Exact and Heuristic Algorithms for Some Spatial-aware Interest Group Query Problems

Chih-Yang Huang¹, Po-Chuan Chien², Yen Hung Chen³

¹Department of Computer Science and Information Engineering, National Taiwan University, Taiwan
²Research Center for Information Technology Innovation, Academia Sinica, Taiwan
³Department of Computer Science, University of Taipei, Taiwan

user@hcy.idv.tw, dibery@citi.sinica.edu.tw, yhchen@utaipei.edu.tw

Abstract

Location-based social networks are important issues in the recent decade. In modern social networks, such as Twitter, Facebook, and Plurk, attempts to get the accurate address positions from their users, and try to reduce the gap between virtuality and reality. This paper mainly aims at both the interests of Internet users and their real positions. This issue is called the spatial-aware interest group query problem (SIGQP). Given a user set $U$ with $n$ users, a keyword set $W$ with $m$ words, and a spatial object set $S$ with $s$ items, each of which contains one or multiple keywords. If a user checks in a certain spatial object, it means the user could be interested in that part of keywords, which is countable to clarify the interests of the user. The SIGQP then tries to find a $k$-user set $U_k$, $k \leq n$, such that the union of keywords of these $k$ users will equal to $W$, and additionally, the diameter (longest Euclidean distance of two arbitrary users in $U_k$) should be as small as possible. The SIGQP has been proved as NP-Complete, and two heuristic algorithms have been proposed. Extended from SIGQP, another problem is in finding the smallest $k$ for $U_k$ to cover all the keywords, with the users’ distance as the secondary criterion, called as “minimum user spatial-aware interest group query problem” (MUSIGQP). This paper designs a branch & bound method and a measure & conquer method to solve SIGQP and MUSIGQP respectively, and a performance analysis is given.

Keywords: Spatial-aware interest group query problem, Exact algorithms, Branch & bound, Measure & conquer, Location-based service

1 Introduction

Mobile phone positioning and social network have been seen as significant and innovative issues in the recent decade [1]. The goal of group queries in location-based social networks is to find a group of users where members are close to each other and have the same interests [2-5].

Li et al. proposed the spatial-aware interest group query problem (SIGQP) in order to combine the issues of positioning and group queries in location-based social networks [6]. Let $U = \{u_1, u_2, ..., u_n\}$ be the user set with $n$ users, $S = \{s_1, s_2, ..., s_n\}$ be the spatial object set of $s$ spatial objects, and $W = \{w_1, w_2, ..., w_m\}$ be the keyword set of $m$ query keywords. Each spatial object contains one or more keywords. It is feasible to count the value of interests of a user if it checks in some spatial objects. The normalized interest value of user $u$ over keyword $w$ is indicated as $I(u, w)$, which is illustrated in Equation 1, where $\text{count}(u, p)$ is the total time of the user $u$ checking in the spatial object $p$, and the collection of $i$ spatial objects that user $u$ has been is denoted as $S_u = \{p'_1, p'_2, ..., p'_i\}$.

$$I(u,W) = \frac{\text{count}(u,w)}{\sum_{p \in S_u} \text{count}(u,p)}$$  (1)

We then furthermore consider a more general case where a user is interested in multiple keywords, collected as a set $W'$. The interest score $I(u, W')$ over that keyword group is evaluated as the aggregation of the interest values of all keywords in $W'$ (i.e., $\sum_{w \in W'} I(u, w)$). A spatial-aware interest group (SIG) query is defined to find a user set $U_k$ with $k$ users, $k \leq n$, to cover all $m$ keywords. Furthermore, an adjustment $\alpha$ is introduced so that we can adjust the weight between the interest and the distance of the users, and the maximization objective function (where the maximization target is the Rank of the selected $U_k$) is defined in Equation 2 [6].

$$\text{Rank}_\alpha(U_k) = \alpha \cdot \min \{I(u,W) | u \in U_k\} + (1-\alpha)(D(U)-D(U_k))$$  (2).

$D(U)$ and $D(U_k)$ are the diameters of the user sets $U$ and $U_k$, which is the farthest Euclidean distance between any two users in that set, and can be calculated by

$$\max_{i,j \in U_k} \sqrt{(x_i-x_j)^2 + (y_i-y_j)^2}.$$
When $\alpha = 1$, the distance of users would be irrelevant and using the fewest group of users to cover all the keywords will be the main issue. We call this problem as a minimum user spatial-aware interest group query problem (MUSIGQP) and highlight on the keywords coverage over the users’ distance. In this case, the minimum set cover (MSC) method, an exact algorithm, can be applied, and we can derive an exact answer toward MUSIGQP. Because MSC uses the least number of sets to cover all keywords, the result of MUSIGQP will consist of as least groups as possible. However, this would totally ignore the factor of the distances, so in later sections, we will show how to deal with the distances between users by integrating the concept of distance into the MSC algorithm.

Our contribution is as follows: We provide a branch & bound method to solve SIGQP and analyze it. To our knowledge, there is no MUSIGQP research for the design of non-trivial brute force exact algorithms, and hence we provide a precise solution to the problem.

In real-world applications, every check-in place could be seen as a spatial object $p$ in SIGQP (MUSIGQP), and there are some features or keywords for those places. For example, Starbucks is representative of the words: “food”, “coffee”, or “drink”, while VieShow Cinemas is the symbol of the words: “movie” and “popcorn”. Also, every word could be seen as a keyword in SIGQP (MUSIGQP). If a user checks in a Starbucks coffee shop, it means the user would be interested in words (keywords) like “coffee” or “drink”. Our algorithms on SIGQP (MUSIGQP) consider not only the keywords but also the users’ distances. It finds users having common interests and staying in near spots. Broadcasting the advertisements and marketing are possible applications for SIGQP (MUSIGQP).

2 Related Work

With the development of location-aware devices, ubiquitous Internet, and the techniques of social computing, access to the position as well as social information of a user are more easily available. A lot of researches [2-9] are aiming at finding a group of users who are close and are interested in the common issues, their distances, and their skills, and their distances are not far.

If we consider spatial query processing, R-tree and R’ tree are available. In the recent 30 years, there are a lot of queries we can use, such as $k$-nearest-neighbor queries [10-14], range queries [15-16], and closest-pair queries [17-19]. There are also some researches [20-25] combining spatial query processing and keywords. In addition, some researches propose group and team query on social network [2-3, 7-8, 26-27]. Li and the others identify Rank$_d(U_d)$ and SIGQP [6, 28-30] and develop two heuristic algorithms, based on the technique of IR-tree [20]. They reduce the set cover problem into a SIGQP and prove it as NP-Complete [31]. However, SIGQP is similar to maximum coverage problem [32-33]. It can be solved by $(1 - 1/e)$-approximation algorithm [32] $(e \approx 2.718)$ through Greedy skills. Its lower bound of approximation rate is $(1 - 1/e) + o(1) \approx 0.632$ [32-34].

3 Problem Formulation & Solution Approach

In a SIGQP, we consider a situation where a user set $U = \{u_1, u_2, ..., u_n\}$, a spatial object set $S = \{p_1, p_2, ..., p_m\}$, and a query keyword set $W = \{w_1, w_2, ..., w_p\}$ exist, and satisfy $\forall u \in U$, $\exists p \in S \land p \in S_u$ ($S_u$ is places that user $u$ has been to, and $S_u$ is not empty, see Table 1.) and $\forall p \in S$, $\exists w \in W \land w \in W_p$, from which we can infer that an element $u \in U$ is factually a set of keywords $w \in W$, and that every user in the user set will be inherently associated with a group of keywords using the transitive law.

### Table 1. Symbols shown in this paper

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U$</td>
<td>A user set with $n$ users with a keyword set and a spatial object set attributes, which is ${u_1, u_2, ..., u_n}$.</td>
</tr>
<tr>
<td>$U_u$</td>
<td>The union of all keywords that are bound to any spatial object in $S_u$ for user $u$ (see below).</td>
</tr>
<tr>
<td>$W$</td>
<td>A keyword set with $m$ words, which is ${w_1, w_2, ..., w_m}$.</td>
</tr>
<tr>
<td>$W_p$</td>
<td>The collection of words that spatial object $p$ has, which is ${W_{p_1}, W_{p_2}, ..., W_{p_m}}$.</td>
</tr>
<tr>
<td>$S$</td>
<td>A spatial object set with $s$ items, which is ${p_1, p_2, ..., p_s}$.</td>
</tr>
<tr>
<td>$S_u$</td>
<td>The collection of all $i$ spatial objects that user $u$ has been, which is ${p_{u_1}, p_{u_2}, ..., p_{u_m}}$.</td>
</tr>
<tr>
<td>$D(U)$</td>
<td>The farthest Euclidean distance between two arbitrary users in the user set $U$.</td>
</tr>
<tr>
<td>$I(u, w)$</td>
<td>Interest score of user $u$ over the keyword (set) $w$.</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>Weight adjustment coefficient between the distance and the weight.</td>
</tr>
</tbody>
</table>

Since the goal of a MUSIGQP is to select a user set $U'' \subseteq U$ such that all keywords in $W$ are covered, we can, without loss of generality, eliminate the spatial object group $S$ from the problem and construct the direct relationship between a user and a keyword as a bipartite graph. Because our goal is still to select a $U''$ such that the union of keywords from all users in $U''$ will be equal to $W$, we can restate the MUSIGQP as a minimum set cover (MSC) problem, by regarding $W$ as the universe set, and $U$ as the collection of sets. The **user set** is denoted as a group of users and each of user has its position as well as part of keywords.

In the following subsections, we will be
concentrated on the branch & bound method for SIGQP as well as the exact solution approach of the minimum set cover problem on MUSIGQP. The symbols used in our problems definition are organized in Table 1.

### 3.1 Branch & Bound Method

We design a heuristic algorithm to solve SIGQP. The $Rank_u$ in Figure 1 is derived from Equation 2. We modify Equation 2 and present it as below for the branch & bound method.

1. user_set BranchAndBound (user_set $U$) {
   2.   Initial:
   3.   user_set the_goal_set // the goal set
   4.   double ratio // set the bound to stop iteration
   5.   int $k$ // find $k$ users and put them into the goal set
   6.   for each user set in $U$ {
   7.     Find a user with the most keywords and put it in the
   8.     } // base case
   9.   while ($|the\_goal\_set| < k$) {
   10.    if ($|the\_goal\_set| = 0$) return $\phi$. // base case
   11.    Find a user with high rank and add it to the_goal_set
   12.    if ($|the\_goal\_set| > k \times ratio$) {
   13.       Find the rank of the remaining users.
   14.       Put them into the goal set based on the rank until
   15.       [the_goal_set] = $k$.
   16.   }
   17.   if ($|the\_goal\_set| > k \times ratio$) {
   18.       Find the rank of the remaining users.
   19.       Put them into the goal set.
   20.   return the_goal_set.
   21.   }

**Figure 1.** Branch and bound method for SIGQP

$$Rank_u(u_i, u_j) = \alpha\times\min\{|u_i|, |u_j|\} + (1-\alpha)\times(D(U)-D(u_i, u_j))$$

There is a coefficient, $\alpha$, set as 0.5. $D(U)$ means the farthest distance between any two users in the set $U$ as Figure 5. $D(u_i, u_j)$ means the distance between the users $u_i$ and $u_j$. Based on Figure 1, we can find the approximation set for SIGQP. The performance will be discussed later.

### 3.2 Measure & Conquer Exact Algorithm for MSC with Consideration of Distance

Fomin, Grandoni, and Kratsch proposed an algorithm to exactly solve the MSC problem by finding the minimum number of sets required to cover all the keywords [35]. However, such number does not help when we wish to find out which sets are chosen. Therefore, we modify this algorithm, which is illustrated in Figure 2, so that the algorithm returns the actually selected sets.

For a user set $U$, when each user $u$ in $U$ covers at most 2 keywords in $W$, finding the minimum user to cover all keywords, denoted by $2\_msc(U)$, can be solved in polynomial time by the graph matching algorithm [31, 35-36]. Additionally, during the recursion process, the notation of $U\{u\}$ means that we remove user $u$ from the user universe set $U$ to the next recursion step as if it never exists (i.e., we do not choose $u$ in the final result), and the notation of $del(u, U)$ means that we select user $u$ in the final result, and therefore all keywords of $u$ are marked as found (and $u$ is eliminated) when the recursion steps down.

### 3.3 Measure & Conquer Exact Algorithm for MSC with Consideration of Distance

When there are multiple combinations of groups that can satisfy the criteria that every keyword is selected, we would show an inclination to choose the group whose diameter, calculated by the max distance between user ($i, j$) in the group, is smaller. The standard method described in the above subsection does not, however, take the Euclidean distance between users into consideration; consequently, such a method may not be able to derive the desired solution.

To handle such situations, we will need to take the distance between users into account. The modified version, denoted as exact_MUSIGQP later in this paper, is illustrated in Figure 3.

1. user_set exact_MUSIGQP (user_set $U$) {
   2.   if ($|U| = 0$) return $\phi$. // base case
   3.   Find the rank of the remaining users.
   4.   if ($|\{u\}| = 2$) // this user has only 2 keywords
   5.       return $2\_msc(U)$. // $2\_msc$ is a kind of Graph
   6.       return $\min\{msc(U\{u\}), u \cup msc(del(u, U))\}$. // recursion, try with $u$ either selected or not selected
   7.   return $\min\{msc(U\{u\}), u \cup msc(del(u, U))\}$. // recursion, try with $u$ either selected or not selected
   8.   }
In the last line, we change the min to $\text{min}_\text{dist}$, which will evaluate the set size first, and when the size of the two sets are equal, the group who has a smaller diameter will be returned. The algorithm of $\text{min}_\text{dist}$ is presented in Figure 4, description of the $\text{diameter}$ function is given in Figure 5.

1. `user_set min_dist(user_set grpA, user_set grpB) {`
2. `if |grpA| < |grpB|`
3. `return grpA;`
4. `else if |grpA| > |grpB|`
5. `return grpB;`
6. `else if diameter(grpA) < diameter(grpB) // size equal`
7. `return grpA;`
8. `return grpB. // the last possibility`
9. `}`

**Figure 4.** Choose the set with a smaller size and a smaller diameter

1. `diameter(user_set grp) {`
2. `d = 0 // the diameter`
3. `for each element pair (A, B) in grp`
4. `if Euclidean_distance(A, B) > d`
5. `d = Euclidean_distance(A, B).`
6. `return d;`
7. `}`

**Figure 5.** Measure the diameter of a group

Regarding the correct solution set as $\text{Opt}$, we can say that a user set $u$ may or may not be in $\text{Opt}$. If $u \in \text{Opt}$, then the solution is then further found by selecting $u$ and remove all keywords in $u$ from other existent users; otherwise, the solution is found by eliminating $u$ from the user set. In this way, we are capable of traversing all possible combination of groups, and because in each step of the recursion, we choose the one with a smaller diameter, in the global scope, we can achieve the optimal solution.

In the next section, we will mainly utilize this method to solve MUSIGQP.

4 Performance Evaluation

All algorithms are implemented in Java programming language. The models of the CPU and RAM are Intel Core i7 2.2 GHz and 16 GB 1600 MHz DDR3, respectively.

4.1 Branch & Bound Method

To test the performance of the branch & bound method, we test it on a situation where 100 users, from which we will select 20 as the solution, exist in the square within $(0, 0)$ and $(10, 10)$ and calculate the similarity with the greedy solution. In addition, we set the size of keywords to 50, and $\alpha$ to 0.5. The experimental result is plotted in Figure 6, with each ratio conducted 100 times and take the average.

It can be easily inferred that by the increase in the branch ratio, we can get better results, as the trend shown in the figure. In addition, we may observe that the similarity has reached 0.769 when the branch ratio is 0.1, and that there is an outstanding breakthrough when the branch ratio has reached 0.5, both of which imply that we can reduce the calculation load while still obtaining favorable results, and one can choose the most suitable ratio, depending on the trade-off between accuracy and time. We also plot the execution time of the branch & bound method in logarithm base 2 as Figure 7.

**Figure 6.** The similarity trend between the branch & bound method and the greedy algorithm

**Figure 7.** Running time versus the value of $k$ users in heuristic algorithm

4.2 Measure & Conquer Exact Algorithm for MSC with Consideration of Distance

To each of our algorithms, an experiment is conducted. We test them on a situation where $k$ users, whose value within the interval $[10, 170]$, $k/2$ keywords(candidates), and all $k$ users will stand in the position where $0 \leq x \leq k$, $0 \leq y \leq k$ in Euclidean space.
Each user has a random group of keywords and a random position. The experimental result is plotted in Figure 8, with every $k$ users conducted 100 times and take the average. Owing to the fact that the time complexity of this algorithm is in exponential time, we represent the logarithm of running time in base 2 as Figure 8.

![Figure 8. Running time versus the value of $k$ users in exact algorithm](image)

Figure 8. Running time versus the value of $k$ users in exact algorithm

It is inferred that the transformation of MUSIGQP into MSC algorithm is feasible. The time complexity of the designed algorithm would be $O(2^{0.465n})$ upper bound. On the other hand, $\Omega(2^{0.396n})$ and $\Omega(2^{0.142n})$ would be the possible lower bounds. This theorem is on the ground of a measure & conquer approach for the analysis of exact algorithms described in [35]. Thus, we could decrease the time complexity of MUSIGQP and transform MUSIGQP into MSC.

5 Conclusion

In this paper, we investigated the spatial-aware interest group query problem and provide two algorithms on SIGQP and MUSIGQP, with their performance being analyzed. One is the branch & bound method and the other is the measure & conquer exact algorithm.

The branch & bound method is a heuristic algorithm whose time complexity is $O(n^3)$. Although this solution provides, rather than the correct answer, only an approximation, it is capable of reducing the time complexity to polynomial time, which is much affordable even when the size scales out.

The other algorithm is based on the MSC algorithm proposed in [35]. We transform SIGQP into MSC. The first trial exact algorithm (Measure & Conquer Exact Algorithm for MSC) doesn’t take the Euclidean space into consideration, while the second one (Measure & Conquer Exact Algorithm for MSC with Consideration of Distance) is concerned about the distance. Both of them take $O(2^{0.465n})$ upper bounds, which is lower than $O(2^n)$, the time complexity of a brute force approach.

All in all, we provide these algorithms and try to decrease the time complexity when SIGQP is proved as NP-Complete.

Acknowledgments

This research was supported by the Ministry of Science and Technology in Taiwan under the grants MOST 107-2813-C-845-025-E and 109 new-type information security practice demonstration course program by Ministry of Education, Taiwan. Corresponding author: Yen Hung Chen, Email: yhchen@utaipei.edu.tw.

References

[12] M. Kolahdouzan, C. Shahabi, Voronoi-Based k Nearest


**Biographies**

**Chih-Yang Huang** received the B.S. degree on Department of Computer Science, University of Taipei in 2019. He is studying for his master’s degree on Department of Computer Science and Information Engineering, National Taiwan University. His research interests include Algorithms, Natural Language Processing, Computational Linguistics, Social Network Analysis and Digital Humanities.

**Po-Chuan Chien** received the B.S. degree on Department of Accounting and Department of Computer Science, National Chengchi University in 2015 and the M.S. degree on Department of Information Management, National Taiwan University in 2017. His
research interests include financial text analysis and algorithm design.

Yen Hung Chen received his Ph.D. degree on Department of Computer Science from National Tsing Hua University in 2006. He is currently serving as an Associate Professor of Department of Computer Science from University of Taipei. His research interests include algorithms, computer theory, graph theory and operations research.