Joint Search by Social and Spatial Proximity (Extended Abstract)

Kyriakos Mouratidis
School of Information Systems
Singapore Management University
kyriakos@smu.edu.sg

Jing Li, Yu Tang and Nikos Mamoulis
Department of Computer Science
University of Hong Kong
{jli, ytang, nikos}@cs.hku.hk

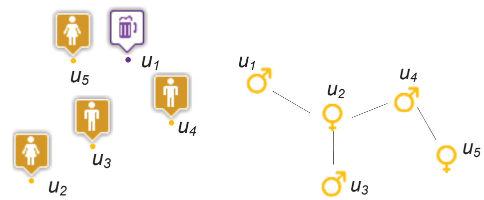
***Abstract*—The diffusion of social networks introduces new challenges and opportunities for advanced services, especially so with their ongoing addition of location-based features. We show how applications like company and friend recommendation could significantly benefit from incorporating social and spatial proximity, and study a query type that captures these two-fold semantics. We develop highly scalable algorithms for its processing, and use real social network data to empirically verify their efficiency and efficacy.**

I. Introduction

The emergence of social networks (*SN*s) brings a new era in the organization and browsing of online information. Manufacturers and service providers are becoming increasingly interested in exploiting popular *SN*s to promote their products and services. At the same time, location-based services are an indispensable feature in *SN*s. The most popular *SN*, Facebook, includes a set of location-based features, while others (such as Foursquare) rely explicitly on the management of user locations. Motivated by this trend, we investigate the integration of social and spatial information in a single query.

Consider a service like badoo.com, where a user u_1 who is looking for company to have lunch or watch a movie, may browse the profiles of nearby users and invite them to join him/her. Existing systems apply a traditional k-nearest neighbor query, potentially with some binary conditions (regarding age, sex, etc), to provide u_1 with the profiles of users in the vicinity. While recommended users are indeed near u_1 geographically, her true preferences of companions would be better captured if *SN* information was also taken into account. Assume, for example, that the users' Euclidean coordinates and social connections are as shown in Figures 1(a) and 1(b), respectively. The closest user to u_1 in the spatial domain is u_5. However, u_4 might be a better match because he locates only slightly farther (compared to u_5) but is "closer" in the social network. Conversely, the closest user socially (u_2) may be too far spatially. Therefore, to provide meaningful recommendations, both *social proximity* and *spatial proximity* should be integrated in the search.

In this extended abstract we summarize [1], where we propose and study the *social and spatial ranking query* (*SSRQ*). *SSRQ* reports the top-k users in the *SN* based on a ranking function that incorporates social and spatial distance from the query user.



(a) Spatial domain (b) Social network

Fig. 1. Motivating example

II. Problem Formulation

We consider a set of *SN* users with known Euclidean coordinates. The *SN* is modeled as an undirected, weighted graph containing an edge for every pair of users that are friends. The edge weight indicates the strength of their relationship – the smaller the weight, the stronger the friendship.

We define *spatial proximity* between users u_i and u_j as their Euclidean distance $d(u_i, u_j)$. We measure their *social proximity* as the shortest path distance between them in the *SN*, denoted as $p(u_i, u_j)$. We use this measure because it is demonstrated to effectively capture social proximity/influence [2], [3]. Following common practice in combining measurements from different domains, we apply a linear function over the (normalized) social and spatial proximity to rank users. Given a query user u_q, the *joint distance* of $u_i \in U$ is:

$$f(u_q, u_i) = \alpha \cdot p(u_q, u_i) + (1 - \alpha) \cdot d(u_q, u_i) \quad (1)$$

where α is a (user- or application-specified) real number between 0 and 1 that determines the relative significance of proximity in the two domains. The *SSRQ* query returns the k users with the smallest joint distance to u_q (for a positive integer k). Note that our definition uses normalized social and spatial proximities, by dividing raw distances with the maximum pairwise distance in Euclidean space and in the social graph, respectively.

III. SSRQ Algorithms

We first present two simple solutions, *Social First Approach* (SFA) and *Spatial First Approach* (SPA); then we hybridize them into an elaborate algorithm, *Twofold Search Approach* (TSA); finally, we describe our most advanced solution, *Aggregate Index Search* (*AIS*), which summarizes both social and spatial information into the same index, and runs a unified search on that index.

SFA considers users in increasing social distance from u_q, using Dijkstra's algorithm. For every user popped from Dijkstra's search heap, *SFA* also computes her Euclidean distance to u_q and, in turn, her joint distance. The k closest users found so far are kept in an interim result. Let u be the last user popped, and f_k be the joint distance of the k-th (i.e., most distant) user in the interim result. The social distance of every un-processed user is at least $p(u_q, u)$. Thus, when $\alpha \cdot p(u_q, u)$ becomes greater than f_k, the interim result is finalized.

SPA considers users in increasing Euclidean distance, using an incremental nearest neighbor search around u_q. For every encountered user, *SPA* computes her social distance to u_q. It maintains in an interim result the k encountered users with the smallest joint distances to u_q. Let u be the last encountered user. The interim result is finalized when $(1-\alpha) \cdot d(u_q, u)$ becomes greater than f_k (value f_k is defined as in *SFA*).

TSA performs two incremental searches around u_q, one in the social and the other in the spatial domain. In its first phase, *TSA* retrieves users from both domains in a round-robin fashion. Users encountered in the social domain, have their joint distance computed directly, and are used to maintain an interim result of the k best. Instead, users encountered in the spatial domain are held in a candidate set (to defer computation of their social distance). Let t_p and t_d be the social and spatial distance of the last encountered user in the respective domain. The first phase of *TSA* terminates when $\alpha \cdot t_p + (1-\alpha) \cdot t_d \geq f_k$. This condition guarantees that the only users that may belong to the final result are either in the interim result or in the candidate set. In the second phase of *TSA*, only the social search continues. Once a user from the candidate set is encountered, her joint distance becomes known; if it is smaller than f_k, the interim result is updated accordingly. *TSA* terminates when no un-processed user from the candidate set may enter the interim result (taking into account her actual spatial distance and that her social distance is at least as large as the social distance t_p where social search has reached).

In [1] we describe optimizations for *TSA*. One of them replaces round-robin probing (in the first phase of *TSA*) with the *Quick Combine* strategy [4]. Another enhances *TSA* by the landmark approach [5]. This approach associates each user with a vector that stores pre-computed social distances from a set of anchor users in the social graph. That vector is used at runtime to derive a lower bound of $p(u_q, u)$ for every user u in the candidate set (in the second phase of *TSA*).

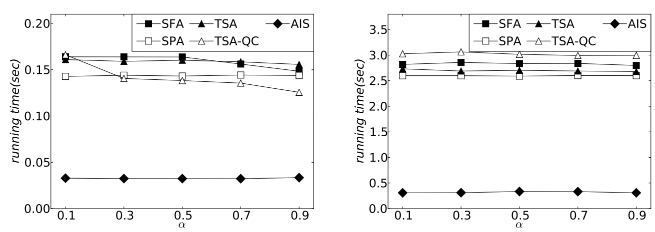
Although *TSA* utilizes tighter bounds than *SFA* and *SPA*, it may still visit numerous users who are close in the social graph but far away in the spatial domain, and vice versa. The reason is that the two searches are oblivious of each other, and may be accessing completely different users. This motivates *AIS*, which summarizes both social and spatial information into the same index, and runs a unified search on it.

The index of *AIS* is a multi-level spatial partitioning structure, where each node is augmented with a *social summary*. *AIS* builds on the landmark approach. The social summary of a node is produced by the landmark vectors of users inside its spatial extent, using a novel aggregation method. Given the landmark vector of the query user u_q, the social summary of a node can be used to derive a lower bound for the social distance of all underlying users to u_q. On the other hand, the spatial extent of the node provides a lower bound for the spatial distance of these users to u_q too. Combining the two bounds into Equation 1, we derive a lower bound for the joint distance of any user under the node. The latter enables a branch-and-bound search that visits index nodes in increasing order of their lower bounds. *AIS* terminates when f_k in its interim result is smaller than the lower bound of the next node to be visited.

In [1] we enhance *AIS* with optimizations. We accelerate graph search by a hybrid bi-directional shortest path technique, which uses Dijkstra's algorithm in one direction and A* in the other. We also employ computation sharing in deriving shortest paths from u_q to different users. Finally, as *AIS* proceeds and explores a larger part of the *SN*, we exploit the knowledge gained to tighten the landmark-derived lower bounds.

IV. Representative Experiments

We use two real datasets. *Gowalla*, from *snap.stanford.edu*, contains 196K users. *Foursquare*, used in [6], contains 1.88M users. Our implementation is in C++. Data and indices are kept in main memory. In Figure 2 we test different values of α, i.e., different weighing of social versus spatial proximity. Label *TSA* corresponds to the landmark-aided version of *TSA*, while *TSA-QC* to its *Quick Combine* variant.



(a) Running time in *Gowalla* (b) Running time in *Foursquare*

Fig. 2. Effect of α

SFA examines vertices in increasing social distance order, which implies that for large α the first few processed vertices are highly likely to already produce the result. *TSA* and *TSA-QC* are also more socially-led (than spatially), since their second phase relies entirely on graph search, thus benefiting from a large α. *SPA* is spatially-led and hence its performance worsens with α. *AIS* is robust to α and retains a clear lead over alternatives, which is the case in all experiments in [1].

References

[1] K. Mouratidis, J. Li, Y. Tang, and N. Mamoulis, "Joint search by social and spatial proximity," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 3, pp. 781–793, 2015.

[2] M. V. Vieira, B. M. Fonseca, R. Damazio, P. B. Golgher, D. C. Reis, and B. A. Ribeiro-Neto, "Efficient search ranking in social networks," in *CIKM*, 2007, pp. 563–572.

[3] P. Yin, W.-C. Lee, and K. C. K. Lee, "On top-k social web search," in *CIKM*, 2010, pp. 1313–1316.

[4] I. F. Ilyas, G. Beskales, and M. A. Soliman, "A survey of top-*k* query processing techniques in relational database systems," *ACM Comput. Surv.*, vol. 40, no. 4, 2008.

[5] A. V. Goldberg and C. Harrelson, "Computing the shortest path: *A** search meets graph theory," in *SODA*, 2005, pp. 156–165.

[6] M. Sarwat, J. Bao, A. Eldawy, J. J. Levandoski, A. Magdy, and M. F. Mokbel, "Sindbad: a location-based social networking system," in *SIGMOD Conference*, 2012, pp. 649–652.