Bi-Relational Network Analysis Using a Fast Random Walk with Restart

Jing Xia, Doina Caragea and William Hsu
Departmet of Computing and Information Sciences
Kansas State University, Manhattan, USA
{*xiajing, dcaragea, bhsu*}*@ksu.edu*

***Abstract*—Identification of nodes relevant to a given node in a relational network is a basic problem in network analysis with great practical importance. Most existing network analysis algorithms utilize one single relation to define relevancy among nodes. However, in real world applications multiple relationships exist between nodes in a network. Therefore, network analysis algorithms that can make use of more than one relation to identify the relevance set for a node are needed. In this paper, we show how the Random Walk with Restart (RWR) approach can be used to study relevancy in a bi-relational network from the bibliographic domain, and show that making use of two relations results in better results as compared to approaches that use a single relation. As relational networks can be very large, we also propose a fast implementation for RWR by adapting an existing Iterative Aggregation and Disaggregation (IAD) approach. The IAD-based RWR exploits the block-wise structure of real world networks. Experimental results show significant increase in running time for the IAD-based RWR compared to the traditional power method based RWR.**

***Keywords*-Relational data mining; node relevancy; random walk; iterative aggregation and disaggregation approach.**

I. Introduction

Identification of nodes relevant to a given node in a relational network is of significant practical importance. Node relevancy information enables the study of complex properties of a network. As an example, in a bibliographic network, information about researchers relevant to a given researcher can be used to predict potential co-author relationships or communities that a researcher should join.

Among several approaches to the problem of identifying a relevancy set for a given node in a network, random walk based algorithms have proven very effective [1]. Traditionally, random walk algorithms exploit one type of relation (e.g., co-author relationships) when finding relevance scores for a node (in our example, an author).

However, in real world applications, for a particular domain there always exist several types of objects (e.g., papers, authors, venues) and relations among objects of interest (e.g., co-author relationships, citation relationships, author-paper relationships). While each relation can be exploited by itself for solving a particular network analysis task, more insights into the properties of the network can be gained if multiple relations are used together. In this work, we focus on bi-relational network analysis using a Random Walk with Restart (RWR) approach and show its advantages as compared to single relational network analysis.

Given the large scale of network data available nowadays, fast implementations of the RWR algorithms are needed even in the case of single relational network analysis [2]. For the analysis of networks with two or more relations, time and memory efficient algorithms are imperative. To address this challenge, we propose a fast implementation of the RWR algorithm for bi-relational networks. This implementation takes advantage of a nice property that real world networks present, specifically their block-wise structure. Based on this property, an *iterative aggregation and decomposition* (IAD) algorithm is adapted to RWR.

The rest of the paper is organized as follows. We define relevance scores and introduce single and bi-relational networks in Section II. We introduce the RWR network analysis approach in Section III. Our adaptation of the RWR approach to bi-relational networks is presented in Section IV, while the IAD-based RWR and a discussion on the efficiency of the method are presented in Section V. Section VI describes the experimental evaluation of the proposed approach. We conclude with a summary and discussion of the related work in Section VII.

II. Background and Motivation

A. Relevance of Nodes

Generally, a relational network can be represented as a graph $G =< V, E >$, where V is the set of nodes and E is the set of edges representing relationships between nodes in the network. Similar to previous work [1], the main question we address in this paper can be stated as follows: given a node $a \in V$, which nodes in V are most relevant to a? To answer this question, for each node $b \in V$, we use RWR to compute a relevance score to a. All scores together form a relevance score vector with respect to the node a. We expect nodes that are highly relevant to a to have higher scores than nodes that are not relevant to a. Thus, the score vector can help identify nodes relevant to a and also quantify relevancy.

B. Bi-Relational Networks

We will use a simple academic community example to motivate and describe bi-relational networks. In a typical academic community, given a researcher a, one may be interested in finding the most relevant researchers b for a. Here, we assume that b is relevant to a if a and b share similar research interests. The set of nodes for this example

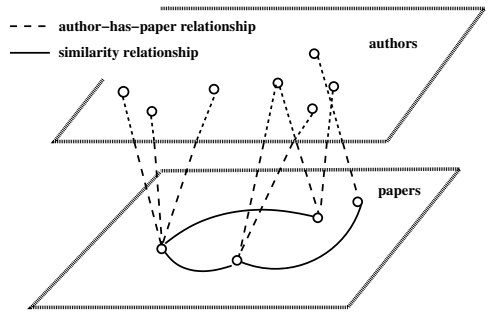

Figure 1. Bi-relational network: authors are associated with papers through author-has-paper relationships; papers are linked through similarity relationships.

consists of researchers and papers. In principle, we can compute the relevancy score vector for researcher a based on co-author relationships (single-relational network) in a bibliographic data set. However, we can also build a bi-relational network in which papers can be linked to each other based on content similarity relationships (i.e., two papers are similar if they share similar words); authors can be linked to papers through author-has-paper relationships. Figure 1 shows a view of the bi-relational network induced by these two relationships. Intuitively, authors who have similar papers share similar interests. Thus, using these relationships together to find authors most relevant to a given author should result in a stronger relevancy compared with the relevancy obtained using each relation independently.

We will use the following definitions for single and bi-relational networks:

- A *single-relational network* is a network induced by a single relation among nodes. If $G =< V, E >$ is the graph corresponding to the network, then E is the set of edges defined by the single relation among nodes V.
- A *bi-relational network* is a network induced by two types of relations among nodes. Formally, a bi-relational network is given by $G =< V_1 \cup V_2, E_1 \cup E_2 >$, where $E_1 \subset V_1 \times V_2$ and $E_2 \subset V_2 \times V_2$.

III. Random Walk with Restart

The notations used in this paper are shown in Table I. The RWR method defines a transition matrix $P_{n\times n}$ (where n is the number of nodes). This matrix models the probability of transition between every two nodes in the network. If P is row normalized (i.e., the sum of elements in a row is 1), then P is irreducible and aperiodic. Therefore, according to the Perron-Frobenius theorem, there is a unique stationary distribution of the matrix P.

Given the transition matrix P, a RWR can be seen as a non-homogeneous Markov chain. A RWR is defined by the following formula:

$$\pi_{(k)}^{(t+1)} = (1-c)\pi_{(k)}^{(t)} P + c \cdot e_k \qquad (1)$$

Table I
Symbols and Definitions

Symbol	Definition
$\pi_{(k)}$	$1 \times n$ stationary distribution vector by running RWR from starting node k
$\pi_{(k)}^{(t)}$	distribution vector after t iterations runs
c	the restart probability, $0 < c < 1$
e_k	$1 \times n$ starting vector, the k^{th} element is 1 and all the other elements are 0
c_k	$1 \times n$ starting vector $c_k = ce_k$
n	the number of nodes in the graph
N	the number of partitions
P	the original transition matrix
M	the transformed transition matrix, $M = (1-c)P$
A	$N \times N$ coupling matrix of M
m_i	the size of each sub matrix M_{ii}
c'_k	$1 \times m_k$ sub-starting vector for the sub-matrix M_{kk} where one element corresponding to the starting node k is 1 and 0 for others
π_i	sub-eigenvector of sub-matrix M_{ii}
$c_{1\times N}$	$1 \times N$ vector where one element corresponding to M_{kk} is 1 and the others are 0

where $\pi_{(k)}$ is the probability distribution of a particle starting at node k, $c \in (0, 1)$ is the restarting probability, and e_k is the initial vector.

As can be seen in the equation, at each iteration, a constant $c \cdot e_k$ interpreted as the "restart" is added. Eq. (1) converges as the number of iterations approaches infinity [3]. Therefore, we have:

$$\pi_{(k)} = (1-c)\pi_{(k)}P + c \cdot e_k = \pi_{(k)}M + c_k \qquad (2)$$

The stationary distribution $\pi_{(k)}$, which represents the probability distribution of reaching any node a from k, can be seen as the relevance score vector corresponding to k.

IV. RWR for Single and Bi-Relational Networks

In what follows, we explain how we apply RWR to single and bi-relational networks, respectively.

A. Single-Relational Networks

Obviously, we can directly apply RWR for single relational networks. To do that, P is constructed from the network $G =< V, E >$ (which is a weighted graph); e_a represents the vector starting with node a. The stationary distribution $\pi_{(a)}$ can be used as the relevance score vector corresponding to the node a.

B. Bi-Relational Networks

Remember that a bi-relational network is defined by $G =< V_1 \cup V_2, E_1 \cup E_2 >$, where $E_1 \subset V_1 \times V_2$ and $E_2 \subset V_2 \times V_2$. Our goal is to use RWR to identify nodes b in V_1 relevant to a node a in V_1 by making use of both relationships in E_1 and E_2. To achieve that, we construct the transition matrix P from $< V_2, E_2 >$. Then, for the given node $a \in V_1$ and every edge $(a, p) \in E_1$ (consequently for every node $p \in V_2$ that is linked to a), we run RWR with

transition probability P and starting vector e_p. The resulting stationary distribution represents the relevance score vector corresponding to the starting node p. Based on the relevance score vector, we choose a set of nodes $V_2' \subset V_2$ that are most relevant to node p. Finally, the nodes relevant to a are defined as those nodes b in V_1 for which there exists an edge (b, q) (where $q \in V_2'$).

For example, let us assume that an author a has four papers. These papers are part of a paper similarity network. To identify authors related to a, we run RWR with a transition matrix given by the paper similarity network and starting vectors corresponding to each of the author's papers, in turn. Thus, we will obtain a set of four relevance score vectors. From each vector, we choose the most related papers and infer that their authors are most relevant for the given author.

V. Scaling Up RWR

The straightforward implementation of RWR requires many iterations over the transition matrix or, even worse, calculating the inverse of the matrix [2]. As multi-relational networks are usually large, using this implementation is impractical for most real world applications. To address this limitation, we propose an approach for scaling up the RWR method. The theory behind our fast RWR approach and the algorithm used in our experiments are described in what follows.

A. RWR Property

In this subsection, we will show a nice property of $\pi_{(k)}$ (the stationary distribution of the RWR starting at k), assuming that the matrix M in (2) has the following diagonal block-structure:

$$M = \begin{pmatrix} M_{11} & 0 & ... & 0 \\ 0 & M_{22} & ... & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & ... & M_{N,N} \end{pmatrix} \quad (3)$$

where each block sub-matrix M_{ii} is of size m_i, for $i = 1, 2, ..., N$ and M_{kk} contains the starting node k. Then, by replacing M with (3) in (2), we get:

$$(\pi_1, ..., \pi_k, ..., \pi_N) = (\pi_1, ..., \pi_k, ..., \pi_N) \times \begin{pmatrix} M_{11} & ... & 0 & ... & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & ... & M_{kk} & ... & 0 \\ \vdots & .... & \vdots & \ddots & \vdots \\ 0 & ... & 0 & ... & M_{NN} \end{pmatrix} + \begin{pmatrix} 0 \\ \vdots \\ c_k' \\ \vdots \\ 0 \end{pmatrix}^T$$

where π_k is the sub-eigenvector for the block sub-matrix M_{kk} and c_k' is a $(1 \times m_k)$ vector corresponding to the sub-matrix M_{kk} (the element corresponding to the starting node k is 1 and all the other elements are 0). As a consequence, each π_i can be obtained from

$$\pi_i M_{ii} = \pi_i, \; i \neq k \quad \text{and} \quad \pi_k M_{kk} + c_k' = \pi_k$$

Note that $\rho(M_{ii}) < 1$ (ρ is the spectral radius of a matrix, i.e. the max eigenvalue of a matrix), therefore $\pi_i = 0$, for $i = 1, 2, ..., N$, $i \neq k$. Thus, we only need to solve the equation $\pi_k M_{kk} + c_k' = \pi_k$. This property explains the observation made in [1], where the authors noticed that most elements in the distribution are close to zero and therefore proposed to perform RWR on the partitioned local block only.

In most real network applications, the network naturally forms a block-wise structure, although not necessarily a perfect diagonal block-structure like the one above. For instance, in the academic community network example, the author network has a block-wise community structure with respect to authors' interests and publications. Similarly, the papers network has a block-wise structure with respect to papers' topics and similarity. We will exploit this type of structure to scale up the RWR approach. To do that, we first construct a block-wise partition for M that looks like:

$$M = \begin{pmatrix} M_{11} & M_{12} & ... & M_{1N} \\ M_{21} & M_{22} & ... & M_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ M_{N1} & M_{N2} & ... & M_{NN} \end{pmatrix} \quad (4)$$

where M_{ii} represents the links within a "community" i and $M_{ij}, i \neq j$ represents the links between "communities" i and j. To construct such a partition for M, we use CLUTO [4] algorithm, which maximizes the edge weight within the community, while minimizing the weight between communities. Once a partition of M is constructed, we can compute the left eigenvector for each diagonal sub-matrix M_{ii}:

$$u_i M_{ii} = \lambda_i u_i (\text{for } i \neq k) \quad \text{and} \quad u_k M_{kk} + c_k' = u_k \quad (5)$$

where $\lambda_i \leq (1 - c)$. We will use the eigenvectors u_i of M_{ii} as an approximation to π_i (the sub-vector in $\pi_{(k)}$ corresponding to M_{ii}) and further combine the u_i local eigenvalue vectors into one global eigenvector for the whole matrix by adapting the Iterative Aggregation/Disaggregation (IAD) [3], [5], [6] method. This will allow to quickly find the steady distribution $\pi_{(k)}$.

B. Fast RWR Using the IAD Method

The combination of the local eigenvectors corresponding to matrices M_{ii} into a global eigenvector for M needs to take into account the weight of each sub-block matrix.

The first part of the IAD algorithm is used to derive this weight vector by constructing an aggregated matrix A from M, in two steps. Assuming that π_i is known for $i = 1, 2, ..., N$, the two steps are as follows: 1) replace each row of each sub-block matrix M_{ij} with the sum of the elements in that row; this results in a matrix $n \times N$ (one column for each M_{ij}); 2) multiply each of the resulting columns by a weight vector ϕ_i, where $\phi_i = \pi_i / ||\pi_i||_1$, for

$i = 1, 2, ..., N$; this results in an aggregated matrix $A_{N\times N}$ (one element for each sub-block matrix M_{ij}).

The elements of the matrix, $A_{N\times N}$ can be written as: $a_{ij} = \phi_i M_{ij} e_j$, where ϕ_i is a row vector with m_i elements and e_j is the **1** column vector with m_j elements. Having constructed the aggregated matrix A, the goal is to find a weight vector for each sub-block matrix M_{ij} by solving the equation $\xi = \xi A + c_{1\times N}$. Indeed, we can show that A has such a stationary distribution. This follows from:

$$(||\pi_1||_1, ||\pi_2||_1, ..., ||\pi_N||_1)A = (||\pi_1||_1, ||\pi_2||_1, ..., ||\pi_N||_1)$$
$$\times \begin{pmatrix} \frac{\pi_1}{||\pi_1||_1} & 0 & ... & 0 \\ 0 & \frac{\pi_2}{||\pi_2||_2} & ... & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & ... & 0 & \frac{\pi_N}{||\pi_N||_1} \end{pmatrix} Me_{n\times N} + c_{1\times N}$$
$$= \pi_{(k)} Me_{n\times N} + c_{1\times N} = (\pi_{(k)} - c_k) \begin{pmatrix} e & 0 & ... & 0 \\ 0 & e & ... & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & ... & 0 & e \end{pmatrix} + c_{1\times N}$$
$$= (||\pi_1||_1, ||\pi_2||_1, ..., ||\pi_k - c'_k||_1, ..., ||\pi_N||_1) + c_{1\times N}$$
$$= (||\pi_1||_1, ||\pi_2||_1, ..., ||\pi_k||_1, ..., ||\pi_N||_1).$$

We used the fact that $\pi_{(k)} M = (\pi_{(k)} - c_k)$ (Eq. 1) and $||\pi_k - c'_k||_1 + c = ||\pi_k||_1$ (which can be easily proved using the definition of the norm 1). If $\xi = (||\pi_1||_1, ||\pi_2||_1, ..., ||\pi_k||_1, ..., ||\pi_N||_1)$ is the stationary distribution of A, we consider this to be the weight vector for the sub-block matrix M_{ii}.

Note that ϕ_i $(i = 1, 2, ..., N)$ depends on is π_i, which is not known in advance; therefore, we will use u_i as an approximation for π_i. For practical problems, this approximation should not result in a significant error as the structure of M is presumably close to the structure in Eq. (3) and $||M_{ii}||_1$ is maximized when creating the block-wise partition of M. Therefore, an approximation is made such that

$$\phi^*_i = u_i / ||u_i||_1 \approx \phi_i = \pi_i / ||\pi_i||_1 \tag{6}$$

We use Eq. (6) to compute an approximation A^* to the aggregated matrix A. Each element of A^* is given by $a^*_{ij} = \phi^*_i M_{ij} e_j$. Next, we determine an approximation eigenvector ξ^* from $\xi^* A^* + c_{1\times N} = \xi^*$ and use it to derive the stationary distribution of M:

$$\pi^*_{(k)} = \left(\xi^*_1 \phi^*_1, \quad \xi^*_2 \phi^*_2, \quad ..., \quad \xi^*_N \phi^*_N\right) \tag{7}$$

The second part of the IAD is used to improve the approximation in Eq. (7). The simplest way to do this is to incorporate Eq. (7) back into Eq. (6) and reiterate with the goal of obtaining a better solution. However, directly using Eq. (7) will have no effect on the approximation [3].

Therefore, similar to [3], we adapt Takahashi's approach [7] to improve the approximation before incorporating Eq. (7) back into Eq. (6). We will construct a matrix W_i, $i = 1, 2, ..., N$ such that

$$W_i = \begin{pmatrix} M_{ii} & s_i \\ r_i^T & q_i \end{pmatrix} \tag{8}$$

where r_i^T is a $1 \times m_i$ vector defined as:

$$r_i^T = \begin{cases} \frac{1}{1-\xi_i} \sum_{j\neq i} \xi_j \phi_j M_{ji} & \text{if i} \neq \text{k} \\ \frac{1}{1-\xi_k} (c_k + \sum_{j\neq k} \xi_j \phi_j M_{jk}) & \text{if i = k} \end{cases} \tag{9}$$

s_i is an $m_i \times 1$ vector defined as: $s_i = e_i - M_{ii} e_i$ for $i = 1, 2, ..., N$ and q_i is a scalar defined as: $q_i = 1 - r_i^T e$ for $i = 1, 2, ..., N$. Therefore, we obtain:

$$(\pi_i, 1 - \xi_i) \begin{pmatrix} M_{ii} & s_i \\ r_i^T & q_i \end{pmatrix} = (\pi_i, 1 - \xi_i) \tag{10}$$

With the constructed block W_i, we can obtain a new π_i and ξ_i through solving the Eq. (10). Finally we update $\phi_i = \pi_i / \xi_i$ with the new values for π_i and ξ_i obtained from W_i. The steps for scaling up RWR are shown in Algorithm 1.

Algorithm 1 IAD-based RWR

Input: a normalized matrix P, the starting vector e_k and the error threshold ϵ

Output: the stationary distribution $\pi_{(k)}$

1. Construct the transformed matrix M from P.

2. Partition M into N partitions using CLUTO [4].

3. Let $\pi^{(0)}_{(k)} = (\pi^{(0)}_1, \pi^{(0)}_2, ..., \pi^{(0)}_N)$ be a given initial approximation to the solution and set $m = 1$.

4. For $i = 1, 2, ..., N$, compute $\phi^{(m-1)}_i$ as:

$$\phi^{(m-1)}_i = \pi^{(m-1)}_i / ||\pi^{(m-1)}_i||_1$$

5. Construct aggregated matrix $A^{(m-1)}$ whose elements are

$$(A^{(m-1)})_{ij} = \phi^{(m-1)}_i M_{ij} e_j$$

6. Solve the eigenvector problem

$$\xi^{(m-1)} A^{(m-1)} + c_{1\times N} = \xi^{(m-1)}$$

7. For $i = 1, 2, ..., N$, construct W_i and derive $\pi^{(m)}_i$ and $\xi^{(m)}_i$ by solving Eq. (10); update $\phi^{(m)}_i = \pi^{(m)}_i / \xi^{(m)}_i$

8. Convergence test: if the the difference between two consecutive estimates $||\pi^{(m)} - \pi^{(m-1)}||_2 < \epsilon$, then stop

$$\pi^{(m)}_{(k)} = (\xi^{(m-1)}_1 \phi^{(m)}_1, \xi^{(m-1)}_2 \phi^{(m)}_2, ..., \xi^{(m-1)}_N \phi^{(m)}_N)$$

Otherwise, set $m = m + 1$ and go to step (4).

C. Efficiency of the IAD-based RWR

IAD-based RWR is a divide-and-conquer method which takes advantage of the block-wise structure of real world

networks. The running time of the algorithm depends mainly on two factors: number of iterations and, for each iteration, the time it takes to solve the Eq. (10) for N block sub-matrices. Solving Eq. (10) takes time proportional to the size of the matrix M_{ii}). The CLUTO algorithm that we use to partition M takes as input the number N of blocks needed and optimizes block size to avoid partitions with a lot of small blocks and several large block. Thus, the resulting partitions are well suited for the IAD approach. The global convergence of the IAD method is still an open problem. However, we will show that for real world networks that have a natural block-wise structure the algorithm converges very fast. As for space, the algorithm stores the diagonal matrices and the sparse matrix of the cross-link network. The aggregated matrix requires $O(N^2)$ space.

VI. Experimental Evaluation

A. Data Sets and Questions

The data set used for the experiments in this paper (called *paper & co-author* data) is constructed from the Cora data set http://www.cs.umass.edu/~mccallum, which contains research papers. For each paper, the following fields are available: title, authors, topic, abstract, avenue (e.g., conference name), among others.

The data set is obtained from Cora as follows: We first extract publications for which title and authors' names are available. From the resulting set of papers, we select those for which abstracts are available. This results in a data set that contains 4,100 papers and 10,830 authors.

Two networks are constructed from this data. First, we construct a single relational network $G =< V, E >$ based on the co-author relation. The weight of an edge (a, b) from author a to author b is defined as the number of publications co-authored by a and b, divided by the total number of the publications authored by a.

Second, we construct a bi-relational network $G =< V_1 \cup V_2, E_1 \cup E_2 >$ based on author-has-paper and paper similarity relations. The weight on an edge $(a, p) \in V_1$ from an author a to a paper p is 1 if a is among paper's p authors and 0 otherwise. The weight of an edge $(p, q) \in E_2$ between two papers p and q is given by the cosine similarity between the abstracts of the two papers. To calculate cosine similarity we build an inverted index over the merged vocabulary of all abstracts. Using the inverted index, each abstract is represented using the TF-IDF (term frequency, index document frequency) weighting scheme.

The questions that we want to answer about the *paper & co-author* data set are the following: **(Q1)** *What are the most relevant authors to an author* a*, as identified through the analysis of the single and bi-relational networks, respectively*? Intuitively, the more similar the papers that two authors share, the more related the authors are. **(Q2)** *How accurate is the process of mining information from the single and bi-relational networks, respectively*?

B. Experimental Design and Results

We answer **(Q1)** through a case study. We run RWR on the single and bi-relational networks described in section VI-A, respectively, to get relevance score vectors. We use the author *Jiawei Han* as a starting node. Table II left column shows the top 10 relevant authors for Jiawei Han, as identified from the single-relational co-author network. As expected, these are mostly his collaborators (researchers that have co-authored papers with him). Table II righ column shows the top 10 relevant authors for Jiawei Han, as identified from the bi-relational network. These are researchers whose interests are similar to Jiawei Han's interests (specifically, database and data mining). This case study shows the advantage of using the bi-directional network in the analysis: it produces results that can be used for predicting potential future collaborations or even potential reviewers for a researcher.

Table II
Authors relevant to Jiawei Han

Single-relational network		Bi-relational network	
n. stefanovic	0.02252	l. lakshmanan	0.23718
j. chiang	0.01853	t. topaloglou	0.05464
w. gong	0.01853	j. mylopoulos	0.04723
b. xia	0.01853	r. missaoui	0.03594
o. r. zaiane	0.01853	r. ramakrishnan	0.02847
m. kamber	0.01787	h. hirsh	0.02330
l. lakshmanan	0.01684	s. sudarshan	0.02099
k. koperski	0.01520	m. j. zaki	0.02015
w. wang	0.01442	a. j. bonner	0.01487
a. pang	0.01431	d. srivastava	0.01453

We conduct two experiments to answer **(Q2)**. The first experiment **(Q2.1)** is designed to evaluate the accuracy of the process of labeling papers with categories using the similarity network only. The second experiment **(Q2.2)** is designed to test the accuracy of the process of predicting co-authors based on the bi-relational network.

(Q2.1) In the Cora data set, each paper has a label indicating the research category associated with the paper. We consider the k-th most related papers to a given paper in the data set (according to their relevance scores). The accuracy is defined as the number of papers categorized in the same category as the given paper, divided by k. Figure 2 shows the results. As expected, the accuracy decreases with the number of papers k considered.

(Q2.2) To test the accuracy of the process of predicting co-author relationships from the bi-relational network, we randomly select three distinct sets of author pairs. The authors in a pair have co-authored some papers, which are removed from the network. We assume that in addition to the papers that a pair of authors have co-authored (removed), the two authors might have published other similar papers. Our intuition is that if a pair of authors share similar papers, then they will be predicted to be co-authors based on the bi-relational author-paper network.

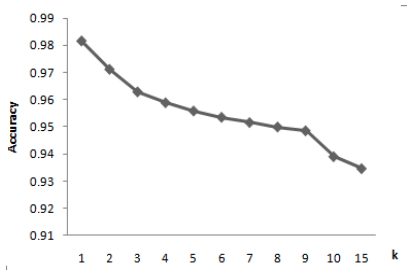

Figure 2. Paper labeling prediction accuracy, as a function of the total number k of papers for which labels are predicted, based on mining the paper similarity network.

For each pair, we run RWR staring at an arbitrary paper of one of the authors in the pair (this will not be a co-authored paper, as those have been removed) and identify the k-th most related authors. A pair is predicted correctly if the co-author in the pair is among the related authors. We define the accuracy as the number of co-authors identified divided by the number of pairs in a data set. Figure 3 shows the results. As expected, the more related authors k are retrieved, the better the prediction accuracy.

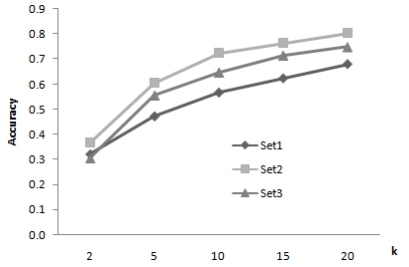

Figure 3. Co-author prediction accuracy, as a function of the number k of authors retrieved for each pair of potential co-authors, based on mining the bi-relational network.

C. Efficiency of Random Walk

Table III shows a comparison between the traditional power method (which multiplies the transition matrix with π until the L_2 norm of successive estimates of π goes below the threshold ϵ) and the IAD-based RWR in terms of the $||\pi^{(1)} - \pi^{(0)}||_2$ values after the first iteration and number of steps to convergence. Results for three partitions are shown.

Table III
VALUES OF $||\pi^{(1)} - \pi^{(0)}||_2$ AFTER THE FIRST ITERATION AND NUMBER OF ITERATIONS FOR CONVERGENCE WHEN $\epsilon = 10^{-5}$, FOR THE TRADITIONAL POWER METHOD VS. IAD APPROACH.

	$\|\|\pi^{(1)} - \pi^{(0)}\|\|_2$		# Iterations	
# Partitions	Power	IAD	Power	IAD
1000	$4.77e^{-2}$	$1.79e^{-4}$	55	2
2000	$2.24e^{-2}$	$1.59e^{-2}$	52	5
4000	$9.01e^{-2}$	$1.33e^{-2}$	88	6

VII. SUMMARY AND RELATED WORK

In this paper, we have shown how to use the RWR approach to analyze a bi-relational network. A similar analysis has been performed for networks with three relations, but was omitted here due to space limitations (to be published as a technical report). Generalization of our approach to multi-relational networks is possible, according to the semantics of the relations in a particular network.

We have also proposed an IAD-based fast RWR implementation. This implementation makes use of the block-wise structure that many networks present. Experimental results on a data set from the bibliographic domain show the benefits of using bi-relational networks as opposed to single networks. The relevance scores defined by RWR have many useful properties. Compared with other pairwise metrics, the relevance scores can capture the global structure of the graph as well as the multi-facet relationships between nodes.

RWR is a popular method for network analysis. Many applications use random walk and related methods as a building block. Tong *et al.* [2] provides an excellent review of RWR. An exact solution for RWR usually requires the inversion of a large matrix. Therefore, fast approximate solutions to the problem have been proposed before [1]. Similar to our proposed approach, other existing solutions make use of the block-wise structure of real world networks. Tong et al. [2] approximate the stationary distribution of RWR by heuristic-based low rank approximation. Sun et al. [1] perform RWR only on the partition that contains the starting point. Their method outputs a local estimation of the stationary distribution.

REFERENCES

[1] J. Sun, H. Qu, D. Chakrabarti, and C. Faloutsos, "Neighborhood formation and anomaly detection in bipartite graph," *Proc. of the Sixth Int. Conf. on Data Mining (ICDM '05)*, pp. 418–425, 2005.

[2] H. Tong, C. Faloutsos, and J.-Y. Pan, "Fast random walk with restart and its applications," *Proc. of the Sixth Int. Conf. on Data Mining (ICDM'06)*, vol. 418, pp. 613–622, 2006.

[3] W. J. Stewart, *Introduction to the Numerical Solution of Markov Chains*. Princeton University Press, 1996.

[4] G. Karypis and V. Kumar, "Multilevel k-way partitioning scheme for irregular graphs," *J. Parallel Distrib. Comput.*, vol. 48, no. 1, pp. 96–129, 1998.

[5] W. J. Stewart, "Numerical methods for computing stationary distribution of finite irreducible markov chains," *J. of Adv. in Computational Probability*, vol. 418, pp. 613–622, 1998.

[6] D. P. O'Leary, "Iterative methods for finding the stationary vector for markov chains," *Linear Algebra, Markov Chains, and Queuing Models*, vol. 418, pp. 613–622, 1992.

[7] Y. Takahashi, "A lumping method for numerical calculations of stationary distributions of markov chains." *Technical Report B-18, Tokyo Inst. of Tech.*, vol. 418, pp. 613–622, 1975.