I See You: Person-of-Interest Search in Social Networks

Hsun-Ping Hsieh
Grad. Inst. of Networking & Multimedia, National Taiwan University, Taipei, Taiwan
d98944006@csie.ntu.edu.tw

Cheng-Te Li
Research Center for IT Innovation, Academia Sinica, Taipei, Taiwan
ctli@citi.sinica.edu.tw

Rui Yan
Baidu Inc., Beijing, China
yanrui02@baidu.com

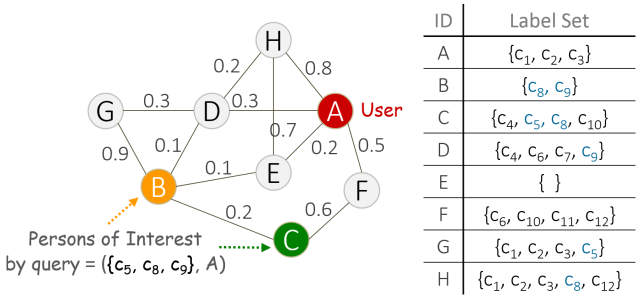
ABSTRACT

Searching for a particular person by specifying her name is one of the essential functions in online social networking services such as Facebook. So many times, however, one would like to find a person but what she knows is few social labels about the target, such as interests, skills, hometown, school, employment, etc. Assume each user is associated a set of social labels, we propose a novel search in online social network, *Person-of-Interest (POI) Search*, which aims to find a list of desired targets based on a set of user-specified query labels that depict the targets. We develop a greedy heuristic graph search algorithm, which finds the target who not only covers the query labels, but also either possesses better social interactions with peers or has higher social proximity towards the user. Experiments conducted on Facebook and Twitter datasets exhibit the satisfying accuracy and encourage more advanced efforts on POI search.

1. INTRODUCTION

In social networking services such as Facebook and LinkedIn, searching for a person of interest by her name is an essential function. Given the name of the target person specified by a user u, the goal of social search is usually to accurately find the target individual t who is what user u desires. So many times, however, the user may not know the name and just have limited information about the target from their past interaction experience, or just desires for finding persons who equip with some requirement specified (e.g. experts or celebrities). While a person can be depicted by a set of labels, such as her interests, gender, skills, hometown, favorite things, the activities they had ever participated together, and so on, it is potential to develop a system allowing person search without specifying names. In this paper, we propose a novel search, *Person-of-Interest (POI) Search*, which leverages the social connections as well as the labels associated on users to identify the desired targets for users in an online social network. By specifying a small set of labels that depicts the person of interest, instead of the name information, POI Search will return a list of target persons that the user would like to find in mind.

SIGIR '15, August 9-13, 2015, Santiago, Chile



ID	Label Set
A	{c_1, c_2, c_3}
B	{c_8, c_9}
C	{c_4, c_5, c_8, c_{10}}
D	{c_4, c_6, c_7, c_9}
E	{ }
F	{c_6, c_{10}, c_{11}, c_{12}}
G	{c_1, c_2, c_3, c_5}
H	{c_1, c_2, c_3, c_8, c_{12}}

Figure 1: A toy social network to elaborate POI Search, in which each node is associated with a set of labels, and each edge has an interaction cost.

The basic idea of the proposed *POI Search* lies in that there should be some relationship on either the input query labels or the social connections (in the network) between the user and the target. We use Figure 1 to elaborate the idea. Consider user A is searching for a person who possess query labels $\{c_5, c_8, c_9\}$. There are two individuals, B and C, who possess the most labels overlapped with the query and should be returned to A. Nevertheless, Nodes B and C play different roles if considering either their social interactions with nodes possessing similar labels or their structural proximity towards user A. Note that edge weights refer to the costs of interaction between individuals, lower weights indicate better interactions and higher proximity. Node B not only has lower interaction cost with those possessing some query labels, i.e., 0.4 to G (contains c_5), 0.1 to D (contains c_9), and 0.2 to C (contains c_5 and c_8) and 0.3 to H (contains c_8), but also has higher structural proximity towards user A (e.g. via paths A-D-B and A-E-B with costs 0.4 and 0.5 respectively). On the contrary, node C has higher interaction cost to D, E, G and H than node B and lower proximity towards A (via path A-F-C with cost 1.1). Therefore, in addition to satisfying the query labels, POI Search further allow users to specify the preference between the proximity towards the user and the minimization of interact cost with other nodes containing some of query labels.

We devise a graph search algorithm to fulfill *POI Search*. The basic idea is two-fold: (a) the target tends to have higher labels overlapped with the query labels, and either (b) the cost of interactions from the target to other nodes possessing some query labels is minimized, or (c) the proximity between

the user and the target is maximized. The proposed method consists of three steps. We design and aim to optimize a *social cost* function that models part (b) and (c), and develop a greedy heuristic method to find a list of targets such that each target not only satisfies part (a) as much as possible, but also minimizes the social cost function.

Related Work. IR field focuses on optimizing Web search using social contents such as voting and tagging that are served as good summaries for web pages. Thus, social ranking methods, such as SocialPageRank [12] and HubRank [1], are validated to able to enhance the search quality. On the other hand, the viewpoint from J. Kleinberg [3], who defines it as searching over a social graph to find the set of individuals closest to a given user, is the most relevant to our POI search. Vieira et al. [11] modify the shortest paths between individuals as a structural ranking function to recommend friends for a single user. Though Schendel et al. [7] recommend items satisfying a given set of user-given query tags in a social tagging graph, social interactions are ignored. By using search log data in LinkedIn and Facebook, Huang et al. [2] and Spirin et al. [8] analyze the correlation between graph distance and various people search behaviors. In addition, Roth et al. [6] and McAuley and Leskovec [4] aims to group and recommend friends based on the social interactions between people. Though Mouratids et al. [5] use both graph and geographical distance for people search, labels that depict users are neglected.

2. PROBLEM STATEMENT

Definition 1: Social Network. A social network is an weighted graph $G = (V, E)$. Each node v has a set of labels. Each edge represents the interaction between two nodes. Edge weights refer to the interaction cost, where lower values indicate better interactions.

Definition 2: Person-of-Interest Query. A person-of-interest (POI) query consists of the user u who input such query and a set of query labels Q, in which $\forall c_i \in Q : c_i \in L$, where L is the universe set of labels.

Definition 3: Candidate Target. A candidate target v is a node whose label set L_v contain at least one query labels, i.e., $L_v \cap Q \neq \emptyset$.

The goal of person-of-interest search is to find a list of targets that not only cover the query labels as much as possible, but also have stronger relationships with respect to the user. For the latter part, considering the social interactions between the user and the *candidate targets* and the connections between candidate targets in the social network, we design the *social cost* to characterize various kinds of person-of-interest that allow a more flexible search. The intuition of the social cost is three-fold. First, it should consider the *proximity* between the user and the target. We exploit the technique of *Random Walk with Restart* (RWR) [10] to estimate such proximity. Second, we allow users to specify the *social vitality* of the target, in which targets with higher social vitality could be some celebrities while those with lower social vitality might or down-to-earth individuals or experts. To make the formulation simpler, we consider the number of the neighboring candidate targets of a target as the measure of social vitality. Third, we measure the cost of social interactions between candidate targets via the sum of weights of edges connecting them. We give the definition of social cost considering these three points.

Definition 3: Social Cost. Given a candidate target t, a set of other candidate targets U ($|U| = \pi$ and $t \notin U$), and the user $u \in V (u \notin U)$, the social cost $F(u, t)$ is defined as:

$$F(u,t) = \alpha \times (1 - RWR(u,t)) + (1-\alpha) \times \sum_{v \in U} dist(v,t), \quad (1)$$

where $\alpha = [0, 1]$ is the parameter that determines the preference between the proximity between u and t and the interaction cost between $v \in U$, π is the social vitality, $RWR(u, t)$ is the score of random walk restarting from u, and $dist(v, t)$ is the sum of edge weights in the shortest path between v and t. Note that in the execution of RWR, we use $1 - w_{u,v}$ as the edge weight since lower interaction cost means higher proximity.

Problem Definition. Given a social network G, the POI query (Q, u), and the parameters α and π, the task of person-of-interest (POI) search is to find a list of k targets S_k such that (a) the label set of each target $t \in S_k$, i.e., L_t, covers the set of query labels Q as much as possible, and (b) the social cost $F(u, t)$ is minimized. Note that we set $\alpha = 0.8$ and $\pi = 5$ by default.

THEOREM 1. *Person-of-Interest Search with the defined social cost is an NP-hard problem.*

The proof of Theorem 1 can be achieved by a reduction from *3-satisfiability* (3-SAT). Due to the page limit, however, we skip this proof.

3. THE PROPOSED METHOD

The proposed solution to person-of-interest search is a greedy heuristic method, consisting of four steps. The first is *Grouping-based Pruning* that aims to eliminate irrelevant nodes and keep the pathways with lower social costs for an efficient search. The second is to construct a *Query-Connected Graph* that combines both nodes and labels to facilitate the search. The third is *Greedy Search* that returns a list of k targets covering query labels and possessing lower social costs. The fourth is to find the *top-k targets* to be returned.

Step 1: Grouping-based Pruning. The first step is to prune irrelevant nodes that are impossible to be the targets. We group nodes containing any query label and extract the cost-effective pathways that connect the user to possible targets. A group, with respect to a query label $c_i \in Q$, is a set of connected subgraphs, in which each node contains at least c_i. After aggregating nodes into groups, we find the lowest-cost path (i.e., minimum sum of edge weights) between each pair of groups. Such paths will guide the following graph search to find the targets with lower costs. Finally, we restore the graph structure from purely the aggregated groups and the lowest-cost paths between them, and a pruned graph G' is derived.

Step 2: Query-Imposed Graph. Our approach is to find the targets satisfying the requirement of minimum cover of the query label set Q and lower social cost at the same time. Therefore, we impose query labels into G'. We create a new node for each label $c_q \in Q$, add the corresponding node v_{c_q} into G', and create edges to connect node v_{c_q} to those nodes possessing label c_q. Each newly-added edge is associated with a large positive value higher than the sum of all edge weights in G'. In the end a query-imposed graph H is constructed.

Step 3: Greedy Search. Given user u, the set of query-imposed nodes $v_{c_q} (c_q \in Q)$, and the query-imposed graph H, we develop a greedy algorithm to find the targets. We

consider the set of nodes $X = \{u\} \cup \{v_{c_q} | c_q \in Q\}$ as seeds, and find a minimum-cost tree $T = (V_T, E_T)$ (from H) that connects all the seed nodes in X. The central procedure is to greedily select the next seed x^* from $X \setminus V_T$ with the least sum of edge weights to the current node set V_T in the growing T, and add the corresponding minimum-cost path $Path(x^*, V_T)$ (in H) into T. The tree T is derived when all the seed nodes in X are included.

Step 4: Top-k Targets. For each node $x \in V_T$ in T, we compute both the number of query labels that x possesses and the score of random walk with restart $RWR(u, x)$, and calculate the rank values of both parts (higher scores, higher rank values). By computing and using the average rank value for each $x \in V_T$, we pick k nodes x_t with higher average rank values as the targets to be returned. Note that the proposed algorithm can be shown to minimize the social cost function $F(u, t)$ with 2-approximation.

4. EVALUATION

We conduct experiments on Facebook and Twitter social network data provided by SNAP[1]. The data statistics are shown in Table 1, in which CC means clustering coefficient while APL refers to average path length. Labels in both data used include the categoies of school, hometown, employer, skill, gender, location, position, etc. In the social networks, we compute the edge weights (i.e., interaction costs) using the Jaccard coefficient: $w_{u,v} = 1 - (|L_u \cap L_v|/|L_u \cup L_v|)$, where L_u is the set of labels of node u. Our general aims to test whether or not the proposed *POI Search* can help users find the desired targets accurately.

Table 1: Statistics of Facebook and Twitter data.

	#nodes	#edges	CC	APL
Facebook	4,039	88,234	0.606	4.7
Twitter	81,306	1,768,149	0.565	4.5

POI Search is compared with three competitors: (a) *label matching* (LM): using the query label set Q to find a list of possible target nodes t such that the labels of t have the higher overlap with Q, $LM(t, Q) = \frac{|L_t \cap Q|}{|L_t|}$; (b) *Center-Piece Subgraph* (CePS) [9] with OR operation: consider those nodes that contain at least one label in Q as source nodes in CePS, run the CePS algorithm, and return a list of possible targets t with higher CePS scores $CePS(t, Q)$; and (c) *CePS+LM*: the selection procedure is similar as CePS but return the list of possible target nodes with toper average rank values of CePS and LM, $CePSLM(t, Q) = \frac{1}{2}(rank(CePS(t, Q) + rank(LM(t, Q)))$.

In the evaluation setting, we first compile the universe set of all labels L, and then randomly select n labels from L to be the query set of labels Q, where $|Q| = n$ and $\forall c_q \in Q, freq(c_q) > \pi$ ($freq(c_q)$ is the number of nodes possessing label c_q). By randomly picking a node as the user u who is assumed to input a set of query labels Q, we perform an exhaustive search to find the list of top-θ optimal targets with the minimum scores of $F(u, t)$ and the maximum cover of query labels Q. The complexity exhaustive search is $O(|V|^2 \binom{|U|}{\pi} \log |V|)$ in the worst case. We utilize parallel programming to find the top-θ optimal targets for each pair of query Q and user u (we choose $\theta = 5$ here). To control the hardness of the problem, in the general evaluation, for each query Q, we select the user u whose top-k optimal targets locates within *3-degrees-of-separation* neighbors. We generate 5,000 pairs of query Q and user u possessing such requirement as the test data, i.e., $p_test = (Q, u) \in P, |P| = 5000$. Given the top-$k$ targets $S_k(Q, u)$ returned by a method and the ground-truth top-θ optimal targets $S^*_\theta(Q, u)$, we define the evaluation metric using accuracy: $accuracy = \sum_{(Q,u)\in P} hit(S_k(Q, u), S^*_\theta(Q, u))/|P|$, where $hit(S_k(Q, u), S^*_\theta(Q, u)) = 1$ if $|S_k(Q, u) \cap S^*_\theta(Q, u)| \neq 0$; otherwise: $hit(S_k(Q, u), S^*_\theta(Q, u)) = 0$. It can be observed that given a fix number of θ, as k increases, the search task tends to be easier and will obtain a higher accuracy. That says, k determines the strictness of the evaluation.

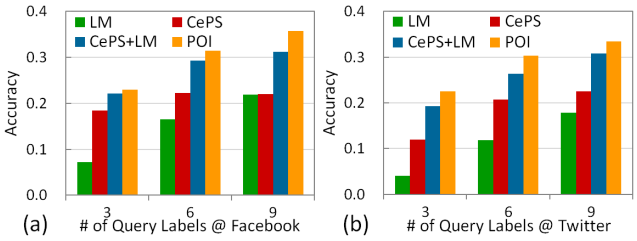


Figure 2: Accuracy scores by varying the number of query labels in Facebook ad Twitter data.

The evaluation plan consists of five parts: (a) general evaluation: varying the number of query labels $|Q|$ to show the performance of *POI Search*, larger set of Q means more specific search query; (b) varying the number of degrees of separation, a higher separation number refers to a harder search task; (c) varying the extent of *label rarity* in the query, defined by $LR(c_i) = 1 - \frac{|v \in V : c_i \in L_v|}{|V|}$, a query with higher label rarity indicates an easier search task; (d) varying the *top-k strictness* in the accuracy metric, a greater number of k means a loose accuracy definition; and (e) varying the parameter α in the objective function $F(u, t)$, a lower value of α leads to an easier search task because it tends to find the targets with higher proximity scores to user u.

Experimental results are shown in Figure 2-5. By varying the number of query labels, in Figure 2, we can find that the proposed POI search outperforms the other competitors, especially when the number of query labels increases. It is because more labels provide more evidences about the desired targets. LM cannot work well because it considers only the label matching while CePS uses only the structural connectivity. Though the integration of CePS and LM (CePS+LM) can boost the effectiveness to some extent, the accuracy scores are still lower than POI. Such results mean the heuristic of random walk cannot work well on finding the desired targets that minimize $F(u, t)$, and our POI search method is better to be closer the optimal solution.

Figure 3 reports the accuracy scores by varying the degree of separation of the ground-truth targets (selecting (Q, u) test pairs that satisfy a certain separation for the evaluation). As the degree of separation increases, the accuracy scores of all the methods get worse. We think it is because if the desired targets are farther from the users, the nodes satisfying query labels, the social costs incurred by the connections between nodes satisfying query labels, and their prox-

[1]http://snap.stanford.edu/data/egonets-Facebook.html
http://snap.stanford.edu/data/egonets-Twitter.html

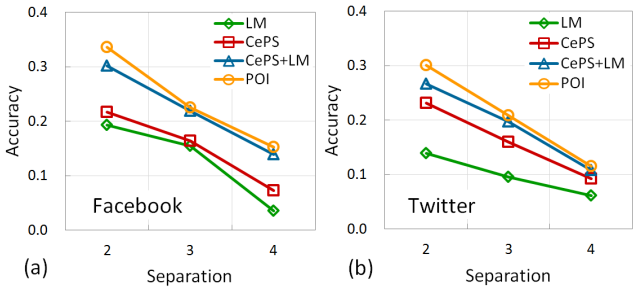


Figure 3: Accuracy scores by varying the degree of separation between user and target.

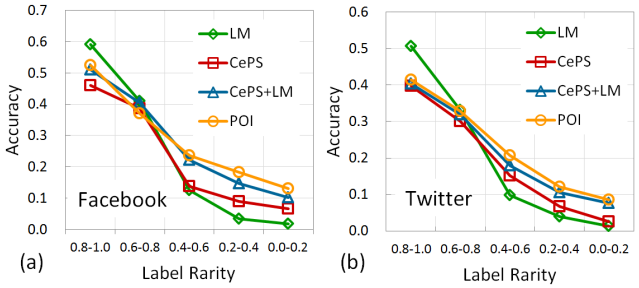


Figure 4: Accuracy scores by varying the label rarity of query labels.

imity to the users would increase accordingly. These factors lead to more candidate targets with lower social costs, and make the search task more challenging.

We shown the results of different extent of label rarity in Figure 4. We can find query labels with higher label rarity values lead to higher accuracy scores. As the label rarity of the query decreases, i.e., the query labels are frequently possessed by many nodes, the accuracy scores go down drastically. These results are reasonable since nodes whose labels with higher label rarity are essentially much less than those with lower label rarity. Therefore, it is natural to easily deal with query labels with higher rarity. Such results also suggest that when searching for desired targets, it would be beneficial for users to provide labels that tend to uniquely recognize the targets.

Figure 5 shows the results of sensitivity analysis for the parameters of (a) top-k strictness and (b) α in $F(u,t)$. A higher number of k in the accuracy metric, i.e., more targets are returned under a fixed number $\theta = 5$ of the ground-truth optimal targets, leads to a higher accuracy score, because the evaluation metric becomes looser. One can treat the top-k strictness as the number of returned targets that the users would view. If the users would like to explore more targets return, the possibility of finding the desired targets will be boosted. On the other hand, in Figure 5(b), a higher α value tends to guide the search to prefer the proximity between nodes possessing query labels and the user, and gets higher accuracy scores. An extreme case is to find the targets considering only the proximity part (i.e., $\alpha = 0.99$), which leads to high accuracy. On the contrary, lower α values shift the search to weight the social costs between the candidate targets, which could make the search towards a local optima, and thus result in lower accuracy.

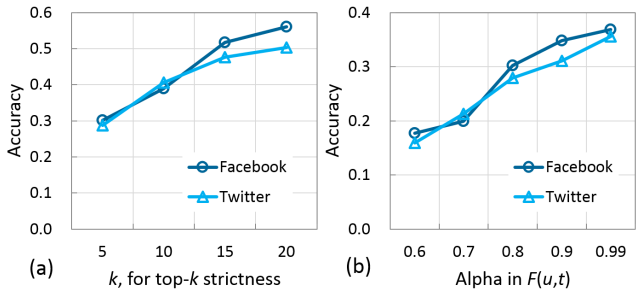


Figure 5: Accuracy by varying (a) top-k strictness, and (b) α parameter in the social cost function $F(u,t)$ using the proposed method.

5. CONCLUSIONS

This paper proposes *POI Search* to find persons of interest for users in online social networks. Considering the label cover, the structural proximity, and the social interaction between the query and the target, we devise a greedy heuristic algorithm to find a list of targets. Experimental results show the promising accuracy. Now we are evaluating POI Search using user study by a developed Facebook application. The future work is to extend POI Search into the realm of heterogeneous information networks.

6. REFERENCES

[1] S. Bao, G. Xue, X. Wu, Y. Yu, B. Fei, and Z. Su. Optimizing web search using social annotations. In *WWW*, 2007.

[2] S.-W. Huang, D. Tunkelang, and K. Karahalios. The role of network distance in linkedin people search. In *SIGIR*, 2014.

[3] J. Kleinberg. Social networks, incentives, and search. In *SIGIR*, 2006.

[4] J. McAuley and J. Leskovec. Learning to discover social circles in ego networks. In *NIPS*, 2012.

[5] K. Mouratidis, J. Li, Y. Tang, and N. Mamoulis. Joint search by social and spatial proximity. In *IEEE TKDE*, 2015.

[6] M. Roth, A. Ben-David, D. Deutscher, G. Flysher, I. Horn, A. Leichtberg, N. Leiser, Y. Matias, and R. Merom. Suggesting friends using the implicit social graph. In *KDD*, 2010.

[7] R. Schenkel, T. Crecelius, M. Kacimi, S. Michel, T. Neumann, and G. Weikum. Efficient top-k querying over social-tagging networks. In *SIGIR*, 2008.

[8] N. V. Spirin, J. He, M. Develin, K. G. Karahalios, and M. Boucher. People search within an online social network: Large scale analysis of facebook graph search query logs. In *CIKM*, 2014.

[9] H. Tong and C. Faloutsos. Center-piece subgraph: Problem definition and fast solutions. In *KDD*, 2006.

[10] H. Tong, C. Faloutsos, and J.-Y. Pan. Fast random walk with restart and its applications. In *ICDM*, 2006.

[11] M. V. Vieira, B. M. Fonseca, R. Damazio, P. B. Golgher, D. D. C. Reis, and B. Ribeiro-Neto. Efficient search ranking in social networks. In *CIKM*, 2007.

[12] Y. Yanbe, A. Jatowt, S. Nakamura, and K. Tanaka. Can social bookmarking enhance search in the web. In *JCDL*, 2007.