Instant Social Graph Search*

Sen Wu, Jie Tang, and Bo Gao

Department of Computer Science and Technology,
Tsinghua University, Beijing, 100084. China
ronaldosen@gmail.com, jietang@tsinghua.edu.cn, elivoa@gmail.com

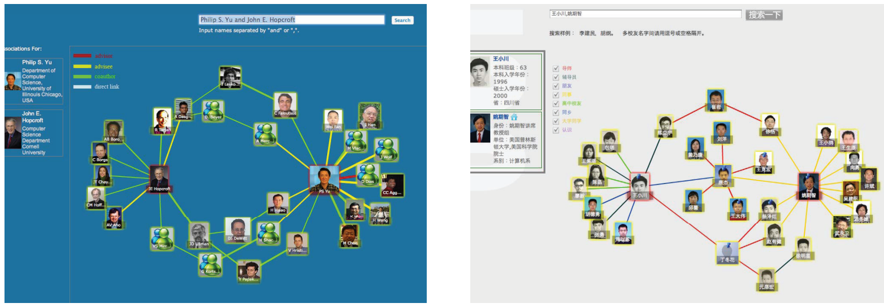
Abstract. In this paper, we study a new problem of instant social graph search, which aims to find a sub graph that closely connects two and more persons in a social network. This is a natural requirement in our real daily life, such as "Who can be my referrals for applying for a job position?". In this paper, we formally define the problem and present a series of approximate algorithms to solve this problem: Path, Influence, and Diversity. To evaluate the social graph search results, we have developed two prototype systems, which are online available and have attracted thousands of users. In terms of both user's viewing time and the number of user clicks, we demonstrate that the three algorithms can significantly outperform (+34.56%-+131.37%) the baseline algorithm.

1 Introduction

With the big success of many large-scale online social networks(e.g., Facebook, Ren-Ren, MySpace, Ning, and Twitter) and the rapid growth of the mobile social networks (e.g., FourSquare, Data.net, Strands), there has been a large increase in the people's social friends especially online social network friends. The online social network is becoming one of the most important ties between people's daily life and virtual web space. For example, Facebook, which is the most-visited site on the web, contains more than 600,000,000 unique visitors(users) since Jan 2011; Foursquare, a location-based mobile social network, has attracted 6 million registered users by the end of 2010. There is little doubt that most of our friends are online now.

In such a case, one important requirement in the social network is to find the connections (also called associations) among persons [14], which has many direct applications. For example, to find referral people for applying for a job position [9]. Indeed, LinkedIn has a very important function, which allows users to see how far (how many degrees) you are from another user and allow users to write recommendation to a friend. In particular, interesting questions arise: "Who are the good referrals for me to apply for the PhD program of a university?", "What are my relationships to the Turing Award winner, Prof. John Hopcroft?", and "Who are the experts on topic X and how to connect him/her?". For all the questions, the answers should be returned in real time. The general problem is referred to as instant social graph search. Please note that the connection between people might be directed, e.g., via a coauthorship; or indirected, e.g., the friend's friend.

* The work is supported by the Natural Science Foundation of China (No. 61073073) and Chinese National Key Foundation Research (No. 60933013, No. 61035004), a special fund for Fast Sharing of Science Paper in Net Era by CSTD.



(a) Coauthor network (b) Alumni network of a university

Fig. 1: Two examples of instant social graph search in a coauthor network and a university alumni network. The left figure shows the social graph between two computer science experts: "Philip Yu" and "John Hopcroft" in the coauthor network. The right figure shows the social graph between "Andrew Yao" (Turing Award winner) and "Xiaochuan Wang" (Vice President of a company) in the alumni network.

Motivating Example. To clearly motivate this problem, Figure 1 gives examples of instant social graph search on a coauthor network and an alumni network of a university. The figure 1(a) shows the social graph between two experts in computer science: "Philip Yu" and "John Hopcroft" and the figure 1(b) plots the social graph between one faculty "Andrew Yao" (Turing Award winner) and one alumnus "Xiaochuan Wang" (Vice President of a company) discovered from the alumni network. In the figure 1(b) different colored links indicate different types of relationships. For example, in the left figure, yellow-colored link indicates advisee relationship, red-colored link indicates advisor relationship, and green-colored link indicates coauthor relationship. While in the right figure, the types of relationships include: advisor, colleague, classmate, high-school alumni, friendship, etc. "Pictures Worth a Thousand Words". We can see such a social graph is very helpful to understand the social connection among persons. With such a graph, we can easily find trusted referrals for connecting a person (e.g., an expert), who are very likely to give a help because they are friends of your friends.

The problem is non-trivial. One fundamental challenge is how to effectively select and generate the social graph between (or among) persons in real time. It is well-known that any two persons in the world are connected in six steps or fewer [13]. This means that almost any persons in the world are within your six-degree social circle. At the same time, this also implies that for any two persons, the number of connections between them would be huge. Obviously it is infeasible to display all the connections between persons in a social graph. Our preliminary study shows that when a graph consists of more than 50 nodes, the user will have difficulties in understanding the meaning of the graph, and quickly lose interest to the graph (with less viewing time).

Challenges and Contributions. In this work, we try to conduct a systematic investigation of the problem of *instant social network search*. The problem poses a set of unique challenges:

- *Goodness.* How to quantify the goodness of a sub network among people? Specifically, given a graph G and a query consisting of multiple person nodes in the graph, how to find a "good" subgraph of G that contains the query nodes.
- *Diversity.* How to diversify the returned graph so that it captures the whole spectrum of the connections among the queried persons? It is widely realized that diversity is a key factor to address the uncertainty in an information need [1, 21].
- *Efficiency.* How to return the queried graphs instantly? As real social networks are getting larger with millions or billions of nodes, it is necessary to design an efficient algorithm which can return the queried social graphs in (milli-)seconds.

To address the above challenges, we first precisely define the problem and then propose an efficient algorithm to solve the problem. We further incorporate the topic diversity into the objective function and propose an enhanced diversity algorithm. We have developed two prototype systems, one is for a coauthor network and the other is for a university alumni network, both of which are online available and has attracted thousands of users. We evaluate the performance of the proposed algorithms in terms of user viewing time and number of user clicks. Experimental results on one-month query log show that the proposed algorithms can significantly outperform (+34.56%-+131.37% in terms of viewing time) the alternative baseline algorithm. We also find that the Diversity algorithm achieves the best performance. Our experiments also validate the efficiency of the presented algorithms, which can return the search results for most queries in 2 seconds.

2 Problem Definition

In this section, we first give several necessary definitions and then present the problem formulation.

A social network is modeled as an undirected graph $G = (V, E, U, W)$, where V represents a set of users, $E \subset V \times V$ represents a set of social relationships between users, $u_i \in U$ represents the importance (or activity) of user v_i, and $w_{ij} \in W$ represents the closenesses between user v_i and user v_j. Given a query of k persons $q = \{v_{q1}, \cdots, v_{qk}\}$, the goal is to find a set of users $S_q \subset V$ to closely connect the queried users in q, by considering the *importance* of nodes, the *closeness* of relationships, and the *connectedness* to the query users. In different networks, the three criteria can be instantiated in different ways. For example, in a coauthor network, importance can be defined as the number of papers published by the author (or the total number of citations of the author, or simply the value of H-index [7]), while the relationship's closeness can be defined as the number of coauthored papers. Formally, we can define the social graph search problem as follows:

Definition 1. ***Social Graph Search**: Given a social network $G = (V, E, U, W)$ and a query $q = \{v_{q1}, \cdots, v_{qk}\}$ of k persons, the goal of social graph search is to find a subgraph G_q of G, such that (1) G_q contains the queried persons, i.e., $\{v_{q1}, \cdots, v_{qk}\} \subseteq V_q$, (2) nodes in the subgraph G_q are closely connected, and (3) the number of nodes in the returned graph is less than a threshold, i.e., $|V_q| \leq M$.*

In the definition, we explicitly constrain the number of persons in the returned social graph as M (condition (3)). This constraint is necessary for controlling the size of the returned subgraph; otherwise, algorithm would trivially return the whole social graph. Now the problem is how to satisfy the second constraint: nodes in the subgraph G_q are closely connected, more specifically, how to quantify the connectness of a graph. To make things simple, we define the connectness as the number of relationships among the selected nodes in the graph G_q. Another challenge is how to diversify the selected nodes in the graph. In Section 3 we will introduce how we achieve these two goals and find the trade-off balance between them.

Several relevant research efforts [2] has been made so far. However, our problem addressed in this paper is very different from existing work. For example, [2] proposes the notion of semantic association and has investigated how to rank the semantic associations based on the information gain. However, association search is different from social graph search. The former is to find association paths to connect two persons, while our goal is to find a social graph to connect multiple persons. Our problem can be viewed as a generalized problem of the association search. Faloutsos et al. [5] also study how to efficiently discover a connection subgraph between nodes in a graph. However, they do not consider the importance of nodes and weight of relationships together, and they do not give an objective method to evaluate the discovered subgraph. Our work aims at satisfying both of the two goals: relevance and diversity. Sozio and Gionis [15] study a community-search problem, which has an objective similar to our work. However, the algorithm cannot be scaled up to handle networks of millions of nodes in real time.

3 Algorithms

The problem of social graph search as we defined in Section 2 is NP-hard, which can be proved by a reduction to the Dominating Set Problem. In this section, we will introduce three algorithms to obtain approximate solutions of the problem, respectively called Path, Influence, and Diversity. For easy explanation, we consider only two persons in the query, i.e., $q = \{v_{q1}, v_{q2}\}$.

3.1 Basic Ideas

There are two basic objectives we want to achieve in the social graph search problem. The first is to find important nodes and the second is to find nodes that could closely connect the queried nodes. In general, the connective social graph between user v_{q1} and v_{q2} can be decomposed into multiple paths between them [8]. Therefore our first idea is to cast the problem as shortest associations finding. According to the weighted importance w_{ij} between users, we can find the shortest association path between any two users using dynamic programming, and then find the top-k shortest paths by relaxing the search condition. This algorithm is called *Path*. It is efficient and easy to implement. However, the algorithm does not consider the importance of nodes and also the possible redundant information (i.e., the same nodes and edges) between different paths.

We therefore propose an influence maximization based algorithm, called *Influence*. The idea is to cast the problem as that of influence maximization [10], whose goal is to

```
Input: G, number of selected pathes k, bound to shortest path δ;
Output: S;
Initialize S = ∅;
Initialize D = inf;
Use Dijkastra algorithm to calculate the shortest path D;
for i = 1 to D + δ do
    create a queue Q;
    enqueue source on Q;
    mark source;
    while Q is not empty do
        dequeue an item from Q into V;
        foreach edge e incident on v in Graph do
            let w be the other end of e;
            if w is not marked: then
                mark w;
                enqueue w onto Q;
            end
        end
    end
end
Set all the marked node on the path in S;
Output S;
```

Algorithm 1: Path algorithm.

find a small set of nodes in a social network that maximize the spread of influence under certain models. To further consider the diversity, we propose an enhanced algorithm called *Diversity*. The basic assumption is that each user may focus on different aspects (topics). Without considering the diversity, the resultant graph may be dominated by a major topic (e.g., a resultant graph from the alumni network may be dominated by one's classmates). The new algorithm incorporates the topic information into an objective function, thus the selection strategy achieves a trade-off between the influence of the selected nodes and the diversity of all topics over the resultant graph.

3.2 The *Path* Algorithm

A straightforward method to deal with the instant social graph search problem is to find the shortest paths between two persons and then use those persons appearing in the paths to construct the social graph. We called this baseline algorithm as Path. More specifically, we take the negative weight $-w_{ij}$ of each edge $e_{ij} \in E$ in the network G as its distance. By using a (heap-based) Dijkastra algorithm [4], we can obtain the shortest path from all nodes to a target node in the network, with a complexity of $O(n\log(n))$. Then we use a depth-first (or width-first) search to find near-shortest pathes by bounding the length (distance) of the path within a factor (i.e., $\leq (1 + \delta)$) of the shortest path. The algorithm is summarized in Algorithm 1.

Limitations. The *Path* algorithm does not consider the correlation (dependency) between two paths, thus it is very likely to choose two "redundant" paths (i.e., paths sharing a number of common nodes). Actually, in our data sets, analysis shows that in many cases, the top 10 shortest paths only have one or two node(s) difference. Another limitation of the algorithm is that it does not consider the importance of each node.

3.3 The *Influence* Algorithm

Our second idea is to cast the social graph search problem as that of influence maximization [10], whose goal is to find a small set of nodes in a social network that maximize the spread of influence under certain models.

In order to achieve this, we first translate the social network into an influence graph where each node indicates a path between the queried nodes. If two paths have a common node, we create an edge between the corresponding nodes in the influence graph and the weight of the edge is the number of common nodes of the two paths. It is easy to know that the new influence graph is a connected graph and then we employ a greedy algorithm [3] to select the nodes in the new graph (i.e., paths in the original graph). The algorithm is based on the Monte Carlo random process. It runs iteratively and in each round, the algorithm selects one vertex into the selected set S such that this vertex together with the current set S maximizes an influence score. Equivalently, this means that the vertex selected in round i is the one that maximizes the incremental score of influence. To do so, for each vertex v that is not in S, the influence spread of $S \cup v$ is estimated with R repeated simulations of random process. The algorithm is presented in Algorithm 2.

Limitations. The *Influence* algorithm considers the network information, and it can avoid redundant nodes (nodes are close with each other in the transferred graph), by adopting a degree discount method [3]. However, it does not consider the diversity problem. In some extreme cases, one major aspect (topic) may dominate the resultant graph. This leads us to propose the *Diversity* algorithm.

3.4 The *Diversity* Algorithm

On a social network, each user may have interest (or expertise) on multiple different topics. When the user searches for social graphs between two persons, he is not only interested in the network that closely connects the two persons, but also interested in how the two persons are connected on different aspects. For example, when the user searches for the social graph between two professors respectively from data mining and theory. The user might be interested in knowing how the two professors build collaborations in different fields.

Hence, we augment the social network model with topic representation, i.e., $G = (V, E, U, W, R)$, where $\mathbf{r}_i \in R$ is a vector denoting the topic distribution of each user v_i with each element r_{ij} representing the probability of user v_i's interest (or expertise) on topic j. Please note that the diversity problem can be also defined in some other ways. For example, we can consider different social ties and thus expect the returned social graph contain diverse social ties. According to the definition, the social graph search problem with diversity can be re-defined as to find a small subset of users to *statistically* represent the topic distribution of the social graph between the queried persons.

```
Input: G, number of selected pathes k;
Output: S;

Initialize S = ∅;
Initialize R = 20000;
for i = 1 to k do
    foreach vertex v ∈ V\S do
        s_v = 0;
        for j = 1 to R do
            s_v += |RanCas(S ∪ {v})|;
        end
        s_v = s_v/R;
    end
    S = S ∪ {argmax_{v∈V\S}{s_v}};
end
Output S;
```

Algorithm 2: Influence algorithm.

The proposed *Diversity* algorithm is based on two principles that are used to select representative users in our physical social network: *synecdoche* (in which a specific instance stands for the general case) and *metonymy* (in which a specific concept stands for another related or broader concept) [12]. Thus one problem is how to define the topic-based representative degree between users. Without loss of generality, we define the representative degree of user v_i on v_j for topic z according to the similarity between two persons on the topic, i.e.,

$$rep(v_i, v_j, z) = \frac{|r_{iz} - r_{jz}|}{r_{iz}} \tag{1}$$

Therefore, our objective is to select a set S of persons who can best represent all the other persons in the social graph on all topics, formally we can define the following objective function:

$$\mathcal{O}(S) = \max_{v_i \in S} \sum_{z} \sum_{v_j \in V \setminus S} rep(v_i, v_j, z) \tag{2}$$

Maximizing the representative degree on all topics is obviously NP-hard. Some trade-offs should be considered as we may need to choose some less representative nodes on some topics to increase the total representative degree on all topics. We give a greedy heuristic algorithm. Each time we traverse all candidate persons in the social graph and find the individual that most increases the representative function $\mathcal{O}(S)$. To increase in representative function achieved by adding a person $v_i \in V$, we only need to consider the topics that v_i can mainly contribute to ($r_{ik} > 0$) and all v_i's neighbors (we say v_j is v_i's neighbor if $rep(v_i, v_j, z) > 0$ for some $v_j \in V \setminus S$). The algorithm is summarized in Algorithm 3:

4 Experimental Results

For evaluation, we have deploy the presented algorithms in two systems: a social graph search in Arnetminer[1] [19] and an alumni network system.

[1] http://arnetminer.org

Input: G, number of selected pathes k;
Output: selected users S;
$S = \emptyset$;
while $|S| < k$ **do**
  $max = -1$;
  foreach $v_i \notin S$ **do**
    foreach $r_{iz} > 0$ **do**
      foreach $v_j \in G$ *that* $rep(v_i, v_j, z) > 0$ **do**
        Compute the increment of $\mathcal{O}(S \cup v_i) - \mathcal{O}(S)$ on topic z;
      end
      Compute the total increment;
    end
    if $increment > max$ **then**
      $v = v_i$; Update max;
    end
  end
  $S = S \cup \{v\}$;
  Update $\mathcal{O}(S)$.
end
Return S;

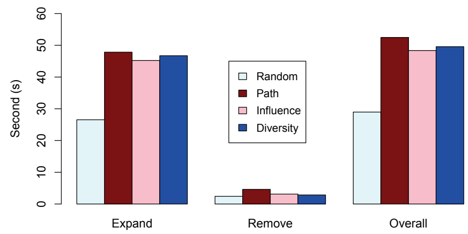
Algorithm 3: Diversity algorithm.

4.1 Experiment Setup

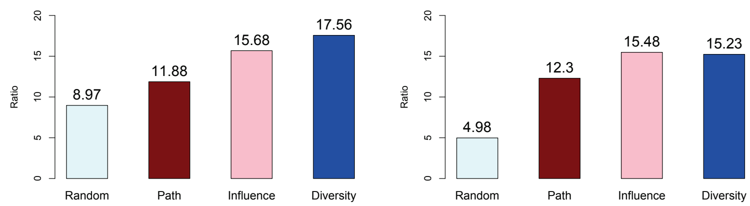
Data Sets. We perform our experiments on the two systems which contain two different data sets: coauthor network and alumni social network.

– Coauthor network. In the coauthor network, we focus on studying the coauthor social graph, which consists of 1,483,246 authors and 47,443,857 coauthor relationships. We also employ a time-dependent factor graph model [22, 20] to discover the advisor-advisee relationships from the coauthor network. The social graph search function has been integrated into academic analysis and mining system for a few months, and attracted tens of thousands of accesses.
– Alumni social network. In the alumni social network, we investigate the alumni network from a university, which is comprised of 17,381 students graduated from its Computer Science department and all faculty members of University. The network contains 2,113,345 relationships of different types (e.g., colleague, advisor-advisee, classmate, high-school alumni, etc.).

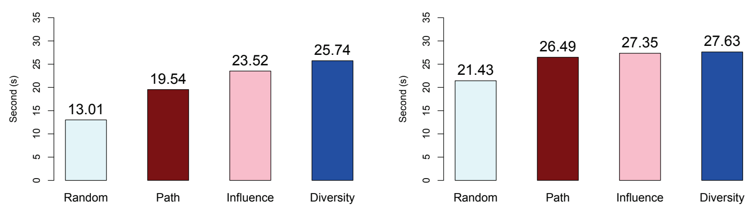
Evaluation Measures. To evaluate the proposed method, we consider two aspects: user's average viewing time and the average number of clicks. User's viewing time stands for how long a user will stay on the returned social graph. Staying for a long time implies that the user may be more interested in the result than that with a shorter time. We also design a user interactive mechanism, which allows the user to expand a person's detailed social information when she/he is interested in knowing more about the person or to remove the node from the returned graph when she/he think the node is irrelevant. For each query, we randomly select one of the proposed three algorithms to generate and return the social graph to the user. We record the user behaviors (viewing time and #clicks) on the returned social graph. We also compare the three algorithms with a baseline algorithm, which randomly selects nodes from the candidate nodes.



(a) #User clicks



(b) Expand/Remove ratio. (Left: Coauthor; Right: Alumni)



(c) Viewing time (Second). (Left: Coauthor; Right: Alumni)

Fig. 2: Performance on the two networks (Coauthor and Alumni).

4.2 Accuracy Performance

As all the comparison methods require the number of users' access and log, we set up the two systems from early 2011. We use the log of four months (March - June, 2011) in the coauthor system (consisting of 57,494 queries) and the log of one month (April, 2011) in the alumni system (consisting of 4,305 queries) to study the performance of different algorithms.Figure 2 shows the results on the coauthor network data and alumni network data.

Effect of user clicking. Figure 2(a) shows the probability of a user clicking a node in the social graph. Expand indicates that the user clicks to see more detailed person's social circle, while Remove indicates that the user clicks to remove a person from the social graph. We see that all the presented four algorithms attract much higher click ratio than the Random algorithm. An interesting phenomenon is that overall the Path algorithm

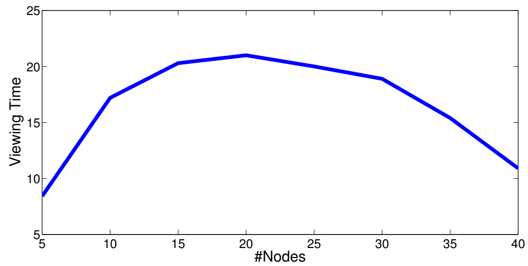


Fig. 3: Viewing time of the number of displayed nodes.

attracts the largest number of user clicks; however, there are also a large number of users click to remove person nodes from the social graph, which implies that there are not only many "interesting" nodes in social graph returned by the Path algorithm, but also many "irrelevant" nodes. To quantify this, we define another measurement called Expand/Remove ratio as ratio of the number of "Expand" clicks divided by the number of "Remove" clicks. Figure 2(b) shows the result of Expand/Remove ratio by the comparison algorithms. It can be seen that the Diversity algorithm has the largest ratio, while the Random and the Path algorithm have lower ratios.

Effect of user viewing time. Figure 2(c) shows the average viewing time of a user on the returned social graph by applying the different algorithms. It can be seen again that the Diversity algorithm results in the longest viewing time, which confirms the findings from Figure 2(b). On average, the presented three algorithms can gain an 73.69%-84.13% increase in terms of the number of (Expand) clicks, and an increase from 34.56%-131.37% in terms of viewing time compared with the baseline (Random) algorithm. In particular, the Diversity algorithm achieves the best performance from the perspective of both Expand/Remove ratio and viewing time.

4.3 Analysis and Discussions.

To obtain deeper understanding of the results, we perform the following analysis.

Effect of the number of displayed nodes. We conduct an experiment to see the effect of the number of the displayed nodes. We use the users' average time of display different nodes to overall performance. The curves of coauthor and alumni network look almost the same. As an example, Figure 3 shows the users spend time on different nods. This suggests that about twenty nodes are good display property.

Error analysis. We conduct an error analysis on the results of our approach. We observe two major types of source of errors.

- Missing data. Sometimes the data is missing because the database does not contain all the coauthor (alumni) relations. For example, there are thousands of papers every year and many different of alumni relations, the database cannot contain all the relations. Thus, the social graph might not also generate the result every time.
- Name ambiguity. In the coauthor network, there might be several persons with the same name. This lead to mistake relationships between persons.

5 Related work

Social graph is an important problem in social network analysis, Tang et al. [18] study the problem of topic-level social network search, which aims to find who are the most influential users [17] in a network on a specific topic and how the influential users connect with each other. In this section, we review the related work on connectivity subgraphs and diversity.

Connectivity Subgraphs. Social graph search is to find a connectivity subgraph among queried users. Faloutsos et al.[5] also address that problem. The main point of that paper is to develop measures based on electrical-current flows of proximity between nodes of the graph that depend on the global graph structure. And there are many ideas, such as Koren et al. [11] refined the proximity measures using the notion of cycle-free effective conductance. The main difference between our approach and above research is that we define users' influence of each person to others and consider the diversity of the subgraph.

Diversity. Diversity is well-recognized as highly property in many data mining tasks, which is very useful to address uncertainty about the information need given a query. One of the most representative works is on expertise search, such as Agrawal et al. [1] and Gollapudi et al. [6]. There are also some works which have focused on diversity result in recommendation. For example, Ziegler et al. [23]. More recently, Tong et al. propose a new approach for diversity of graph search [21]. The difference of our work from existing lies in that we consider the diversity in the resultant social graphs.

The work is also related to the social relationship mining. For example, Tang et al. [20] propose a learning framework based on partially labeled factor graphs for inferring the types of social relationships in different networks. Tang et al. [16] further study the problem of inferring social ties across heterogeneous networks. However, these methodologies do not consider the network search problem.

6 Conclusions

In this paper, we study a novel problem of instant social graph search, which aims to find a subgraph of representative users to closely connect the queried persons. We formally define this problem and present three algorithms to solve the problem. We have developed two systems to validate the effectiveness and efficiency of the presented algorithms. We have deployed the algorithms in two real systems: an academic mining system and an alumni network system. In terms of both users viewing time and number of clicks, we found that the presented algorithms significantly outperform (+34.56%-+131.37% in terms of viewing time) the baseline method. We also found that the Diversity algorithm can achieve the best performance. The presented algorithms are efficient, and can perform most social graph searches in 2 seconds.

Detecting the personalized social graph represents a new research direction in social network analysis. As further work, it is interesting to study how user's feedback can be used to improve the search performance (e.g., interactive learning).

References

1. R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. In *WSDM'09*, pages 5–14, 2009.
2. B. Aleman-Meza, M. Nagarajan, C. Ramakrishnan, L. Ding, P. Kolari, A. P. Sheth, I. B. Arpinar, A. Joshi, and T. Finin. Semantic analytics on social networks: experiences in addressing the problem of conflict of interest detection. In *WWW'06*, pages 407–416, 2006.
3. W. Chen, Y. Wang, and S. Yang. Efficient influence maximization in social networks. In *KDD'09*, pages 199–207, 2009.
4. E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.
5. C. Faloutsos, K. S. McCurley, and A. Tomkins. Fast discovery of connection subgraphs. In *KDD'04*, pages 118–127, 2004.
6. S. Gollapudi and A. Sharma. An axiomatic approach for result diversification. In *WWW'09*, pages 381–390, New York, USA, 2009. ACM.
7. J. E. Hirsch. An index to quantify an individual's scientific research output. *Proceedings of the National Academy of Sciences*, 102(46):16569–16572, 2005.
8. G. Karypis and V. Kumar. Parallel multilevel k-way partitioning scheme for irregular graphs. *SC Conference*, 0:35, 1996.
9. H. Kautz, B. Selman, and M. Shah. Referral web: Combining social networks and collaborative filtering. *Communications of the ACM*, 40(3):63–65, 1997.
10. D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *KDD'03*, pages 137–146, 2003.
11. Y. Koren, S. C. North, and C. Volinsky. Measuring and extracting proximity graphs in networks. In *ACM Trans. Knowl. Discov. Data*, volume 1, New York, USA, December 2007. ACM.
12. T. K. Landauer. Behavioural research methods in human-computer interaction. *M. Helander, T.K. Landauer, and P. Prabhu, (Eds.), Handbook of Human-Computer Interaction*, 1997.
13. S. Milgram. The small world problem. *Psychology Today*, 2:60–67, 1967.
14. C. Ramakrishnan, W. H. Milnor, M. Perry, and A. P. Sheth. Discovering informative connection subgraphs in multi-relational graphs. In *SIGKDD Explor. Newsl.*, volume 7, pages 56–63, New York, USA, December 2005. ACM.
15. M. Sozio and A. Gionis. The community-search problem and how to plan a successful cocktail party. In *KDD'10*, pages 939–948, 2010.
16. J. Tang, T. Lou, and J. Kleinberg. Inferring social ties across heterogenous networks. In *WSDM'12*, 2012.
17. J. Tang, J. Sun, C. Wang, and Z. Yang. Social influence analysis in large-scale networks. In *KDD'09*, pages 807–816, 2009.
18. J. Tang, S. Wu, B. Gao, and Y. Wan. Topic-level social network search. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '11, pages 769–772, 2011.
19. J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su. Arnetminer: Extraction and mining of academic social networks. In *KDD'08*, pages 990–998, 2008.
20. W. Tang, H. Zhuang, and J. Tang. Learning to infer social relationships in large networks. In *ECML/PKDD'11*, 2011.
21. H. Tong, J. He, Z. Wen, R. Konuru, and C.-Y. Lin. Diversified ranking on large graphs: an optimization viewpoint. In *KDD'11*, pages 1028–1036, 2011.
22. C. Wang, J. Han, Y. Jia, J. Tang, D. Zhang, Y. Yu, and J. Guo. Mining advisor-advisee relationships from research publication networks. In *KDD'10*, pages 203–212, 2010.
23. C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. Improving recommendation lists through topic diversification. In *WWW'05*, pages 22–32, New York, USA, 2005. ACM.