Group Preference Queries for Location-Based Social Networks

Yuan Tian [1], Peiquan Jin [1,2], Shouhong Wan [1,2], Lihua Yue [1,2]

[1] School of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China

[2] Key Laboratory of Electromagnetic Space Information, Chinese Academy of Sciences, Hefei 230027, China
jpq@ustc.edu.cn

Abstract. Location-based social networks involve a great number of POIs (points of interest) as well as users' check-in information and their ratings on POIs. We note that users have their own preferences for POI categories. In addition, they have their own network of friends. Therefore, it is necessary to provide for a group of users (circle of friends) a new kind of POI-finding service that considers not only POI preferences of each user but also other aspects of location-based social networks such as users' locations and POI ratings. Aiming to solve this problem, in this paper we present a new type of query called *Spatial Group Preference* (*SGP*) query. For a group of users, an SGP query returns top-*k* POIs that are most likely to satisfy the needs of users. Specially, we propose a new evaluation model that considers user preferences for user preferences for POI categories, POI properties including locations and ratings, and the mutual influence between POIs. Based on this model, we develop algorithms based on R-tree to evaluate SGP queries. We conduct experiments on a simulation dataset and the results suggest the efficiency of our proposal.

Keywords: Location-Based Social Network; Group Preference; Spatial Query

1 Introduction

Recently, with the popularity of GPS-enabled smart phones, location-based social network becomes a hot topic. Location-based social networks allow a group of users (circle of friends) to share their location information each other. Together with other information about POIs (points of interest) such as POI locations, POI category labels (i.e., restaurant, hotel, café, etc.), and POI ratings, we can provide a variety of services for people [1].

In this paper, we study an interesting type of query called *Spatial Group Preference* (SGP) query. Given a set of POIs and a group of users, an SGP query retrieves *k* POIs that are expected to satisfy the overall need of the group of users. We assume that each user in a group has his or her current location and a preference list for POI categories, e.g., restaurant, theater, and shopping mall.

One basic solution to the SGP query is using distance to select the best candidates, which has been widely studied in previous spatial databases. This approach neglects the users' preferences for POI categories as well as the POI properties (e.g., POI ratings). Many previous studies [11, 19] have proposed to integrate POI properties into POI recommendation. However, these works are not user-aware because they do not consider the preferences of users. To the best of our knowledge, there are very few works that can be directly used to effectively answer SGP queries. As a result, it lacks of effective evaluation models that can evaluate the quality of POIs according to the needs of a group of users.

In this paper, we first give the formal definition on the SGP query. Then, we propose a new evaluation model for SGP queries. Our model integrates different metrics, including distance, POI ratings, user preferences, and influence between POIs, to provide a comprehensive ranking for POI candidates. Based on the model, we propose an R-tree-based algorithm for evaluating SGP queries. Briefly, we make the following contributions in this paper:

- We define a new kind of query called SGP query for location-based social networks. (Section 3)
- We propose a new evaluation model that considers user preferences for user preferences for POI categories, POI properties including locations and ratings, and the mutual influence between POIs. Compared with existing approaches, this model is user-aware and can provide more reasonable ranking for POIs. (Section 4)
- We conduct experiments on a simulation dataset as well as on a real dataset with respect to various configurations. The results suggest the efficiency of our proposal. (Section 5)

2 Related work

In 2004, Papadias et al. first proposed the concept of group nearest neighbor (GNN) query [2] in spatial databases, which is to find a suitable gathering place for a group of users scattered throughout the Euclidean distance space. They established the R-tree index to organize candidate gathering points and further extended this approach to Aggregate Nearest Neighbor (ANN) query [3]. Both of them studied the group nearest neighbor query in Euclidean distance space, which is not suitable for road networks [4, 5] due to the differences between road networks and the Euclidean distance space. There are other researches [6-10] paying attention to improve the efficiency and to reduce the I/O cost of queries in Euclidean distance space or road network. However, all the above works do not consider group preferences.

In 2007, Yiu et al. proposed a new kind of query called top-*k* spatial preference query [11]. This query returns top-*k* candidate objects whose rankings are defined by the quality of other objects around them. Further, the work in [12] proposed an improved scoring method based on textual similarity for getting candidate objects. The idea of considering the influence of near objects was also applied to road networks [13-16]. For example, in [16] the authors mapped distances and the scores of other kinds of objects surrounding the candidate object into a two-dimensional space based on distance and score, and then used the dynamic skyline method to reduce the number of candidate objects. However, existing works on spatial preference queries are not user-aware, meaning that they do not consider the user preferences for POIs.

In 2016, Miao Li et al. studied location-aware group preference queries [17] that are a combination of GNN query and group preference. They return top-k sites that consist of those sites having the minimum distance to the locations of scattered uses in a group and the sites matching the group preference. However, this approach only uses distance to evaluate the group preference queries. It is not suitable for location-based social networks where POI ratings need to be considered in query evaluation.

3 Problem Statement

We assume that there are m categories of POIs, which are represented as $C = \{c_1, c_2, c_3, c_4, \dots, c_m\}$. Each $c_i (1 \le i \le m)$ represents a category. A set of POIs is represented by $P = \{p_1, p_2, p_3, p_4, \dots, p_l\}$. Each $p_i (1 \le i \le l)$ is represented by a triple $\langle p_i.loc, p_i.t, s(p_i)\rangle$. Here, $p_i.loc$ is the location of p_i; $p_i.t$ is the category associated with p_i and $s(p_i)$ is the rating of p_i, which is a value between 0 and 1 and can be obtained from social network platforms.

Definition 1 (SGP Query). *Given a set of POIs P, a group of querying users $Q = \{u_1, u_2, u_3, \dots, u_n\}$, a targeted category c, and an integer k, a Spatial Group Preference (SGP) query retrieves a set S ⊆ P that consists of k POIs, such that:*

(1) $(\forall x)(x \in S) \rightarrow x.t = c$

(2) $(\forall x)(\forall y)(x \in S \wedge y \in P - S \wedge y.t = c) \rightarrow sd(x, Q) \ge sd(y, Q)$

Here, sd(x, Q) returns the satisfaction degree between x and Q. □

$Q = \{u_1, u_2, u_3, \dots, u_n\}$ represents a set of users and each user u_i is denoted as a tuple $\langle u_i.loc, u_i.CW\rangle$, where $u_i.loc$ is the current location of user u_i and $u_i.CW$ represents the preferences for POI categories of the user. To quantify the preferences, we assign a weight to each preferred POI category for each user. Consequently, we have $u_i.CW = \{\langle u_i.c_1, u_i.w_1\rangle, \langle u_i.c_2, u_i.w_2\rangle, \dots, \langle u_i.c_{m'}, u_i.w_{m'}\rangle\}$. Each tuple in $u_i.CW$, e.g., $\langle u_i.c_1, u_i.w_1\rangle$, means that the category c_1 has a weight of w_1. We assume that $0 \le u_i.w_j \le 1$, yielding $\sum_{j=1}^{m'} u_i.w_j = 1$. In addition, we use C_Q^r to represent all the categories contained in Q, i.e., $C_Q^r = \bigcup_{i=1}^{n} \bigcup_{j=1}^{m'} u_i.c_j$.

The key issue of answering an SGP query is how to define the satisfaction degree model $sd(x, Q)$. In location-based social networks, there are mainly two factors that impact the satisfaction degree of POIs: (1) the distances between the locations of users in the group and candidate POIs labeled with the category given in the query; (2) the influence of surrounding candidate POIs that have categories in C_Q^r. We define two metrics, namely *distance relevance* and *preference relevance*, to formally reflect these two factors in computing satisfaction degree.

Definition 2 (Distance Relevance). *Given a candidate POI p and an SGP query Q, the distance relevance between p and Q is defined by (1) [17]:*

$$\delta(p,Q) = 1 - \frac{d(Q,p)}{D_{max} \cdot |Q|} \tag{1}$$

Here, $d(Q,p)$ represents the sum of the distances of each user to p. D_{max} is the diagonal length of the MBR covered by all the involved POIs. □

Definition 2 (Preference Relevance). *Given a candidate POI p and an SGP query Q, the preference relevance between p and Q is defined by (2):*

$$\tau(p,Q) = \sum\nolimits_{c_i \in C_Q^r} \frac{1}{|Q|} \cdot \frac{W(c_i) \cdot r_{c_i}^r(p)}{1 + \frac{d\left(p, {p'}_{c_i}^r(p)\right)}{r}} \tag{2}$$

Here, ${p'}_{c_i}^r(p)$ is one POI, labeled with c_i, within the range r of candidate POI p. And $r_{c_i}^r(p)$ indicates the comprehensive score of ${p'}_{c_i}^r(p)$, which is obtained from social network. We assume there is $0 \leq r_{c_i}^r(p) \leq 1$. A greater value represents a greater comprehensive score. And $W(c_i)$ indicates the degree of attention of all users in group to category c_i. The calculation of it has been mentioned above. There $d\left(p, {p'}_{c_i}^r(p)\right)$ shows the distance between p and p' in Euclidean distance space. Note that gathering category c is the special case of C_Q^r, and $d\left(p, {p'}_{c}^r(p)\right) = 0$ this time. The similar form of this formula is used to balance the relations between the textual relevance and network proximity of two objects. Here, we use the score of candidate POI affected by other POIs with group preference around it instead of textual relevance and definition in the denominator is similar to original definition in that paper. □

Definition 3 (Satisfaction Degree). *Given a set of POIs P, a group of querying users $Q = \{u_1, u_2, u_3, \ldots, u_n\}$, the satisfaction degree of p as a gathering point of Q is defined by (3), where α is a smoothing parameter.*

$$sd(p,Q) = \alpha \cdot \delta(p) + (1-\alpha) \cdot \tau(p)\ (1) \tag{3}$$

When $\alpha = 1$, the SGP query is similar to the GNN query. If $\alpha = 0$, the SGP query becomes a variant of the top-*k* spatial preference query. □

4 Algorithms for SGP Queries

The SGP query is a new query that has not been mentioned before, so there is no ready-made method to solve the problem. But there are some basic approaches to it.

4.1 Baseline Algorithm（BA）

In the baseline algorithm, we build Rtrees for POIs in P labeled with each category, respectively. Rtree is one of the most popular and widely used data structure for indexing spatial objects. First of all, traversal the R_c (Rtree of category c) starting from the root node. If it is a non-leaf node, we execute a recursive call to all of its child nodes; else for each POI p in the leaf node, we traversal each R_{c_i} with $c_i \in C_Q^r$

from top to bottom and compute the value according to the model of satisfaction. At the end of the algorithm, it returns top-k POIs labeled with category c, and with the highest satisfaction.

4.2 Pruning Algorithm（PA）

In the BA algorithm, all candidate POIs as well as POIs around them labeled with group preference categories are retrieved. Thus, we present the pruning algorithm to reduce the number of POIs are retrieved. In this method, aRtrees [11] instead of Rtree are established for all POIs in P, according with different categories. The aRtree is similar to Rtree, but is added some aggregate information to each node in Rtree. In this paper, aggregate information tagged to a node is max comprehensive score of POIs in the child nodes of this node. Algorithm 1 shows the details of the pruning process.

```
Algorithm 1. $f_{pruning}(node\ N)$
If $sd^{+}(N) > kth\_bestvalue$ // pruning strategy 1
    If $N$ is a non-leaf node
        execute $f_{pruning}()$ recursively for each child node of $N$
    Else
        For all entities $e \in N$
            For each $c_i \in C_Q^r \wedge c_i \neq c$ do
                $Score\_comput(e, N, R_{c_i}.Root)$
            Compute the value of $sd(e)$ and Update $H$ and $kth\_bestvalue$
$Score\_comput(Point\ p$, Node $N_1$, $Node\ N_2)$
If $sd^{+}(N_1, N_2) > kth\_bestvalue$
    If $N_2$ is a non-leaf node then
        execute $Score\_comput()$ recursively for each child node of $N$
    Else
        For each entry $e' \in N_2$ and $dist(p, e') \leq r$ do
            Compute and return the actual value of $\frac{1}{|Q|} \cdot \frac{w(c_i) \cdot r_{c_i}^{r}(p)}{1+\frac{d(p,e')}{r}}$
End $f_{pruning}$
```

Here, $sd^{+}(N)$ and $sd^{+}(N, N')$ are upper bounds of $sd(p, Q)$, where p is in node N. Only if these upper bounds are greater than kth$_bestvalue$ (a global variable indicating the kth highest satisfaction degree), the algorithm will continue. Firstly, we invoke function $f_{pruning}()$ at the root node of aR_c (indexing all the POIs labeled with category c) and traverse it from top to bottom. Then, $kth\_bestvalue$ is compared with the upper bound of each node that is visited in the traversal route. This procedure is recursively performed on a non-leaf node until a leaf node is encountered. The $Score\_comput$ () method is invoked to compute the $\frac{1}{|Q|} \cdot \frac{w(c_i) \cdot r_{c_i}^{r}(p)}{1+\frac{d\left(p, p'^{r}_{c_i}(p)\right)}{r}}$ in a$\mathrm{R}_{\mathrm{c_i}}$ for each POI p in the leaf node of aR_{c}. Finally, H(one global variable that maintains the results to be returned) and $kth\_bestvalue$ are updated according to $sd(p,Q)$ of p. The implementation of the $Score\_comput()$ method is similar to that of $f_{pruning}()$. Line 10-11 describes the recursive calls on non-leaf nodes of aR$_{\mathrm{c_i}}$ and Line 13-14 indicates how to obtain the value of $\frac{1}{|Q|}\cdot\frac{w(c_i)\cdot r^{r}_{c_i}(p)}{1+\frac{d\left(p,p'^{r}_{c_i}(p)\right)}{r}}$ for p in aR$_\mathrm{c}$ and $p'^{r}_{c_i}(p)$ in aR$_{\mathrm{c_i}}$.

4.3 Optimized Pruning Algorithm (OPA)

In fact, same POIs with group preference may be within the ranges r of the adjacent POIs associated with category c. In other words, the satisfaction degree of respective candidate POIs in one leaf node of aR_c may be affected by the same POIs with group preference. Therefore, we take a set of all the POIs in one leaf node N_1 of aR_c, instead of just one POI, within the range r of N_1 one time and compute the component score on aR_{c_i}. We describe this improved pruning scheme in Algorithm 2.

```
Algorithm 2. OPA(Node $N_1$, Node $N_2$)
    $P = \{p|p\in N_1\}$
    If $sd^{+}(N_1,N_2) > kth\_bestvalue$
        If $N_2$ is a non-leaf node then
            execute $Score\_comput()$ recursively for each child node of $N_2$
        Else
            For each entry $e \in P$ do
                For each entry $e' \in N_2$ and $dist(e,e') \le r$ do
                    Compute and return the actual value of $\frac{1}{|Q|}\cdot\frac{w(c_i)\cdot r^{r}_{c_i}(e)}{1+\frac{d(e,e')}{rad}}$
End OPA
```

Algorithm 2 is an optimized version of Algorithm 1. Nodes N_1 and N_2 are two parameters of the function, where N_1 is visited leaf node in aR_c and N_2 is root node of a$R_{c_i}(c_i \in C^{r}_{Q} \wedge c_i \neq c)$. When a leaf node N_1 is visited when traversing aR_c, its all candidate POIs are obtained one time and stored in a set P. Then, the component scores $\frac{1}{|Q|}\cdot\frac{w(c_i)\cdot r^{r}_{c_i}(e)}{1+\frac{d\left(p,p'^{r}_{c_i}(e)\right)}{r}}$ of them are computed concurrently at a single traversal of the aR_{c_i}. Obviously, the candidate POI set corresponding to one leaf node in aR_c access less tree nodes in aR_{c_i}, which also speeds up the query.

5 Experiments

We conduct experiments on a simulated data set that contains a set of POIs. Each POI is tagged with a geographical location and a category description. We randomly assign a value between 0 and 1 for each POI to represent the rating in social networks. All algorithms are implemented in Java, and an Intel(R) Core(TM) i3-4210M CPU @2.60 GHz with 8 GB RAM is used for the experiments. The index structure is memory resident, and the maximum number entries of a node is set to 100. In all experiments, we run 100 queries and report the average costs of the queries. The default settings of the parameters are: C_Q^r=8, *n*=8, *r*=100, *a*=0.5, and *k*=8.

We first evaluate the impact of the number of POIs. The runtime and pruning rates are shown in Table 1 and Table 2, respectively. The runtime increases in all algorithms when the number of POIs ranges from 100k to 2M. The runtime of BA increases rapidly, because it employs no pruning strategies. OPA gets the best runtime due to its optimization on PA. The pruning rate also increases in all algorithms with the increase of the number of POIs.

Table 1. Runtime (ms) of varying the number of POIs (*l*)

#POI(l) *Algorithm*	100000	500000	1000000	1500000	2000000
BA	888.7	8328.24	21188.5	37399.36	56305.26
PA	284.92	876.38	1606.84	2295.94	3151.42
OPA	170	619.2	1173.54	1693.74	2425.86

Table 2. Pruning rate (%) of varying the number of POIs (*l*)

#POI(l) *Algorithm*	100000	500000	1000000	1500000	2000000
BA	0	0	0	0	0
PA	16.15	55.14	65.98	71.85	75.35
OPA	16.78	55.79	66.61	72.46	75.96

Next, we study the performance of our algorithms for different numbers of categories. As shown in Table 3, the runtime of all algorithms decreases with the increase of *m*. In the case of a constant number of POIs that are indexed, the larger the number of POIs is, the smaller average number of POIs labeled with each category is. This leads to the fact that a smaller number of POIs labeled with group-preference categories are retrieved.

Table 3. Runtime (ms) of varying the number of categories (*m*)

#Categories(m) *Algorithm*	4	8	16	32
BA	72082.85	25236.1	9400.4	3808.6
PA	3051.68	1595	866.24	501.34
OPA	2281.16	1156.04	631.56	356.54

6 Conclusion

In this paper we study a new kind of spatial queries named spatial group preference queries in location-based social networks. We define a satisfaction degree model to measure whether a candidate POI meets the group's needs. Then, we propose two algorithms based on R-tree-based pruning strategies. Our preliminary experimental results over a simulation dataset show the efficiency of the algorithms. Our future work will focus on devising new efficient indexes to accelerate SGP queries. We will also evaluate our proposal on real datasets.

Acknowledgements. This work is supported by the National Science Foundation of China (61379037 and 61672479). Peiquan Jin is the corresponding author.

References

1. Jin, P., Cui, T., Wang, Q., Jensen, C. S., Effective similarity search on indoor moving-object trajectories, In: Navathe, S., Wu, W., et al. (eds.) DASFAA 2016. LNCS, vol. 9643, pp. 181–197. Springer, Cham (2016).
2. Papadias, D., Shen, Q., Tao, Y., et al.: Group nearest neighbor queries. In: 20th International Conference on Data Engineering, pp. 301–312. IEEE, Boston, MA, USA (2004).
3. Papadias, D., Tao, Y., Mouratidis, K., et al.: Aggregate nearest neighbor queries in spatial databases. ACM Transactions on Database Systems 30(2), 529–576 (2005)
4. Yiu, M.L., Mamoulis, N., Papadias, D.: Aggregate nearest neighbor queries in road networks. IEEE Transactions on Knowledge and Data Engineering. 17(6), 820-833 (2005)
5. Ioup, E., Shaw, K., Sample, J., et al.: Efficient AKNN spatial network queries using the M-tree. In: 15th International Symposium on Advances in Geographic Information Systems, Article 46, Seattle, Washington, USA (2007).
6. Xie, X., Jin, P., Yiu, M., et al., Enabling scalable geographic service sharing with weighted imprecise voronoi cells, IEEE Transactions on Knowledge and Data Engineering 28(2), 439–453 (2016).
7. Li, H., Lu, H., Huang, B., et al.: Two ellipse-based pruning methods for group nearest neighbor queries. In: 13th Annual ACM International Symposium on Advances in Geographic Information Systems, pp. 192–199, Bremen, Germany (2005)
8. Li, F., Yao, B., Kumar, P.: Group enclosing queries. IEEE Transactions on Knowledge and Data Engineering 23(10), 1526–1540 (2011).
9. Li, Y., Li, F., Yi, K., et al.: Flexible aggregate similarity search. In: 2011 International Conference on Management of Data, pp. 1009–1020, ACM, Athens, Greece (2011).
10. Yan, D., Zhao, Z., Ng, W.: Efficient processing of optimal meeting point queries in Euclidean space and road networks. Knowledge and Information Systems 42(2), 319–351 (2015).
11. Yiu, M. L., Dai, X., Mamoulis, N., et al.: Top-k spatial preference queries. In: 23rd International Conference on Data Engineering, pp. 1076–1085. IEEE, Istanbul, Turkey (2007).
12. Gao, Y., Wang, Y., Yi, S.: Preference-Aware Top-k Spatio-Textual Queries. In: Song, S., Tong, Y. (eds.) WAIM 2016, LNCS, vol. 9998, pp. 186–197, Springer, Cham (2016).
13. Ro cha-Junior, J. B., Gkorgkas, O., et al.: Efficient processing of Top-k spatial keyword queries. In: Pfoser, D., et al. (eds.) SSTD 2011, LNCS, vol. 6849, pp. 205–222, Springer, Berlin, Heidelberg (2011).
14. Cho, H., Kwon, S., Chung, T.: ALPS: An efficient algorithm for top-k spatial preference search in road networks. Knowledge and Information Systems 42(3), 599–631 (2015)
15. Attique, M., Cho, H. J., Jin, R., et al.: Top-k Spatial Preference Queries in Directed Road Networks. ISPRS Journal of Geo-Information 5(10), 170 (2016)
16. Rocha-Junior, J.B., Vlachou, A., Doulkeridis, C., Nørv˚ag, K.: Efficient processing of top-k spatial preference queries. Proceedings of VLDB Endowment 4(2), 93–104 (2010)
17. Li, M., Chen, L., Cong, G., et al.: Efficient Processing of Location-Aware Group Preference Queries. In: 25th International on Conference on Information and Knowledge Management, pp. 559–568, ACM, Indianapolis, Indiana, USA (2016).