Invited Paper

Estimating the Relative Importance of Nodes in Social Networks

Heyong Wang[1,a)] Carl K. Chang[1,b)] Hen-I Yang[1,c)] Yanping Chen[2,1,d)]

Received: November 2, 2012, Accepted: March 20, 2013

Abstract: In social networks, nodes usually represent people and edges represent the relationship and connections between people. Ranking how important the nodes are with respect to some query nodes has a lot of applications in social networks. More often, people are interested in finding the *Top-k* most "relatively important" nodes with respect to some query nodes. A major challenge in this area of research is to define a function for measuring the "relative importance" between two nodes. In this paper, we present a measure called *path probability* to represent the connection strength of a between the ending node and the starting node. We proposed a measure of relative importance by using the sum of the path probabilities of all the "important" paths between a node with respect to a query node. Another challenge of computing the relative importance is the scalability issue. Most popular solutions are random walk based algorithms which involve matrix multiplication, and therefore are computationally too expensive for large graphs with millions of nodes. In this paper, by defining the *path probability* and introducing a small threshold value to determine whether a path is important or significant, we are able to ignore a lot of unimportant nodes so as to be able to efficiently identify the *Top-k* most relatively important nodes to the query nodes. Experiments are conducted over several synthetic and real graphs. The results are encouraging, and show a strong correlation between our approach and the well known random walk with restart algorithm.

Keywords: social networks, relative importance, path probability, random walk

1. Introduction

The notion of social networks and the methods of social networks analysis (SNA) have attracted considerable research interest for a long time [1]. This long–term interest has translated into practical use in many socio–economic applications such as emergency social network [2], education [3], epidemic network [15], [16], and many economic applications [4], etc. Many popular social networking applications on the Web, such as MySpace, Facebook, LinkedIn, and Bebo, have attracted hundreds of millions of users, many of whom have integrated these sites into their daily routines.

A social network is a complex network with a set of actors engaging in relationships with one another. To perform SNA, visualization techniques such as graph-drawing can be very useful for gaining qualitative insight about the structure of graphs; meanwhile, there is also an urgent need for quantitative tools to characterize graph properties beyond simple lists of "who is connected to whom." Many quantitative methods have been used in social network analysis including graph, matrix algebra, and sociometry [5]. Different quantitative methods are suitable for different application areas. Often used in computer science research, graph has been widely adopted to represent social network due to its intuitive illustration [6]. From the perspective of graph theory, nodes represent actors, edges represent relations between nodes, and relations between nodes can be either simplex or multiplex [17].

In a social network graph, there always exists relationship between some nodes, and current research focuses more on the relations among a set of nodes than individual nodes and their attributes [6]. For example, *Top-k* is one kind of problems to measure the relations among nodes [7], [8], [14], [25]. In our view, there are mainly two kinds of *Top-k* problems in existing social network graphs. One is to identify the *Top-k* matched graphs that contain a query graph from a set of graphs. Another is to find the *Top-k* nodes that are relatively important to some query nodes in a large graph. The second question has an extensive significance in practice. For instance, researchers new to a field often face a challenge of "reading in" that field. In this context, they need to be able to locate the classical papers that initiated inquiry into that domain, and what they plan to read should afford them a sufficient understanding of various pertinent topics. It can be really a hard work due to potentially a very large number of existing literatures. These tasks currently can be accomplished with the help of social network analysis whereas authors can be represented as nodes and the co-author relationships can be represented as edges. Based on such a network structure, we can recommend the *Top-k* relevant authors and papers for further reading according to a given set of authors in this paper, we pay our main attention on how to find the *Top-k* relatively important nodes with respect to

1 Department of Computer Science, Iowa State University, Ames, Iowa 50011, United States
2 School of Computer Science, Xi'an University of Posts and Telecommunications, Xi'an, Shaanxi 710121, China
a) wheyong@gmail.com
b) chang@iastate.edu
c) hiyang@gmail.com
d) chenyanping@xupt.edu.cn

some query nodes.

To do that, we should accomplish two important steps - one is to impose a metric upon measuring the strength of relations, and the other is to rank nodes according to certain user requirements. Centrality and prestige can be quite useful when evaluating relative importance. The three most widely used centrality measures are degree, closeness, and betweenness [6], [10], [11]. Many graph-theoretic centrality concepts are discussed in Hage and Harary [12]. There are also varying methods with respect to prestige including degree prestige and proximity prestige [9]. As to ranking, one way is to rank the nodes according to its characteristics (e.g., price and size of house objects in a real estate database, or color and texture of images in a multimedia database); and another way is making use of random walk [13]. Markov chain-based models have been proven successful in ranking web pages [14], [25]. PageRank [14], [19], [22] and Hits [24], [25] are well-known approaches that attempt to computationally measure the global importance of individual nodes in directed graphs. Research in this area has since been very active in developing a variety of extensions and new algorithms.

Based on the above analysis, in this paper, we focus on finding the *Top-k* nodes with respect to a set of query nodes. In pursuit of this goal, firstly, we present a new measure called *path probability* to estimate the importance of a path to a query node (the starting node) based on the probabilities of random walk, and then, the relative importance of a node t with respect to s defined as the sum of all significant path probabilities from s to t. Finally, experiments on both a toy data set and two real data sets are conducted. Experimental results are reported in Section 6.

The rest of the paper is organized as follows: In Section 2, related work of this research is discussed. Section 3 gives the problem formulation. Section 4 defines path probability and how it is calculated. In Section 5, we present a relative importance algorithm called *SigPathSum* to find *Top-k* nodes of interest. In Section 6, the results of experiments conducted on three data sets are presented. We conclude our work in Section 7.

2. Related Work

Measuring the importance of nodes in graphs or networks has been studied for a long time, especially in the social networks, link analysis [39], and bibliometrics areas [40]. Most of the work focuses on ranking the nodes globally in the whole networks or graphs. Freeman [10], [11], [26] first defined a set of measures to describe the global importance of nodes in the social networks called centrality. He proposed three measures of centrality: one is based on degree, and the other two are based on the shortest paths between pairs of nodes. Several other centrality measures [11], [21], [26] were later proposed in order to represent how central someone is in the disease–spreading social network. Kerrebroeck et al. [36] introduced loop ranking, a new ranking measure based on the detection of closed paths that can be computed relatively efficiently. Using degree or shortest paths for ranking nodes may work for some social networks, but it ignores other paths that might be important in other social networks. Our idea of ranking nodes is based on a set of weighted paths between nodes.

The idea of using weighted paths to approximate global measures of importance has also been studied for a long time in the literature of social networks. Katz [28] introduced a way to measure the degree of influence of an actor in a social network by taking into account the total number of walks between a pair of actors. Stephenson and zelen [29] defined a similar measure called information centrality that is based on the information contained in all possible paths between pairs of nodes. In addition, in social networks and bibliometrics areas, work has been carried out on using the principal eigenvector of a matrix derived from the underlying graph to measure the importance of nodes in the networks. Much of this work can be seen as precursors to the eigenvector methods for web page ranking in the web graphs literature.

The two seminal contributions for ranking nodes in web graphs are PageRank algorithm by Page and Brin [19], [22] and HITS algorithm by Kleinberg [24], [25]. Both algorithms have been widely used in ranking the global importance of web pages in some applications. A lot of variants of PageRank and HITS have been developed since then. Lempel and Moran [30] described a variation of HITS called SALSA that can be understood as a random walk on a bipartite graph of hubs. Borodin et al. [31] described a number of algorithms for ranking nodes in a web graph, including extensions of both SALSA and HITS.

Unlike the algorithms discussed above focusing on global measure of node importance, some algorithms derived from PageRank and HITS try to rank the relative importance of nodes with respect to some other nodes. Haveliwala [23] and Jeh and Widom [32] developed the personalized ranking algorithms by extending the PageRank algorithm. Chang, Cohn, and Macallum [33] presented a personalized variant of HITS algorithms. White and Smyth [27] proposed a new framework and class of techniques for measuring relative importance in the networks. They introduced newly-defined measures of node importance and extensions of techniques previously proposed. Fouss et al. [35] presented a new perpective on characterizing the similarity between elements on graphs for the recommendation systems by using the average commute time that is the pseudoinverse of the laplacian matrix of the graph. However, in order to find the *Top-k* nodes relatively important to some query nodes, all the relative importance algorithms discussed above have to use matrix multiplication to calculate the relative importance score for every single node in the graph. It is computationally too expensive for complex graphs. Zhang and Wang [37] introduced a hierarchical method for estimating relative importance of complex networks. Their approach has to perform hierarchical partition that may cause a large overhead; moreover, the paper has not present experimental results.

In this paper, we define a new relative importance measure for nodes in the graphs. Based on the new measure, we present an algorithm aiming to provide a way to estimate relative importance of nodes for large social networks.

3. Problem Formulation

Our research focuses on selecting *Top-k* most relatively important nodes with respect to a set of query nodes in the graphs such as those depicting social networks. We believe in many cases,

Top-k most important nodes can achieve the goals of many applications, especially when the size of the graph or network is very large. For instance, in the case of terrorist network, given some known terrorists (nodes) and their communications (edges), the ability to quickly identify *Top-k* terrorist suspects is essential to prioritize the elimination of security threats, especially if the resources are constrained.

We now describe the problem formulation in steps, each step is closely related with the previous one:

1).Given a graph $G(V, E)$, where V is a set of nodes and E a set of edges, compute the "relative importance" of a node t in G with respect to another (query) node s, also in G. We represent this relative importance as $I(t|s)$. Our proposed solution works for graphs with either weighted or un–weighted edges.

2). Given a graph $G(V, E)$, any query node s from G, and an integer k, find the k nodes in G with the highest "relative importance" with $I(v_i|s)$, $1 \le v_i \le k$.

3). Given a graph $G(V, E)$, a set of query nodes S from G, and a node t in G, compute the "relative importance" of t with respect to the set of query nodes S. This can be denoted as $I(t|S)$, a non-negative quantity. We define:

$$I(t|S) = \frac{1}{|S|} \sum_{s \in S} I(t|s) \tag{1}$$

4). Given a graph $G(V, E)$, a set of query nodes S from G, and an integer k, find k nodes with the highest "relative importance" scores with respect to $I(v_i|s)$, $1 \le v_i \le k$.

These four problems are closely related, and computing the relative importance $I(t|s)$ of one node t to the query node s is fundamental. We treated un–weighted graphs as a special case of weighted graphs with unity edge weight 1, therefore the presented solutions for these four problems apply to both weighted as well un–weighted graphs.

4. Path Probability

In the section, we will present a novel relative measure for relative importance of nodes with respect to a set of query nodes. Intuitively, if a node i has many heavily weighted paths to another node j, then j is very important to i.

Definition 1: *Non-loop path*. A non-loop path p on a graph is a path with no repeated nodes, defined as follows:

$$p = ((v_1, v_2, \ldots, v_m) | \forall i, j : 1 \le i, j \le m, v_i \in V \text{ and } v_i \ne v_j \text{ if } i \ne j) \tag{2}$$

All the paths we mention in this paper is are non-loop paths. When a query node has only one neighbor, using the random walk model would make the neighbour equally important to the query node. In order to make sure a query node is more important than any of its neighbors with respect to itself, a small flying out factor f is introduced into the random walk model and it can be interpreted as the information loss during the propagation.

Generally speaking, the importance of a node a to its neighbor s is defined as the probability of s randomly jumps to node a with a flying out probability:

$$P(s, a) = \begin{cases} (1 - f)\dfrac{e(s, a)}{\sum_{v \in neighbor(s)} e(s, v)} & s \ne a \\ 1 & s = a \end{cases} \tag{3}$$

where f is a value between 0 and 1, usually f is quite small. Also $e(s, a)$ is the corresponding edge weight of a node a to its neighbor s, and $e(s, v)$ is similarly defined.

The selection of f is based on two criteria 1. how well f can preserve the random walk property 2. whether it converge in an acceptable rate. Ideally, we want to keep f as small as possible. However, as f becomes very small, the computational time increase dramatically. In our experiment, we found when $f = 0.1$, the results are quite similar to $f = 0.001$ with an average Kendall Tau Distance> 0.98, and as f increases from 0.1, the Kendall Tau Distance decreases very fast.

In addition, on average the computational time for $f = 0.1$ is only about two times of the computational time for $f = 0.3$. In this paper, we let $f = 0.1$ for the reason that it can help to achieve both criteria. In order to decide whether a path p is important with respect to a node is important(vs. a node is important), a new concept called *path probability* of a path p is defined to measure the contribution of the importance of the ending node t to the query node s through this path p.

Definition 2: *Path Probability*. $PPath(s, v_1, v_2, \ldots, v_m, t)$ is defined as probability of moving from a starting node s to another node t in the graph following a non-loop path $s, v_1, v_2, \ldots, v_m, t$:

$$PPath_{s,v_1,v_2,\ldots,v_m,t}(s, t) = P(s, v_1)\left(\prod_{i=1}^{m-1} P(v_i, v_{i+1})\right) P(v_m, t)(s, t) \tag{4}$$

where $P(v_i, v_{i+1})$ is defined in Eq. (3). It is easy to prove that path probability is a value between 0 and 1 including both 0 and 1. In general, we use $PPath_p(s, t)$ to denote a *path probability* without spelling out the path.

Each path carries some signal of the connection between the starting node s and the ending node t. If path probability of a path is large, this path carries a strong signal of connection from starting node to ending node.

Definition 3: *Significant Path*. A *significant path* from node s to node t is defined as a path p with *path probability* greater than or equal to threshold value c:

$$PPath_p(s, t) \ge c \tag{5}$$

Definition 4: *Relative Importance*. The *relative importance* of a node t with respect to node s is defined as the sum of all important/significant *path probabilities* from s to t:

$$I(t|s) = \sum_g PPath_g(s, t) \tag{6}$$

where $\forall g \subseteq G, start(g) = s \text{ and } end(g) = t,\ PPath_g(s, t) \ge c$. We use the arithmetic mean of the path probabilities as the relative importance of node t to the set of query nodes S (see Eq. (1)). If node s has a lot of significant paths to t, it means strong signal can be carried from s to t through different paths. This indicates that t is important to s.

5. Algorithm to Estimate Relative Importance Based on Path Probability

One of the biggest challenges for graph mining is the scalability issue because a lot of graph problems are NP-hard. Many algorithms for estimating the relative importance of nodes in the graph involve matrix multiplication that is usually computational too expensive especially for large graphs such as social networks.

As mentioned before, the problem targeted in this paper is to find the *Top-k* relatively important nodes to the query node where k is usually relatively small compared to the size of the graph. If the path probabilities of all paths from the query node s to a node t are smaller than the threshold value, then this node t is not important to node s. Therefore, we only consider nodes with at least one path to s such that its path probability with respect to s is greater than c. In this section, an algorithm aggregating the path probabilities of each node with respect to the query nodes called *SigPathSum* is presented.

We now focus on calculating the relative importance of nodes with respect to a query node given a threshold value. The whole process is a depth-first search in a graph. Since each path does not have loop, the longest possible path is no more than N hops where N is size of the graph.

Figure 1 illustrates an example how the path probabilities are aggregated. The algorithm can be described as follows:

The algorithm calculates the relative importance of nodes with respect to one query node. It starts from the query node s and travels to its neighbors in the breadth-first search manner. At each step, it calculates the path probability from the query node s to the current visiting node. If the path probability is smaller than the threshold value, that means the path is not important to the nodes that are not yet visited because the probability of visiting the unvisited nodes through this path from the query node is too small. Since there is the flying out probability and usually the path probability of a path diminishes quickly as a path extends to be suffixed with more nodes.

If there is a set of query nodes, the above algorithm will run for each node in the set. And the average value of the relative importance scores of each node to the set of query nodes is used to represent the relative importance of the node to the query nodes. The k nodes with highest relatively importance scores are selected as the k most relatively important nodes.

Suppose the maximal number of neighbors for any node is k, and the total number of nodes in the graph is n and the threshold value is *threshold*. When the path probability falls below *threshold*, it will stop expanding. According to the definition of *path probability*, when a path expands to its neighbor, its path probability is at most $(1-f)$ of its previous path probability.

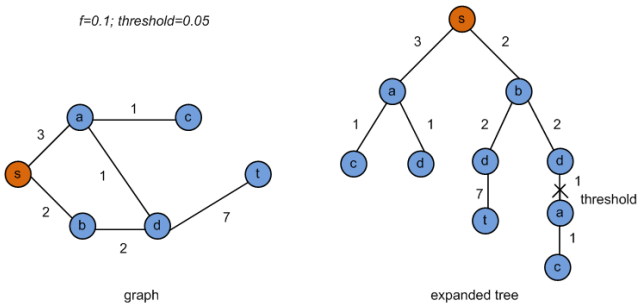


Fig. 1 An example of relative importance calculation process of tree expansion.

Algorithm 1 *SigPathSum*

Algorithm (Compute the relative importance of each node to one query node)
SigPathSum (prefix, suffix, threshold, pathProb)
$prefix$: a stack stores the nodes visited
$suffix$: a list stores the nodes to visit
threshold: a value decides whether a path is important
pathProb: the path probability of the path visited
if $|suffix| == 0$ **then**
  return
end if
$v \leftarrow prefix.peek()$
for $u \in neighbor(v)$ **do**
  if $u \notin suffix$ **then**
    if $pathProb \times P(v,u) < threshold$ **then**
      return
    else
      $newPre \leftarrow prefix \cup \{u\}$
      $newSuffix \leftarrow suffix - \{u\}$
      $newPathProb \leftarrow pathProb \times P(v,u)$
      $u's\ relative\ importance += newPathProb$
      $call\ SigPathSum(newPre,$
      $newSuffix, threshold, newPathProb)$
    end if
  end if
end for

At step t, the *path probability* is $pathProb(t)$, while at step $t+1$, $pathProb(t+1) < pathProb(t)*(1-f)$. Suppose after m steps, the path probability falls below the threshold value, and it stops expanding. Since m is integer, $m = ceiling(log(threshold)/log(1-f))$. Considering the maximal number of neighbors for any node (i.e., k), the total time complexity is (in the worst case, each node has k neighbors, but since there is one incoming edge (except for the root node), each node has at most $k-1$ outgoing edges):

$$\begin{aligned} S_m &= 1 + (k-1) + \ldots + (k-1)^m \\ &= ((k-1)^{(m+1)} - 1)/(k-2) \end{aligned}$$

where, $m = ceiling(log(threshold)/log(1-f))$.

6. Experiments and Evaluation

In this section, we examine the rankings of nodes using proposed algorithm for several graphs and query sets, with the goal of better understanding how the path probability method works on both synthetic data and the real data. In addition, the ranking results are also compared with one of the most popular approaches: random walk with restart (PageRank with Prior) [20], [27].

6.1 Evaluation on Synthetic Data

In this section, we examine the *Top-10* rankings of our algorithm on two toy graphs first proposed by Ref. [27]. The first graph is an undirected graph with 10 nodes. The second graph is a directed graph with 10 nodes. We compare our algorithm

Table 1 Comparison of relative importance ranking for the nodes in Fig. 2 with respect to node J.

Rank	Path Probability	RWR
1	J 1.000	J 0.300
2	C 0.331	C 0.102
3	E 0.331	E 0.102
4	H 0.331	H 0.102
5	A 0.172	A 0.061
6	B 0.172	B 0.061
7	D 0.172	D 0.061
8	F 0.172	F 0.061
9	G 0.172	G 0.061
10	I 0.172	I 0.061

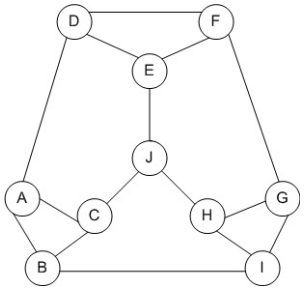


Fig. 2 A undirected toy graph [27].

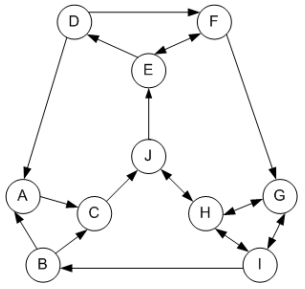


Fig. 3 A directed toy graph [27].

with Random Walk with Restart (RWR) or PageRank with Priors as described in Ref. [20]. We use the same "fly out probability" $f = 0.1$ as Ref. [20].

We first consider a simple undirected graph with 10 nodes, each node with degree three and a total of fifteen edges. Both algorithms give the same rank to the query node J shown in **Table 1**. For both algorithms, J is the most relative important node to J, which makes sense. C, E and H have the same relative importance to J, and are relatively more important than A, B, D, F, G, and I with the same relative importance scores.

Figure 3 shows a directed variation of the previous graph and edges are of equal weight. (**Fig. 2**). The relationships between nodes are more complicated than the undirected graph. For example, consider Fig. 3 as a communication network, and the query node A tries to broadcast message to the rest of the network. It is easy to find C to be the most important node to A, from formula 3, we can get that $P(A, A) = 1$, then we can compute $P(A, C) = 0.9 * 1 = 0.9$, $PPath_{A,C,J}(A, J) = P(C, J) * P(A, C) = 0.81$, $PPath_{A,C,J,E}(A, E) = P(J, E) * PPath_{A,C,J}(A, J) = 0.3645$. Upon examination of Fig. 3, one can reach the same conclusion described above, because without C, it is impossible for A to send message to any other nodes. Similarly, J is the second most important node to A, because without J, A can not send message to other nodes except C. And B might be the least important node to A, because it has the longest shortest path to A among all the nodes in the graph. Our defined measure of relative importance (or *SigPathSum*) can capture such relationships because it takes both the number of paths and the length of the path from the query node into consideration. The ordering of running path probability ranking algorithm is: A, C, J, H, E, F, G, I, D and B.

Table 2 Comparison of relative importance ranking for the nodes in Fig. 3 w.r.t. nodes A and F.

Rank	Path Probability	RWR
1	F 0.541	F 0.201
2	A 0.529	A 0.168
3	C 0.376	C 0.122
4	J 0.280	E 0.107
5	E 0.269	J 0.106
6	G 0.215	G 0.104
7	H 0.173	H 0.086
8	I 0.110	I 0.056
9	D 0.094	D 0.038
10	B 0.026	B 0.013

Consider an example of a graph with more than one query nodes. We use A and F as the query nodes, same as used in Ref. [27]. **Table 2** shows the results or using both Path Probability and Random Walk with Restart (RWR) approaches with the graph in Fig. 3. Both approaches produce almost identical rankings except the order of E and J. Path Probability algorithm ranks J one position ahead of E primarily because J is more important than E to A, while Random Walk with Restart rank E one position ahead of J. Nevertheless, both E and J have similar relative importance scores to query nodes A and F using either algorithm. *Path Probability* method ranks F and A as the *Top-2* most important nodes which is very obvious. It also considers C as the third most important, because without C, A cannot communicate with any node in the network.

6.2 Evaluation with Real-world Data

We use two real-word data sets to illustrate the applicability of our proposed solution for large, complex graphs. The first data set is the 9–11 terrorist network, and the second data set is the DBLP computer scientist co-author network.

6.2.1 Experiments on September 11 Dataset

The 9–11 terrorist network contains 62 nodes and 304 edges shown in **Fig. 4** [41], where nodes represent terrorists or terrorist suspects, and edges represent the real communication between them. The nodes with colors are the 19 terrorists who hijacked the four airplanes in U.S. We first query the *Top-10* terrorists (shortened last name only) to the query node hijacker Marwan Al-Shehhi who plays a key role in communicating with the hijackers in four different airplanes. The Path Probability approach is able to identify 8 hijackers who all have a lot of direct communications with other hijackers in other airplanes, and Ramzi Bin al-Shibh who has a lot of connections with three major hijackers (who have the most connections with other hijackers) and the European Al Qaeda terrorist cells. If the 9 identified nodes are removed, the communication between the query node Marwan Al-Shehhi and other nodes in the terrorist network would have been disrupted. We compare the results with the list of nodes identified by the RWR algorithm shown in **Table 3**. The *Top-10* nodes selected by both approaches are exactly the same except for minor

Fig. 4 9–11 terrorist network [41].

ordering difference. We also tried two other query nodes: Essid Sami Ben Khemais and Djamal Beghal. Essid Sami Ben Khemais is a European Al Qaeda terrorist cell from Italy, while Djamal Beghal is a leader in the Al Qaeda European network. The results are presented in Table 3. The following are some interesting findings from our analysis. Tarek Maaroufi, Kamel Daoudi, and Zacarias Moussaoui who are known to have strong ties with Khemais and Beghal play a major part in the European operations of Al Qaeda. Using the path probability approach, among the *Top-10* relatively important terrorists to Atta, Al-Shehhi and Jarrah who are the master hijackers and the key members of "Hamburg cell", a group of radical terrorists based on Hamburg, we were able to identify eight hijackers and another two people: Bin al-Shibh and Bahaji. In Table 3, Bin al-Shibh is found to be the most relatively important node. Although he was not a hijacker, after further observation, we know that he is another key member of the Hamburg Cell, and the "key facilitator for the September 11 attacks" [19], [38]. Bahaji was an alleged member of the Hamburg cell that provided money to the perpetrators of the September 11 attacks.

We use the normalized K-Min (minimizing Kendall Tau distance) [34] to measure similarities between the *Top-10* list produced by Path Probability algorithm and RWR. Normalized K-Min distance is a value between 0 and 1. If two list s1 and s2 are identical, K-Min(s1, s2) = 1; if s1 and s2 are in reverse order, K-Min(s1, s2) = 0. From Table 3, One can easily observe the similarity between the lists produced by different approaches. We randomly select twelve sets of nodes with size ranging from 1 to 4, and calculated the K-Min for path probability and the RWR. The mean value of K-Min is 0.956, with the minimal K-Min of 0.889 and maximal K-Min of 1. The results show that Path Probability and RWR output very similar results for those twelve sets of query nodes. The reason could be that both approaches are random walk based. Since the size of the network is small (only 62

Table 3 Importance ranking for the nodes in Fig. 4 with respect to query nodes.

	Al-Shehhi		Khemais and Beghal		Atta, Al-Shehhi and Jarrah	
Rank	Path Probability	RWR	Path Probability	RWR	Path Probability	RWR
1	Al-Shehhi 1.000	Al-Shehhi 0.351	Khemais 0.509	Khemais 0.503	Atta 0.497	Atta 0.165
2	Atta 0.229	Atta 0.062	Beghal 0.507	Beghal 0.502	Al-Shehhi 0.477	Al-Shehhi 0.155
3	Jarrah 0.134	Jarrah 0.036	Moussaoui 0.129	Moussaoui 0.081	Jarrah 0.417	Jarrah 0.135
4	Bin al-Shibh 0.127	Bin al-Shibh 0.035	Maaroufi 0.116	Maaroufi 0.074	Bin al-Shibh 0.144	Bin al-Shibh 0.039
5	Al-Omari* 0.124	Hanjour 0.0329	Qatada 0.109	Qatada 0.071	Bahaji 0.123	Hanjour 0.036
6	Ahmed 0.121	Al-Omari* 0.0328	Daoudi 0.105	Daoudi 0.0706	Hanjour 0.123	Bahaji 0.033
7	Bahaji 0.116	Ahmed 0.0327	Courtaillier 0.093	Courtaillier 0.062	Essabar 0.102	Essabar 0.027
8	Hanjour 0.114	Bahaji 0.031	Bensakhria 0.091	Walid 0.0622	Budiman 0.093	Budiman 0.026
9	Alshehri 0.108	Suqami 0.030	Walid 0.091	Bensakhria 0.060	Aziz Al-Omari 0.092	Aziz Al-Omari*0.092
10	Suqami 0.106	Alshehri 0.029	Khammoun 0.072	Khammoun 0.049	Raissi 0.088	Raissi 0.024
K-Min	0.933		1.0		0.978	

Table 4 Importance ranking for the nodes in DBLP SIGMOD Co-author network with respect to Jiawei Han.

Rank	RWR	SigPathSum			
		$c = 0.001$	$c = 0.0001$	$c = 0.00001$	$c = 0.000001$
1	Han 0.358	Han 1.000	Han 1.000	Han 1.000	Han 1.000
2	Pei 0.025	Pei 0.081	Pei 0.087	Pei 0.087	Pei 0.087
3	Fu 0.019	Fu 0.068	Fu 0.072	Fu 0.072	Fu 0.072
4	Chiang 0.017	Zaiane 0.059	Zaiane 0.063	Zaiane 0.064	Zaiane 0.064
5	Zaiane 0.017	Chiang 0.059	Chiang 0.063	Chiang 0.064	Chiang 0.064
6	Wang 0.016	Wang 0.055	Wang 0.060	Wang 0.061	Wang 0.061
7	Koperski 0.015	Koperski 0.054	Koperski 0.057	Koperski 0.058	Koperski 0.058
8	Lakshmanan 0.014	Lakshmanan 0.044	Lakshmanan 0.047	Lakshmanan 0.049	Khemais 0.509
9	Chang 0.013	Chang 0.043	Jamil 0.046	Jamil 0.047	Jamil 0.047
10	Jamil 0.013	Jamil 0.043	Lu 0.046	Lu 0.047	Lu 0.047
Time (1 ms)	5219	32	141	2484	26641
K-Min (vs. RWR)	1.0	0.978	0.945	0.945	0.945

Table 5 Average correlations of *Top-k* ($k = 10, 20$) for 20 query sets.

		RWR	SigPathSum			
			$M = 100$	$c = 0.0001$	$c = 0.00001$	$c = 0.000001$
RWR		1.000	0.952	0.935	0.941	0.930
SigPathSum	M=100	0.952	1.000	0.962	0.940	0.937
	$c = 0.0001$	0.935	0.962	1.000	0.962	0.955
	$c = 0.00001$	0.941	0.940	0.962	1.000	0.971
	$c = 0.000001$	0.930	0.937	0.955	0.971	1.000

Table 6 Comparison of average computational time between Path Probability and the RWR for 20 query sets.

	SigPathSum			
settings	$M = 100$	$c = 0.0001$	$c = 0.00001$	$c = 0.000001$
$\frac{PathProbability}{RWR}$	0.012	0.096	1.430	13.050

nodes), the difference between computational time can be ignored for both approaches.

6.2.2 Experiments on DBLP Dataset

In this section, we use the DBLP data set to demonstrate that the two–phase framework with path probability can be used to quickly identify the *Top-k* most important nodes to the query nodes. The DBLP data set presents information on computer science publications listed in the DBLP Computer Science Bibliography [42]. The data in this dataset was derived from a snapshot of the bibliography which contains a sample dataset of the authorship graph from the ACM SIGMOD conference [43]. The sample data set contains 3,379 computer scientists (nodes) and 8,430 co-authorship (edges).

Finding the *Top-k* most relatively important nodes to the query nodes (researchers) in the ACM SIGMOD DBLP co-authorship network can be helpful to the query of researchers to find the best suitable collaborative partners to co-author papers to the SIGMOD conference or write research proposals in this area.

We first use Jiawei Han (a professor in data mining area at University of Illinois at Urbana-Champaign) as a query node. We run both RWR approach and Path Probability with different threshold values. The *Top-10* most important nodes to Jiawei Han in the co-authorship network is shown in **Table 4**. The results produced by both algorithms are very similar with the *K-Min* distance greater than 0.94. As the threshold decreases, the *Top-k* list outputted by the path probability approach does not change much while the computational time increases sharply from 32 milliseconds to 26,641 milliseconds.

We randomly select 20 query sets: 10 sets with only one node, 5 sets with two nodes, 5 sets with three nodes. We ran both RWR algorithm and our proposed algorithm with different fixed threshold values ($c = 0.0001, 0.00001, 0.000001$). We also pick the *Top* $M = 100$ nodes with largest path probabilities to the query nodes and use that to decide a threshold value. For each query set, the

Top-10 list and *Top-20* list are compared between different algorithms. We compare the computational time of our approach with different threshold values with RWR algorithm. The running time for RWR is quite consistent, while the ratio of running time increased for the *Path Probability* method is faster than the ratio of the threshold value decreases. The results shown in **Table 5** and **Table 6** indicate that our approach can be a good approximation of the RWR with much less time (with average K-min above 0.93 while consuming only 1% of time). In addition, using *SigPathSum* algorithm, when threshold is smaller than certain value (e.g., $c \leq 0.0001$), the results appears to change not much.

6.3 Discussion

The experiments show that in real-life application *SigPathSum* can help identify the terrorist suspects given some known terrorists, and recommend possible collaboration to researchers. How to choose the threshold value does play a very important role in this approach. For graphs with node degree less than 100, $c = 0.0001$ is an effective threshold value based on the empirical results. For graph with node degree greater than 1,000, empirical results show using (*Top-100*) nodes with largest path probabilities to the query nodes to decide the threshold value is a good choice. For large graphs, RWR is very expensive computationally, both in term of computational time and storage space required. *SigPathSum* can be a good choice for approximation for RWR.

7. Conclusion

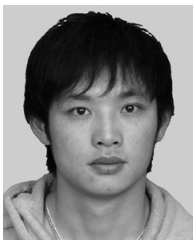
In this paper, we proposed a solution to find the *Top-k* most "relatively important" nodes with respect to some query nodes in social networks. The measure is based on the aggregated path probabilities of the significant paths between two nodes in the network. We define the path probability to measure the significance of a path between two nodes. The experiments show that the proposed approach can effectively identify relatively important nodes with respect to the query node in the terrorist network and the DBLP co-authorship network. It also shows that outcome of our approach has a very strong correlation with random walk with restart (or Personalized PageRank) algorithm. Using certain threshold value ($c = 0.0001$) can generate comparable results of RWR in one order less time, therefore it can be applied to analyze large-scale dataset.

Acknowledgments The work presented in this paper was partially supported by National Program on Key Basic Research Project (973 Program No.2011CB302600), National Natural Science Foundation of China (Program No.60803162), Natural Science Basic Research Plan in Shaanxi Province of China (Program No.12JQ8029), Scientific Research Program Funded by Shaanxi Provincial Education Department (Program No.11JK1061) and China Scholarship.

References

[1] Granovetter, M.S.: The Strength of Weak Ties, *American Journal of Sociology*, Vol.78, No.6, pp.1360–1380 (1973).
[2] White, C., Plotnik, L. and Kushma, J., et al.: An Online Social Network for Emergency Management, *International Journal of Emergency Management.*, Vol.6, No.3/4, pp.369–382 (2009).
[3] School Principals and Social Networking in Education: Practices, Policies, Report. edWeb.net, IESD, Inc., MCH, Inc., MMS Education (2010).
[4] Jackson, M.O.: An Overview of Social Networks and Economic Applications, *The Handbook of Social Economics*, North Holland Press (2011).
[5] Eve, R., Horsfall, S. and Lee, M.: *Chaos, Complexity, and Sociology: Myths, Models, and Theories*, London: Sage Publications (1997).
[6] Hanneman, R.A. and Riddle, M.: *Introduction to social network methods*, Riverside, CA: University of California, Riverside (2005), available from ⟨http://faculty.ucr.edu/˜hanneman/⟩.
[7] Zhu, Y., Qin, L., Yu, J.X. and Cheng, H.: Finding top-k similar graphs in graph databases, *Proc. 15th International Conference on Extending Database Technology, EDBT'12*, pp.456–467 (2012).
[8] Zou, L., Chen, L. and Lu, Y.: Top-K Subgraph Matching Query in A Large Graph, *PIKM 07*, pp.139–146 (2007).
[9] Wasserman, S. and Faust, K.: *Social Network Analysis*, Cambridge University Press (1994).
[10] Freeman, L.: A set of measures of centrality based upon betweenness, *Sociometry*, No.40, pp.35–41 (1977).
[11] Freeman, L.C.: Centrality in social networks: Conceptual clarification, *Social Networks*, Vol.1, No.3, pp.215–239 (1979).
[12] Hage, P. and Harary, F.: *Structural Models in Anthropology*, Cambridge University Press, Cambridge (1983).
[13] Rudnick, J. and Gaspari, G.: *Elements of the Random Walk: An Introduction for Advanced Students and Researchers*, Cambridge University Press (2004).
[14] Brin, S. and Page, L.: The anatomy of a large-scale hypertextual web search engine, *Proc. 7th International World Wide Web Conference*, Brisbane, Australia, pp.107–117 (1998).
[15] Heckathorn, D.D., Broadhead, R.S., Anthony, D.L. and Weakliem, D.L.: AIDSand social networks: Prevention through network mobilization, *Sociological Focus*, No.32, pp.159–179 (1999).
[16] Meltzer, D., Chung, J., Khalili, P., Marlow, E., Arora, V. and Schumock, G., et al.: Exploring the Use of Social Network Methods in Designing Healthcare Quality Improvement Teams, *Social Science & Medicine*, Vol.71, No.6, pp.1119–1130 (2010).
[17] Hu, J., Wang, B. and Lee, D.: Evaluating Node Importance with Multi-Criteria, *IEEE/ACM International Conference on Green Computing and Communications & International Conference on Cyber, Physical and Social Computing*, pp.792–797, IEEE Computer Society, Washington, DC, USA (2010).
[18] Faloutsos, C., McCurley, K.S. and Tomkins, A.: Fast Discovery of Connection Subgraphs, *KDD 2004*, pp.118–127, Seattle, WA (2004).
[19] Page, L.: PageRank: Bringing Order to the Web, Stanford Digital Library Project, talk, August 18, 1997 (archived 2002).
[20] Tong, H., Faloutsos, C. and Pan, J.Y.: Fast Random Walk with Restart and its Applications, *ICDM*, pp.613–622 (2006).
[21] Newman, M.E.J.: A Measure of Betweenness Centrality Based on Random Walks, *Soc. Netw*, No.27, pp.39–54 (2005).
[22] Page, B. and Motwani, W.: The PageRank Citation Ranking: Bringing Order to the Web, Stanford University, Computer Science Department Technical Report, available from ⟨http://ilpubs.stanford.edu:8090/422/⟩.
[23] Haveliwala, T.: Topic-Sensitive PageRank: A Context-Sensitive Ranking Algorithm for Web Search, *IEEE Trans. Knowledge and Data Engineering*, Vol.15, No.4, pp.784–796 (2003).
[24] Kleinberg, J.: *Hubs, Authorities, and Communities*, Cornell University (1999) (Retrieved 2008).
[25] Kleinberg, J.: Authoritative Sources in A Hyperlinked Environment, *J. ACM*, Vol.46, No.5, pp.604–632 (1999).
[26] Freeman, L.C., Borgatti, S.P. and White, D.R.: Centrality in Valued Graphs: A Measure of Betweenness Based on Network Flow, *Social Network*, Vol.13, No.2, pp.141–154 (1991).
[27] White, S. and Smyth, P.: Algorithms for Estimating Relative Importance in Networks, *Proc. 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.266–275, Washington, D.C. (2003).
[28] Katz, L.A.: New Status Index Derived from Sociometric Index, *Psychometrika*, pp.39–43 (1953).
[29] Stephenson, K.A. and Zelen, M.: Rethinking centrality: Methods and examples, *Social Networks*, Vol.11, No.1, pp.1–37 (1989).
[30] Lempel, R. and Moran, S.: The stochastic approach for link-structure analysis (SALSA) and the TKC effect, *Computer Networks: The International Journal of Computer and Telecommunications Networking*, Vol.33, No.16, pp.387–401 (2000).
[31] Borodin, A., Roberts, G.O., Rosenthal, J.S. and Tsaparas, P.: Finding authorities and hubs from link structures on the world wide web, *The 11th International World Wide Web Conference*, pp.415–429 (2001).
[32] Jeh, G. and Widom, J.: Scaling Personalized Web Search, *Proc. 12th International World Wide Web Conference*, NY, pp.271–279 (2003).
[33] Chang, H., Cohn, D. and McCallum, A.: Creating Customized Au-

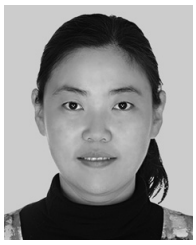
thority Lists, *Proc. 17th International Conference on Machine Learning*, Stanford, CA. (2000).
[34] Fagin, R., Kumar, R. and Sivakumar, D.: Comparing top k lists, *SIAM J. Discrete Math.*, Vol.17, No.1, pp.134–160 (2003).
[35] Fouss, F., Pirotte, A., Renders, J. and Maerens, M.: Random-walk computation of similarities between nodes of a graph, with application to collaborative recommendation, *IEEE Trans. Knowledge and Data Engineering*, Vol.19, No.3, pp.415–429 (2006).
[36] Kerrebroeck, V. and Marinari, E.: Ranking vertices or edges of a network by loops: A new approach, *Physical Review Letters*, Vol.101, No.9, 098701 (2008).
[37] Zhang, W. and Wang, Q.: A Hierarchical Method for Estimating Relative Importance in Complex Networks, *International Symposium on Computer Science and Computational Technology*, Vol.1, pp.63–65 (2008).
[38] en:wikipedia:org=wiki=Hamburgcell, 10, May (2012).
[39] Thelwall, M.: *Link Analysis: An Information Science Approach*, Academic Press (2004).
[40] Yang, C.C. and Tang, X.: A content and social network approach of bibliometrics analysis across domains, *Proc. 2012 iConference*, pp.515–517 (2012).
[41] Krebs, V.: Unlocking Terrorist Networks, *Fisrt Monday*, Vol.7, No.4 (2002).
[42] available from ⟨http://www.informatik.uni–trier.de/˜ley/db/⟩, (accessed 2011-11-10).
[43] available from ⟨http://www.cs.cmu.edu/˜htong/work/CePS.zip⟩, (accessed 2011-11-10).



Heyong Wang received his M.S. degree in Computer Science from Iowa State University. He is now working for Microsoft while continuing this research towards a Ph.D. degree under the guidance of Professor Carl Chang. He is specialized in graph mining, and social network mining.

Carl K. Chang received his Ph.D. from Northwestern University in 1982. He is currently Professor and Chair of Computer Science at Iowa State University. He was editor-in-chief for IEEE Software (1991–1994) and IEEE Computer (2007–2010). He is a Fellow of IEEE and AAAS. His current research interests include software engineering and services computing.

Hen-I Yang received his Ph.D. from University of Florida in 2008 and became a post-doc research scientist at Iowa State University from 2009 to 2012. His current research interests include pervasive computing and computer networking.



Yanping Chen received her Ph.D. from Xi'an Jiaotong University in 2007, and became an associate professor at Xi'an University of Posts and Telecommunications in 2009. Now, she is a visiting scholar at Iowa State University. Her current research interest is service computing and quality of service.