c\mathbf{D}_L^{-1} \mathbf{G} \mathbf{Q}_4^*. \end{aligned}$$

This completes the proof of the lemma.