Xu YX, Chen W, Xu JJ *et al.* Discovering functional organized point of interest groups for spatial keyword recommendation. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 33(4): 697–710 July 2018. DOI 10.1007/s11390-018-1850-3

Discovering Functional Organized Point of Interest Groups for Spatial Keyword Recommendation

Yan-Xia Xu, Wei Chen, Jia-Jie Xu, *Member*, *CCF*, *ACM*, Zhi-Xu Li, *Member*, *CCF*, *ACM*
Guan-Feng Liu, *Member*, *CCF*, *ACM*, and Lei Zhao*, *Member*, *CCF*, *ACM*

School of Computer Science and Technology, Soochow University, Suzhou 215006, China

E-mail: {xyx.edu, wchzhg}@gmail.com; {xujj, zhixuli, gfliu, zhaol}@suda.edu.cn

Received December 27, 2017; revised May 7, 2018.

Abstract A point of interest (POI) is a specific point location that someone may find useful. With the development of urban modernization, a large number of functional organized POI groups (FOPGs), such as shopping malls, electronic malls, and snacks streets, are springing up in the city. They have a great influence on people's lives. We aim to discover functional organized POI groups for spatial keyword recommendation because FOPGs-based recommendation is superior to POIs-based recommendation in efficiency and flexibility. To discover FOPGs, we design clustering algorithms to obtain organized POI groups (OPGs) and utilize OPGs-LDA (Latent Dirichlet Allocation) model to reveal functions of OPGs for further recommendation. To the best of our knowledge, we are the first to study functional organized POI groups which have important applications in urban planning and social marketing.

Keywords functional organized point of interest (POI) group, POI clustering, OPG-LDA (organized point of interest group-latent Dirichlet allocation) model, spatial keyword recommendation

1 Introduction

A point of interest (POI) is a uniquely identified specific site[1]. POIs, such as hotels, restaurants, are very fundamental factors in recommendation. A lot of work has been done to investigate POIs-based recommendation[2-7]. However, POIs-based recommendation has its limitations.

1) Due to the diversity of requirements, POIs-based recommendation is not applicable in some cases. For example, if a lady wants to buy clothes, POIs-based recommendation usually recommends the best shop for the lady. Cao *et al.*[3] recommended POIs based on spatial distance. Chen *et al.*[5] recommended POIs on the basis of the spatial distance and rating information. Guo *et al.*[8] recommended POIs based on users' budgets. All of the work mentioned above just takes a few requirements of users into consideration. However, the lady may have a lot of other requirements on the styles, brands, materials, sense of comfort and so on.

2) POIs-based recommendation does not consider potential query keywords of users. For example, a user probably wants to buy some accessories after buying a laptop, such as a blue-tooth mouse, a megaphone, and a printer. Recommending a computer store is far from enough in this situation because the buyer probably has to go far away for a mouse, megaphone and printer.

3) POIs-based recommendation is computationally expensive. A lot of POIs-based recommendation proves to be NP-hard[9-11].

In order to solve these issues, we propose the concept of functional organized POI group (FOPG) which contains a set of close POIs, such as shopping malls and electronic malls. If a lady wants to buy clothes, a shopping center can be recommended so that she can shop around before buying. If a man wants to buy a

Regular Paper

Special Section on Recommender Systems with Big Data

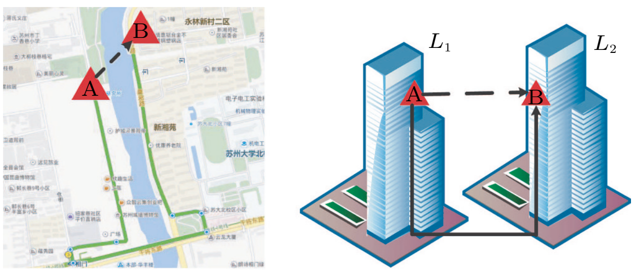
This work was supported by the National Natural Science Foundation of China under Grant Nos. 61572335, 61472263, 61402312 and 61402313, the Natural Science Foundation of Jiangsu Province of China under Grant No. BK20151223, and the Collaborative Innovation Center of Novel Software Technology and Industrialization, Jiangsu, China.

*Corresponding Author

laptop, an electronic mall can be recommended where he can buy the laptop and some accessories simultaneously. Besides, FOPGs-based recommendation is more efficient than POIs-based recommendation because the number of FOPGs is much smaller than that of POIs.

To mine FOPGs from POIs datasets, we are facing several challenges.

1) It is difficult to measure the real distance between two POIs. Let us take Fig.1(a) as an example. Given two POIs, i.e., A and B, the dotted line denotes the spatial distance between two POIs and the solid line is a path returned by BaiduMap. It is obvious that the real distance is much longer than the spatial distance between A and B because there is a river between them. Besides, hierarchical distance can lead to the result that the spatial distance is much shorter than the real distance. For example, in Fig.1(b), A is in the 15th floor of building L_1, B is in the 15th floor of building L_2, the dotted line represents the distance measured in the traditional maps, and the solid line denotes the real distance for a user. To narrow the gap between the spatial distance and the real distance, we need to consider both coordinates and the addresses of POIs when computing the distance between two POIs. Two POIs in Fig.1(a) are located in different streets and two POIs in Fig.1(b) are located in different buildings. Their addresses vary from one another.



(a) (b)

Fig.1. Spatial distance vs real distance.

2) Semantic drift is another challenge. For example, if there are three POIs, A, B and C close together. A's street address is "abcdef", B's street address is "abcdkm", and C's street address is "hicdkm". Due to the similarity to some degree, A and B are put into one FOPG, while B and C are put into another FOPG. Finally, A and C are in the same FOPG although they have different street addresses. To solve this issue, we propose a grid-based method to reduce semantic drift.

3) Efficiency is the third challenge. This paper utilizes density-based clustering algorithm, i.e., DBSCAN[12] to cluster POIs because the number of FOPGs is unknown and the shapes of FOPGs are arbitrary. Gan and Tao[13] claimed that DBSCAN actually requires $O(n^2)$ time. Besides, when clustering POIs, it is necessary to compute addresses similarity which is not efficient[14]. To improve the efficiency, three pruning rules are proposed. Details are illustrated in Section 4.

As far as we know, we are the first to study functional organized POI groups. Regions of POIs have been explored in previous work. Yuan *et al.*[15] segmented a city into disjoint regions according to major roads. Feng *et al.*[16] found the best region which the sub-modular monotone aggregate score of the spatial objects inside is maximized. The methods in previous work cannot be used for discovering FOPGs because of several limitations.

1) Most of defined regions in previous work have fixed sizes and shapes, while the size and shape of an FOPG are usually arbitrary.

2) In an FOPG, the spatial distance and real distance between POIs are both short. The previous work only considers the spatial distance between POIs and ignores the real distance between POIs, which results in putting irrelevant POIs into FOPGs.

In general, the contributions in this paper are summarized as follows.

- POIs-based recommendation cannot satisfy users' various and variable requirements. To solve this issue, this paper proposes the concept of functional organized POI groups (FOPGs) for recommendation. FOPGs-based recommendation can meet users' demands better than POIs-based recommendation.
- To solve semantic drift issue in discovering FOPGs, this paper propose a novel approach, i.e., STC-DG. To improve the efficiency of STC-DG, we design three pruning rules.
- We carry out extensive experiments based on two real-life datasets to compare POIs-based recommendation with FOPGs-based recommendation. Besides, we evaluate effectiveness and efficiency of the method for discovering FOPGs. The experiments demonstrate that FOPGs-based recommendation is better than POIs-based recommendation and the method for discovering FOPGs has a good performance.

The rest of the paper is organized as follows. Section 2 reviews related work. In Section 3, we define the problem formally. Section 4 and Section 5 describe the method for discovering FOPGs. In Section 6, we show a series of experiments. Finally, Section 7 concludes the paper in brief.

2 Related Work

Our work is related to various topics including urban computing, region search, spatial clustering, spatial keyword query, and location-based recommendation.

2.1 Urban Computing

The increasing availability of large-scale and real-world datasets contributes to investigating about urban computing. Yin *et al.*[17] modelled location-based user rating profiles to produce high-quality recommendations. In addition, Yuan *et al.*[15] used both human mobility and points of interests to discover regions of different functions. Zhu *et al.*[18] helped businesses promote their locations by advertising wisely through the underlying location based social networks (LBSNs). Yin *et al.* inferred users' social communities by incorporating spatio-temporal data and semantic information[19]. Tong *et al.* investigated crowdsourcing task decomposition and allocation[20-21]. Different from the previous work, our work aims to find FOPGs which are useful for spatial keyword recommendation.

2.2 Region Search

Liu *et al.*[22] proposed a new problem of finding subject oriented top-k non-overlapping hot regions. Choi *et al.*[23] focused on solving the maximizing range sum problem in spatial databases. Cao *et al.*[24] retrieved regions connected by road segments. All of them defined regions as width- and height-fixed rectangles or radius-fixed circles which are contrary to the fact that regions are often arbitrarily shaped in reality. Besides, they did not take the real distance between two POIs into consideration which results in putting irrelevant POIs into FOPGs.

2.3 Spatial Clustering

Han and Kamber[25] summarized clustering algorithms, including partitioning clustering methods, hierarchical clustering methods, density-based clustering methods, grid-based clustering methods, etc. Density-based methods are employed in this paper because they can discover clusters of arbitrary shapes. Besides, density-based clustering methods can be extended from full space to subspace clustering. Grid-based methods quantize the whole space into a number of cells. The processing time does not depend on the dataset size, but on the number of grids. Besides, grid-based clustering methods not only have high efficiency, but also can be integrated with other clustering methods.

2.4 Spatial Keyword Query

Chen *et al.*[26] summarized spatial keyword query which is one of the most important queries in spatial textual searching. Given a location and a set of keywords, the spatial keyword query aims to find a single object that is close to the location and covers the set of keywords. Chen *et al.*[26] divided geo-textual indices into three types: spatial indexing scheme, text indexing scheme, and combination scheme. Besides, there are many important variants of spatial keyword query. Some work aimed to discover groups of objects that collectively meet the keywords[3,27]. Maria *et al.*[28] took a region and a set of keywords as inputs and found objects which are in that region and cover the keywords. All the work mentioned above offers users POIs. However, POIs-based recommendation hardly satisfies users' various and variable requirements. Besides, POIs-based recommendation is computationally expensive.

2.5 Location-Based Recommendation

An increasing number of location-based social networks (LBSNs) contribute to deep investigations on location-based recommendation[29-34]. To improve the location-based recommendation, a lot of researchers have made great efforts. Xie *et al.* proposed a novel method for dynamic user preferences modeling based on the learnt embedding of POIs[33]. Zhang *et al.* exploited the sequential influence of locations on users' check-in behaviors for location recommendations[29]. Gao *et al.* introduced a novel location recommendation framework based on the temporal properties of user movement[30]. Wang *et al.* proposed a geographical sparse additive generative model for spatial item recommendation and the model considers both user personal interest and the preference of the crowd in the target region[31]. Yin *et al.* proposed a unified probabilistic generative model to jointly model spatial, temporal and semantic effect[34]. All of the work mentioned above recommends POIs for users. In this paper, we propose FOPGs to meet the various requirements from users. For example, if a lady wants to go shopping, one shop is not enough for the lady because she needs to buy shoes, clothes, jewelry, perfume, etc. The lady prefers a shopping center (FOPG) rather than a shop (POI).

3 Problem Statement

In this section, we present problem statement and give related definitions. At first, we show the notations used throughout the paper in Table 1.

Table 1. Definitions of Notations

Notation	Definition
w_s	Spatial similarity
φ	Size of a cluster
w_a	Address similarity
g	Grid granularity
c_i	A cluster
t	Tag of POIs
a	Address of POIs
f	Function of OPGs
d_s	Spatial distance

Given a set of POIs $\mathcal{O} = \{o_1, o_2, o_3, \ ..., o_{|\mathcal{O}|}\}$, each $o_i \epsilon \mathcal{O}$ is in the form of $(o_i.x, o_i.y, o_i.a, o_i.t)$ where $o_i.x$ is latitude, $o_i.y$ denotes longitude, $o_i.a$ represents street address, and $o_i.t$ represents a set of tags. We aim to find a set of clusters $\mathcal{C} = \{c_1, c_2, c_3, ..., c_{|\mathcal{C}|}\}$ where c_i is an FOPG consisting of POIs and functions.

Definition 1 (Spatial Similarity). *Given a set of POIs $\mathcal{O} = \{o_1, o_2, o_3, ..., o_{|\mathcal{O}|}\}$, for any objects o_i and o_j, the spatial distance is*

$$d_s(o_i, o_j) = |o_i.x - o_j.x| + |o_i.y - o_j.y|,$$

and then spatial similarity between o_i and o_j is defined as

$$w_s(o_i, o_j) = \frac{D - d_s(o_i, o_j)}{D},$$

where $D = \max_{o_m \in \mathcal{O}, o_n \in \mathcal{O}} d_s(o_m, o_n)$ and $0 \leqslant w_s(o_i, o_j) \leqslant 1$.

Definition 2 (Address/Textual Similarity). *Given two POIs o_i and o_j, $\mathcal{T}_{ij}$ denotes the longest common subsequence*[35] *between $o_i.a$ and $o_j.a$. There are three metrics for textual similarity including Jaccard, Cosine and Dice, defined in* (1), (2) *and* (3) *respectively.*

$$Jaccard: \ w_a(o_i, o_j) = \frac{|\mathcal{T}_{ij}|}{|o_i.a| + |o_j.a| - |\mathcal{T}_{ij}|}, \quad (1)$$

$$Cosine: \ w_a(o_i, o_j) = \frac{|\mathcal{T}_{ij}|}{\sqrt{|o_i.a| \times |o_j.a|}}, \quad (2)$$

$$Dice: \ w_a(o_i, o_j) = \frac{2 \times |\mathcal{T}_{ij}|}{|o_i.a| + |o_j.a|}, \quad (3)$$

where $w_a(o_i, o_j)$ denotes the textual similarity between o_i and o_j.

Definition 3 (Organized POI Group (OPG)). *Given a set of POIs $\mathcal{O}$, $\mathcal{P}$ is a sub-set of $\mathcal{O}$. POIs in $\mathcal{P}$ are close and similar in addresses. $\mathcal{P}$ is a complete OPG only if $|\mathcal{P}|$ is larger than $\widehat{\varphi}$ and $\forall o_i \in \mathcal{P}$, $\nexists o_j \in \mathcal{O} - \mathcal{P}$ satisfies all inequalities as follows.*

$$w_s(o_i, o_j) \geqslant \widehat{w_s},$$
$$w_a(o_i, o_j) \geqslant \widehat{w_a},$$

where $\widehat{\varphi}$ is the threshold of cluster size, $\widehat{w_s}$ is the threshold of spatial similarity, and $\widehat{w_a}$ is the threshold of textual similarity.

Definition 4 (Functional Organized POI Group (FOPG)). *An FOPG has not only a set of POIs but also functions such as $(P_{f_1}, P_{f_2}, \cdots, P_{f_n})$ where P_{f_i} is the probabilistic of the i-th function. Each function is in the form of $(P_{t_1}, P_{t_2}, \cdots, P_{t_m})$ where P_{t_i} is the probabilistic of the i-th tag.*

4 Discovering OPGs

We have shown the definition of functional organized POI groups (FOPGs) which can satisfy users' various and variable requirements. To discover FOPGs, there exist two steps. This section mainly describes the first step, i.e., discovering organized POI groups (OPGs) by clustering POIs. Two algorithms, i.e., STC-D and STC-DG, are proposed for the first step. Besides, three pruning rules are proposed to improve the efficiency of STC-DG.

4.1 Algorithm STC-D

Algorithm STC-D (Spatio-Textual Clustering Based on Density) is a variant of the DBSCAN[12] algorithm. STC-D replaces spatial distance with spatial similarity and street address similarity. Besides, STC-D utilizes dynamic grid partitioning[36] to avoid traversing the whole data space to find neighbors of a point. Let us look at Fig.2. The granularity of the grid is set to $(1 - \widehat{w_s}) \times D$. To find neighbors of a point p which is in the cell l, we only consider points inside cell l and its eight adjacent cells. Points in the other cells cannot be neighbors of p because if a point q is not in cells 1~9, the spatial distance between p and q must be longer than $(1 - \widehat{w_s}) \times D$. Then, the spatial similarity between p and q is smaller than the threshold of spatial similarity $\widehat{w_s}$.

Algorithm 1 is the pseudo-code of STC-D. Queue *que* stores all potential core points of a cluster. c_i stores all points of the i-th cluster, including core points and border points (refer to the previous work [12] for definitions of core points and border points). The algorithm consists of three parts. Firstly, it selects an unclustered point o_k as a potential core point (lines 1∼3). Secondly, it repeats lines 5∼16 until there is no potential core point in *que*. When *que* is empty, a complete cluster c_i is obtained. Thirdly, if the size of cluster c_i is larger than the threshold of cluster size $\widehat{\varphi}$, c_i is an OPG (lines 18 and 19). The second part is the most important part in the algorithm. At first, it selects a potential core point o_j (line 5). Then, it searches the neighbors of o_j by using dynamic grid partitioning (line 6). If the number of neighbors is larger than $\widehat{\varphi}$, o_j becomes a core point (lines 7 and 8) and all neighbors of o_j become potential core points (lines 9∼13). Otherwise, o_j becomes a border point (line 15).

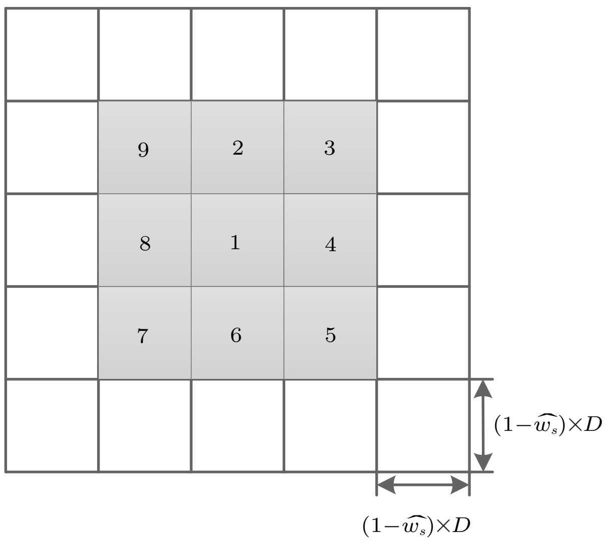


Fig.2. Dynamic grid partitioning.

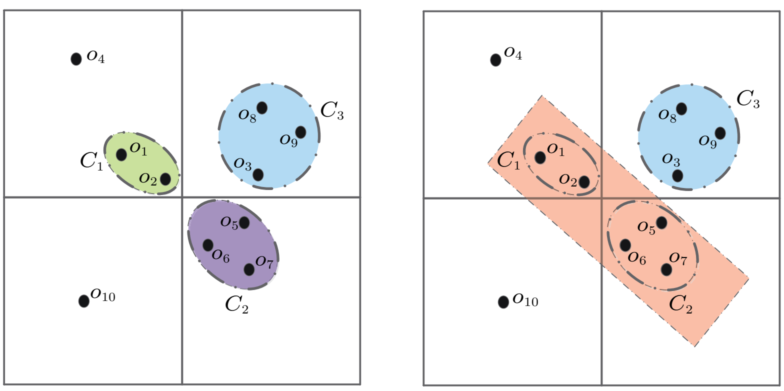
Algorithm 1 . STC-D

Require: a set of POIs $\mathcal{O}$
Ensure: a set of OPGs $\mathcal{C}$
1: **for** each unclustered object o_k in $\mathcal{O}$ **do**
2:   $que \leftarrow empty$, $c_i \leftarrow empty$
3:   $que.push(o_k)$
4:   **while** $!que.isEmpty()$ **do**
5:     $o_j = que.pop()$
6:     $N_j = FindNeighbors(o_j)$
7:     **if** $|N_j| \geqslant \widehat{\varphi}$ **then**
8:       $c_i.push(o_j)$
9:       **for** each $n \in N_j$ **do**
10:         **if** $!c_i.contain(n)$ **then**
11:           $que.push(n)$
12:         **end if**
13:       **end for**
14:     **else**
15:       $c_i.push(o_j)$
16:     **end if**
17:   **end while**
18:   **if** the size of c_i larger than $\widehat{\varphi}$ **then**
19:     Insert cluster c_i into $\mathcal{C}$
20:   **end if**
21: **end for**
22: **return** $\mathcal{C}$;

4.2 Algorithm STC-DG

However, there exist two issues in STC-D. Firstly, some irrelevant POIs can be included in an OPG or several OPGs are combined due to semantic drift of street addresses. Let us look at Fig.3. o_6 and o_7 are similar to o_5 in street addresses. o_5 and o_3 are similar in addresses. o_3 is similar to o_8 and o_9 in addresses. STC-D will put o_3, o_5, o_6, o_7, o_8 and o_9 together although o_5, o_6, o_7 belong to In City Mall and o_3, o_8, o_9 belong to Guidu Building. Secondly, STC-D cannot discover OPGs efficiently. Gan and Tao[13] claimed that DBSCAN requires $O(n^2)$ time.

Motivated by issues mentioned above, STC-DG



(a) (b)

POI	Street Address
o_1	1F In City Mall No.1699 Xiandai Blvd
o_2	2F In City Mall Xiandai Blvd
o_3	No.158 Xiandai Blvd
o_4	No.15 Piaoyi Street
o_5	No.1699 Xiandai Blvd
o_6	3F In City Mall No.1699 Xiandai Blvd
o_7	In City Mall No.1699 Xiandai Blvd
o_8	Guidu Building No.158 Xiandai Blvd
o_9	5F Guidu Building No.158 Xiandai Blvd
o_{10}	No.17 Xiandai Blvd

(c)

Fig.3. (a) Grid-based clustering. (b) Combination of clusters. (c) Street addresses of POIs.

(Spatio-Textual Clustering Based on Density and Grid) is proposed. Different from STC-D that uses density-based clustering methods, STC-DG combines density-based clustering method and grid-based clustering method. Algorithm STC-DG takes two steps. At first, it utilizes the traditional DBSCAN algorithm to cluster POIs based on their addresses. The POIs clustering is limited in one grid. Secondly, it combines clusters in adjacent grids according to the keyword of the cluster. The keyword of each cluster C_n is a street address of an object o_i in C_n and o_i satisfies that

$$\forall o_j \in C_n, \sum_{k=1}^{|C_n|} w_a(o_j, o_k) \leqslant \sum_{k=1}^{|C_n|} w_a(o_i, o_k).$$

To illustrate STC-DG, let us look at an example in Fig.3 ($\widehat{\varphi} = 2$, $\widehat{w_a} = \frac{1}{2}$, Jaccard similarity). In Fig.3(a), it utilizes the DBSCAN algorithm to cluster POIs in each grid and it obtains three clusters, i.e., C_1, C_2, and C_3 (refer to previous work [12] for DBSCAN and we replace spatial distance with address similarity). Fig.3(b) shows clusters combination. Before combining clusters, it needs to compute the keyword of each cluster. In cluster C_2, $w_a(o_5, o_6) = \frac{3}{7}$, $w_a(o_5, o_7) = \frac{1}{2}$ and $w_a(o_6, o_7) = \frac{6}{7}$. Finally, the street address of o_7 is the keyword of cluster C_2. The keywords of C_1 and C_3 are street addresses of o_1 and o_8 respectively. Due to $w_a(o_1, o_7) = \frac{6}{7} > \widehat{w_a} = \frac{1}{2}$, it combines C_1 and C_2, which is shown in Fig.3(b). In this example, all the address similarities are computed based on the Jaccard metric which is shown in (1).

Algorithm 2 is the pseudo-code of STC-DG. At first, it limits POIs clustering in one grid (line 1). Then, it extracts keyword for each cluster according to the addresses of objects in the cluster (lines 2∼4). Finally, it combines adjacent clusters on the basis of their keywords to obtain the complete OPGs (lines 6∼14). The function *IsCombine* (line 8) returns yes if keywords of two clusters are similar and two clusters are adjacent. The function *Combine* (line 9) generates a new cluster which contains all POIs of two clusters. Besides, the keyword and the location of the new cluster are recomputed. After generating a new cluster, the new cluster replaces two old clusters (lines 10 and 11). If there exists no combination, we can come to the conclusion that we have obtained all OPGs.

Algorithm 2 . STC-DG

Require: a set of POIs $\mathcal{O}$
Ensure: a set of OPGs $\mathcal{C}$

```
1: clusterlist ← ClusterBasedGrid()
2: for each cluster c_i in clusterlist do
3:     c_i.keyword ← ExtractKeyword(c_i)
4:     Insert c_i into C
5: end for
6: while existing combination do
7:     for each pair clusters c_j and c_k in C do
8:         if IsCombine(c_j, c_k) then
9:             c_m ← Combine(c_j, c_k)
10:            Insert c_m into C
11:            Delete c_j, c_k
12:        end if
13:    end for
14: end while
15: return C
```

4.3 Algorithm STC-DG+

To improve the efficiency of STC-DG, we devise three pruning rules, including length pruning, prefix pruning, and bounds pruning.

4.3.1 *Length Pruning*

The basic concept of length pruning is that similar strings cannot have a large length difference.

Theorem 1. *Given two POIs o_i and o_j, if they do not meet the inequalities as follows, this pair can be pruned. For different metrics* (*shown in* (1), (2) *and* (3)), *there exist different inequalities.*

$$Jaccard: \widehat{w_a}|o_i.a| \leqslant |o_j.a| \leqslant \frac{|o_i.a|}{\widehat{w_a}}. \quad (4)$$

$$Cosine: \widehat{w_a}^2|o_i.a| \leqslant |o_j.a| \leqslant \frac{|o_i.a|}{\widehat{w}_a^2}. \quad (5)$$

$$Dice: \frac{\widehat{w_a}|o_i.a|}{2-\widehat{w_a}} \leqslant |o_j.a| \leqslant \frac{(2-\widehat{w_a})|o_i.a|}{\widehat{w_a}}. \quad (6)$$

Proof. If $o_i.a$ and $o_j.a$ are similar based on Jaccard metric, we have $\frac{|\mathcal{T}_{ij}|}{|o_i.a|+|o_j.a|-|\mathcal{T}_{ij}|} \geqslant \widehat{w_a}$ according to (1) and then $\widehat{w_a}(|o_i.a| + |o_j.a|) \leqslant (1 + \widehat{w_a})|\mathcal{T}_{ij}|$. Due to that $|\mathcal{T}_{ij}|$ is not greater than $|o_j.a|$, we have $\widehat{w_a}(|o_i.a| + |o_j.a|) \leqslant (1 + \widehat{w_a})|o_j.a|$. Finally, we obtain $|o_j.a| \geqslant \widehat{w_a}|o_i.a|$. Due to $|\mathcal{T}_{ij}|$ no greater than $|o_i.a|$, $\widehat{w_a}(|o_i.a| + |o_j.a|) \leqslant (1 + \widehat{w_a})|o_i.a|$, we have $|o_j.a| \leqslant \frac{|o_i.a|}{\widehat{w_a}}$. Hence, we have proved (4). (5) and (6) can be proved using the same method. □

4.3.2 *Prefix Pruning*

We select two prefixes from two strings. If the two prefixes have no overlaps, then the two strings are not similar.

Theorem 2. *If $o_i.a$ and $o_j.a$ are similar, the length of the longest common subsequence $|\mathcal{T}_{ij}|$ must exceed $\mathcal{L}$. For different metrics* (*shown in* (1), (2) *and* (3)), *$\mathcal{L}$ is different.*

$$Jaccard: \mathcal{L} = \frac{\widehat{w_a}(|o_i.a| + |o_j.a|)}{(1 + \widehat{w_a})}.$$

$$Cosine: \mathcal{L} = \widehat{w_a}\sqrt{|o_i.a| \times |o_j.a|}.$$

$$Dice: \quad \mathcal{L} = \frac{\widehat{w_a}(|o_i.a| + |o_j.a|)}{2}.$$

Proof. If $o_i.a$ and $o_j.a$ are similar based on Jaccard, we have $\frac{|\mathcal{T}_{ij}|}{|o_i.a|+|o_j.a|-|\mathcal{T}_{ij}|} \geqslant \widehat{w_a}$ according to (1). Then, $\widehat{w_a}(|o_i.a| + |o_j.a|) \leqslant (1 + \widehat{w_a})|\mathcal{T}_{ij}|$. Finally, $|\mathcal{T}_{ij}| \geqslant \frac{\widehat{w_a}(|o_i.a|+|o_j.a|)}{1+\widehat{w_a}}$ and $\mathcal{L} = \frac{\widehat{w_a}(|o_i.a|+|o_j.a|)}{1+\widehat{w_a}}$. We can obtain $\mathcal{L}$ based on Cosine and Dice in the same way. □

The length of the prefix equals the length of the object minus $\mathcal{L}$. For example, there are two strings "ABCDE" and "HKDEM" and $\widehat{w_a}$ is 0.8. The lengths of two prefixes are both $\lceil 5 - \frac{0.8\times(5+5)}{1+0.8} \rceil = 1$ based on Jaccard. As two selected prefixes "A" and "H" have no overlaps, two strings are not similar.

4.3.3 Bounds Pruning

Given two POIs o_m, o_n, and a matrix $\boldsymbol{M}$ with $|o_m.a|+1$ rows and $|o_n.a|+1$ columns, $M[i,j]$ denotes the length of the longest common subsequence between $o_m.a_1^i$ and $o_n.a_1^j$ where a_k^r represents the substring of a starting from the k-th character to the r-th character (according to the definition of $M[i,j]$, we have $0 \leqslant M[i,j]-M[i-1,j] \leqslant 1$ and $0 \leqslant M[i,j]-M[i,j-1] \leqslant 1$ which will be used in the proof). Hence, we aim to obtain $M[|o_m.a|, |o_n.a|]$ which is the length of the longest common subsequence between $o_m.a$ and $o_n.a$. The value $M[i,j]$ can be computed as follows.

$$M[i,j] = \begin{cases} 0, \quad \text{if } (i=0 \text{ or } j=0), \\ M[i-1,j-1]+1, \\ \quad \text{if } (o_m.a_i^i = o_n.a_j^j), \\ \max(M[i-1,j], M[i,j-1]), \\ \quad \text{otherwise.} \end{cases}$$

The basic idea of bounds pruning is that we compute a lower bound and an upper bound of w_a in each step to terminate the computation ahead of time. In each step k, we compute a set of values $\mathcal{S}(k) = \{M[i,j] \mid i+j = k+1\}$ (shown in Table 2).

Theorem 3. *Consider a set of values* $\mathcal{S}(k) = \{M[i,j] \mid i+j = k+1$ *and* $1 \leqslant i \leqslant |o_m.a|$ *and* $1 \leqslant j \leqslant |o_n.a|\}$ *which can be obtained in step k. Matrix* $\boldsymbol{M}$ *has the property that* $\max(\mathcal{S}(k)) \leqslant \max(\mathcal{S}(k+1))$.

Proof. According to the definition of $M[i,j]$, we get $M[i,j] \leqslant M[i+1,j]$ and $M[i,j] \leqslant M[i,j+1]$. For each value $M[n,k+1-n]$ in $\mathcal{S}(k)$, there exists $M[n,k+2-n]$ or $M[n+1,k+1-n]$ in $\mathcal{S}(k+1)$ no less than $M[n,k+1-n]$. Hence, $\max(\mathcal{S}(k)) \leqslant \max(\mathcal{S}(k+1))$. □

Table 2. Illustration of Computation Process of $\boldsymbol{M}$

Step	Computed Value
1	$\{M[1,1]\}$
2	$\{M[1,2], M[2,1]\}$
⋮	⋮
k	$\{M[i,j] \mid i+j=k+1\}$
⋮	⋮
$\lvert o_m.a\rvert + \lvert o_n.a\rvert - 1$	$\{M[\lvert o_m.a\rvert, \lvert o_n.a\rvert]\}$

Therefore, we can conclude that the maximal value of $\mathcal{S}(i)$ $(1 \leqslant i \leqslant |o_m.a| + |o_n.a| - 1)$ is no larger than $M[|o_m.a|, |o_n.a|]$ which is the maximal value in the last step. If we utilize the maximal value of $\mathcal{S}(i)$ to obtain w_a which exceeds $\widehat{w_a}$, then $o_n.a$ and $o_m.a$ are similar and we can terminate the computation.

Theorem 4. *Given a set of values* $\mathcal{G}(k) = \{M[i,j] + g(i,j) \mid i+j = k+1\}$ *where* $g(i,j) = \min(|o_m.a| - i, |o_n.a| - j)$, *matrix* $\boldsymbol{M}$ *has the property that* $\max(\mathcal{G}(k)) \leqslant \max(\mathcal{G}(k-1)) + 1$ *and* $\max(\mathcal{G}(k)) \leqslant \max(\mathcal{G}(k-2))$.

Proof. Let $M[i,k+1-i] + g(i,k+1-i)$ be the maximal value of $\mathcal{G}(k)$.

If $M[i,k+1-i] = M[i-1,k-i]+1$, since $g(i,k+1-i) = g(i-1,k-i)-1$ and $M[i,k+1-i] + g(i,k+1-i) = M[i-1,k-i] + g(i-1,k-i)$, we have $\max(\mathcal{G}(k)) \leqslant \max(\mathcal{G}(k-2))$. Besides, $M[i,k+1-i] \leqslant M[i-1,k+1-i]+1$ and $g(i,k+1-i) \leqslant g(i-1,k+1-i)$. Finally, we have $\max(\mathcal{G}(k)) \leqslant \max(\mathcal{G}(k-1))+1$.

If $M[i,k+1-i]$ equals 0 or $\max(M[i,k+1-i-1], M[i-1,k+1-i])$, $\max(\mathcal{G}(k)) \leqslant \max(\mathcal{G}(k-1))+1$ and $\max(\mathcal{G}(k)) \leqslant \max(\mathcal{G}(k-2))$ can be proved in the same way. □

$M[|o_m.a|, |o_n.a|]$ is a value of $\mathcal{G}(|o_m.a| + |o_n.a| - 1)(g(|o_m.a|, |o_n.a|) = 0)$. According to Theorem 4, we obtain that $\max(\mathcal{G}(|o_m.a| + |o_n.a| - 1))$ is no greater than $\max(\mathcal{G}(|o_m.a| + |o_n.a| - 2)) + 1$ and $\max(\mathcal{G}(|o_m.a|+|o_n.a|-3))+1$. Besides, we can also obtain that $\max(\mathcal{G}(|o_m.a|+|o_n.a|-1))$ is no greater than $\max(\mathcal{G}(|o_m.a| + |o_n.a| - 4)) + 1$ and $\max(\mathcal{G}(|o_m.a| + |o_n.a| - 5)) + 1$ because $\max(\mathcal{G}(|o_m.a| + |o_n.a| - 2)) \leqslant \max(\mathcal{G}(|o_m.a|+|o_n.a|-4))$ and $\max(\mathcal{G}(|o_m.a|+|o_n.a|-3)) \leqslant \max(\mathcal{G}(|o_m.a|+|o_n.a|-5))$. In the same way, we can have $\max(\mathcal{G}(|o_m.a| + |o_n.a| - 1)) \leqslant \max(\mathcal{G}(k)) + 1$ where $1 \leqslant k \leqslant |o_m.a| + |o_n.a| - 1$. Finally, $M[|o_m.a|, |o_n.a|] \leqslant \max(\mathcal{G}(k)) + 1$ where $1 \leqslant k \leqslant |o_m.a| + |o_n.a| - 1$.

Therefore, we utilize the maximal value of $\mathcal{G}(k)+1$ to compute w_a. If w_a is smaller than $\widehat{w_a}$, then the pair is dissimilar and we can terminate the computation ahead of time.

All of the three rules are affected by $\widehat{w_a}$. Different rules are applicable in different situations. The details will be further described in Section 6.

5 Extracting Functions of OPGs

We have introduced discovering OPGs. Then we will describe the second step for discovering FOPGs, i.e., extracting functions of each OPG for further recommendation.

5.1 Analogy

We find that discovering functions of OPGs is similar to discovering topics of each document. Table 3 makes an analogy between OPGs-functions and documents-topics. There exist many solutions for discovering topics in documents, such as TF-IDF[37], and latent Dirichlet allocation (LDA)[38]. LDA has a better performance than other methods in discovering functions of regions[15] which is similar to our work. Hence, OPGs-LDA is used to discover functions of OPGs.

Table 3. Analogy Between OPGs-Functions and Documents-Topics

OPGs-Function	Documents-Topic
A set of OPGs	Corpus
Tags of POIs	Words
An OPG	A document
Functions of OPGs	Topics of documents

5.2 Details of OPGs-LDA

OPGs-LDA is a generative process of all tags in OPGs. The generative process is as follows:

1) choose a functions-tags distribution;

2) choose an OPGs-functions distribution;

3) for each tag in an OPG, there exist two steps: a) choose a function from the OPGs-functions distribution; b) according to selected function, choose a tag from the functions-tags distribution,

Fig.4 shows the OPGs-LDA model. In Fig.4, nodes represent variables, edges denote possible dependences, and plates denote replicated structures. It should be noted that shaded nodes represent observed variables. $t_{k,n}$ represents the n-th tags of the k-th OPG which depends on the parameter of functions-tags distribution β and the n-th function of the k-th OPG $f_{k,n}$. The N plates denote the collection tags within OPGs. The n-th function of the k-th OPG depends on functions proportion of the k-th OPG θ_k. θ_k depends on the parameter of OPGs-functions distribution α. The M plates denote the collection OPGs. Firstly, OPGs-LDA chooses functions-tags and OPGs-functions distributions. Dirichlet distribution is selected because it is widely used[38-39]. Then, EM algorithm or Gibbs sampling can be used for estimating parameters β and α (see [38] for more details).

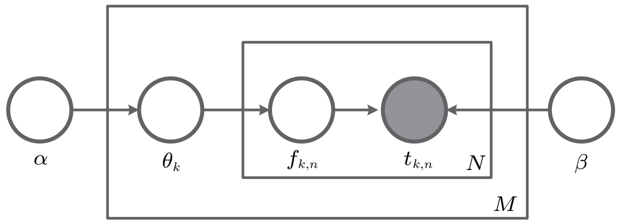


Fig.4. OPGs-LDA model.

6 Experiments

In this section, we conduct extensive experiments on real datasets to evaluate the performance of solutions. All algorithms are written in C++ and results are generated in a computer with Intel 3.2 GHz Core CPU and 8 GB memory. We have two kinds of real datasets①. The dataset of 2015① contains 414 138 POIs in Beijing. The dataset of 2005① contains 152 807 POIs in Beijing. Each POI consists of latitude, longitude, street address and tags.

6.1 Effectiveness Evaluation

This section consists of two parts. We first evaluate the accuracy of discovered FOPGs. Second, we compare POIs-based recommendation with FOPGs-based recommendation.

6.1.1 Accuracy Analysis

To evaluate the accuracy of discovered FOPGs, we ask volunteers to manually label which FOPG each POI belongs to. We focus on the following three performance metrics to evaluate the accuracy of discovered FOPGs.

• *Textual/Address Dispersion.* There exists semantic drift in finding FOPGs. Textual dispersion is proposed to evaluate textual quality. Given a set of FOPGs $\mathcal{C} = \{c_1, c_2, ..., c_m\}$ where $c_i = \{o_1, o_2, ..., o_{|c_i|}\}$, let

①http://lbsyun.baidu.com/, Aug. 2017.

$D_t(i,j) = 1 - w_a(o_i, o_j)$ be the textual distance between o_i and o_j, and then the textual dispersion of $\mathcal{C}$ is defined as

$$TD(\mathcal{C}) = \frac{\sum_{k=1}^{|\mathcal{C}|} \left(\frac{2 \times \sum_{i \neq j, o_i, o_j \epsilon c_k} D_t(i,j)}{|c_k| \times (|c_k| - 1)}\right)}{|\mathcal{C}|}.$$

The challenge is to keep textual dispersion as low as possible.

- *Precision Rate.* Some irrelevant POIs could be put into FOPGs. Hence, it is necessary to evaluate the precision rate of discovered FOPGs. STC-D or STC-DG discovers M POIs that belong to the FOPG c_i. However, only N POIs of M discovered POIs actually belong to the FOPG c_i, and then precision rate is defined as follows.

$$PR = \frac{N}{M}.$$

The challenge is to keep precision rate as high as possible.

- *Recall Rate.* Some POIs in an FOPG could be eliminated. Hence, recall rate is proposed to evaluate discovered FOPGs. Given an FOPG c_i containing K POIs, STC-D or STC-DG only finds N POIs that actually belong to c_i, and then recall rate is defined as follows.

$$RR = \frac{N}{K}.$$

The challenge is to keep recall rate as high as possible.

We firstly evaluate the textual dispersion, precision rate and recall rate of STC-D and STC-DG. Secondly, we analyze the effect of parameters on textual dispersion, precision rate and recall rate.

As shown in Fig.5(a) and Fig.5(d), it is clear that the textual dispersion of STC-DG is smaller than that of STC-D because limiting POIs clustering in a grid reduces semantic drift greatly. Fig.5(b) and Fig.5(e) show that the precision rate of STC-DG is higher than that of STC-D. However, STC-DG has its drawback that the recall rate of STC-DG is lower than that of STC-D (shown in Fig.5(c) and Fig.5(f)) because border POIs of an FOPG are easily eliminated when grid clustering is applied.

Figs.5(a)$\sim$5(c) show the impact of grid granularity g on the textual dispersion, precision rate and recall rate respectively. From Figs.5(a)$\sim$5(c), we can clearly see that the increasing of g leads to the increasing of the textual dispersion and the decreasing of the precision rate because the larger the grid is, the higher the possibility that irrelevant and dissimilar POIs are put into FOPGs is. Besides, when the grid is small, the number of POIs in the grid is possibly smaller than $\widehat{\varphi}$ which results in elimination of POIs. Hence, the increasing of g leads to the higher recall rate.

Fig.5(d) shows the impact of textual similarity

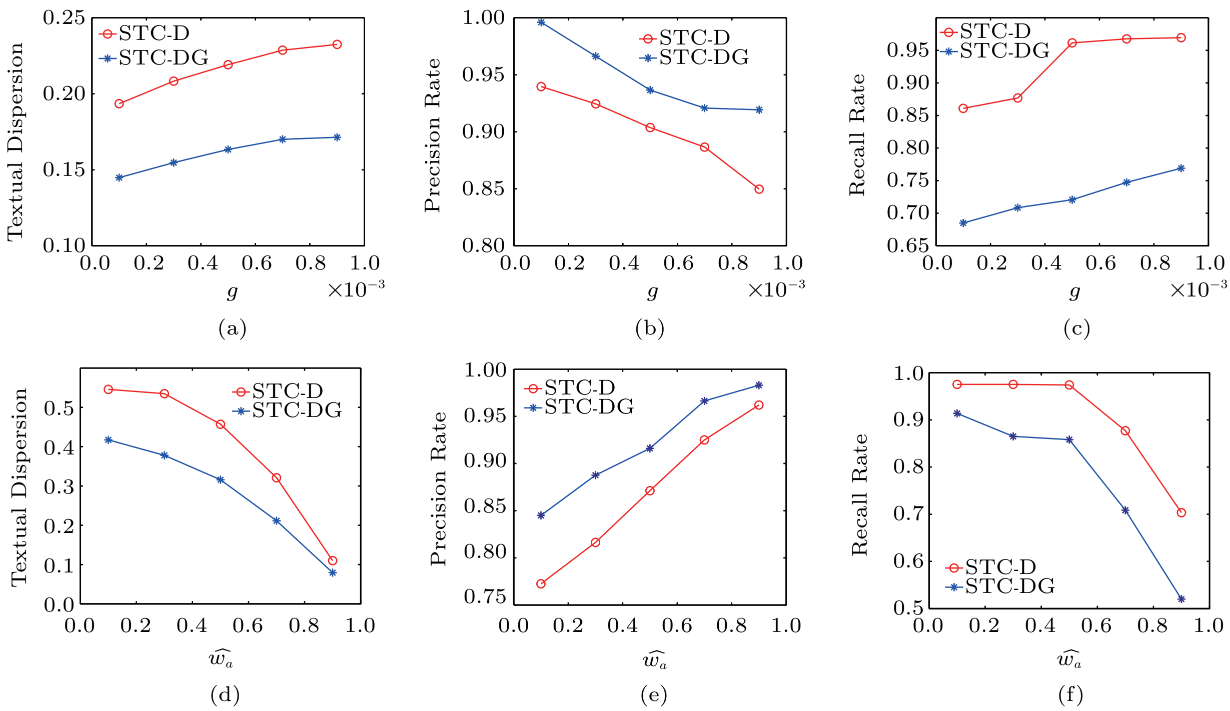


Fig.5. Effectiveness evaluation.

threshold $\widehat{w_a}$ on the textual dispersion. The increasing of textual similarity threshold $\widehat{w_a}$ leads to the decreasing of textual dispersion because the higher $\widehat{w_a}$ is, the more similar the addresses are. Fig.5(e) shows that the increasing of textual similarity threshold $\widehat{w_a}$ contributes to the higher precision rate because when $\widehat{w_a}$ is larger, more irrelevant POIs are eliminated. Besides, if $\widehat{w_a}$ is very large, a part of relevant POIs will be eliminated as well. Hence, as shown in Fig.5(f), recall rate decreases when $\widehat{w_a}$ increases.

6.1.2 *FOPGs Versus POIs Recommendation*

FOPGs are proposed due to the limitations of POIs-based recommendation. Compared with POIs-based recommendation, FOPGs-based recommendation has two advantages. 1) Recommending FOPGs offers users more flexible choices than recommending POIs. 2) Recommending FOPGs is more efficient than recommending POIs because the number of FOPGs is much smaller than that of POIs. Recommendation details can refer to the previous work[27].

We compare FOPGs recommendation with POIs recommendation based on recommendation time (RT) and the variety of POIs (VoP). Given a set of query keywords $K = \{k_1, k_2, \ldots, k_{|K|}\}$ and a set of returned POIs, for each keyword k_i, there are P_i POIs covering k_i, and then VoP is

$$VoP = \frac{\sum_{i=1}^{|K|} P_i}{|K|}.$$

The challenge is to make VoP as large as possible.

In Fig.6, we can see that the FOPGs recommendation is several orders of magnitude faster than the POIs recommendation. This results from two factors. 1) The number of POIs is much larger than that of FOPGs. 2) An FOPG offers more keywords than a POI. If a user queries three keywords, e.g., gym, restaurant, and cinema, POIs recommendation offers three POIs while FOPGs recommendation offers only one FOPG.

From Fig.7, we can see that VoP of POIs recommendation is almost 1 because most POIs have only one keyword. Given n query keywords, POIs recommendation offers n POIs. It cannot offer more flexible choices when users want to shop around before buying. FOPGs recommendation can offer more choices for users. From Fig.7, we can clearly see that VoP of FOPGs recommendation ranges from 10 to 50. It means that if a user wants to buy clothes, FOPGs recommendation can offer at least 10 adjacent clothes shops, which is flexible for users.

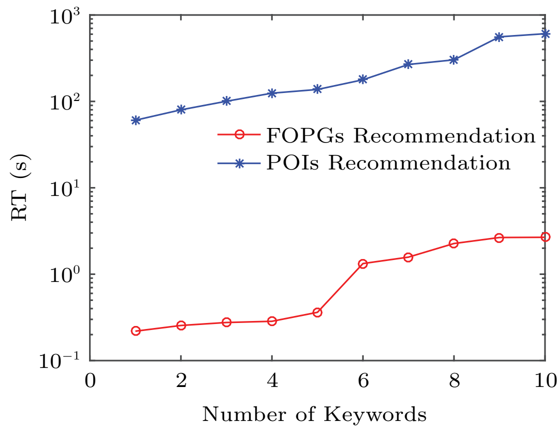


Fig.6. Evaluation of recommendation time.

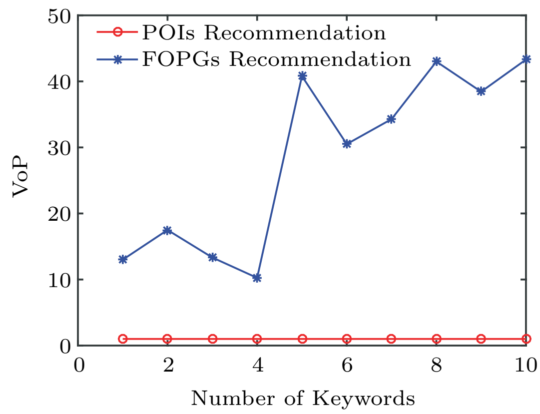


Fig.7. Evaluation of variety of POIs.

6.2 Efficiency Evaluation

In this subsection, we evaluate the efficiency of three methods, i.e., STC-D, STC-DG, and STC-DG$^+$ based on Jaccard metric, Cosine metric and Dice metric. Besides, we will analyze impacts of grid granularity g, the threshold of cluster size $\widehat{\varphi}$ and the threshold of textual similarity $\widehat{w_a}$ on the efficiency.

From Figs.8(a)$\sim$8(c) and Figs.9(a)$\sim$9(c), we can see that the rate of time growth of STC-D is obviously higher than that of STC-DG because grid-based clustering can greatly simplify the computation. Besides, the running time of STC-DG$^+$ is less than that of STC-DG due to three pruning rules. Then, we analyze the impact of grid granularity on running time. From Fig.8 and Fig.9, we can see that the increasing of g leads to the increasing of time cost. When g is large, POIs in farther distance can be clustered, which directly increases the running time. Besides, we find a phenomenon that when grid granularity is small, STC-D performs better

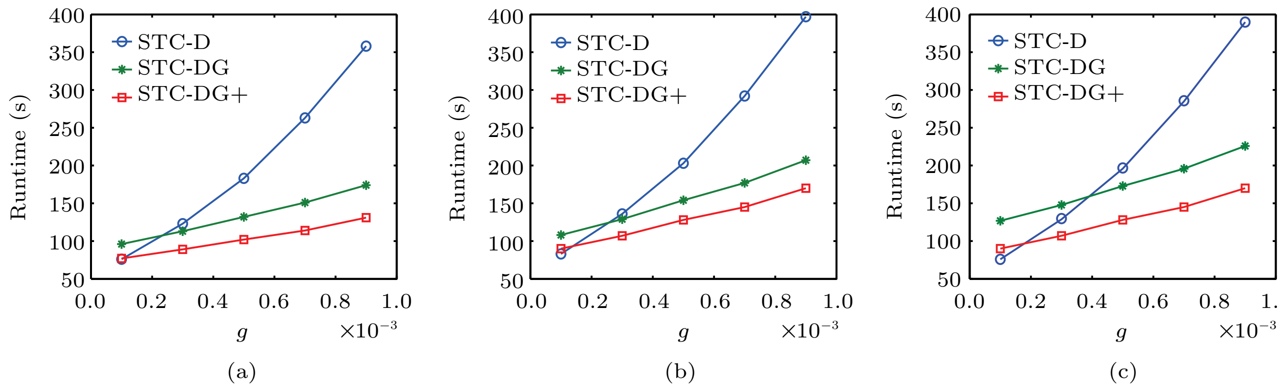


Fig.8. Efficiency evaluation based on the dataset of 2015. (a) Jaccard metric. (b) Cosine metric. (c) Dice metric.

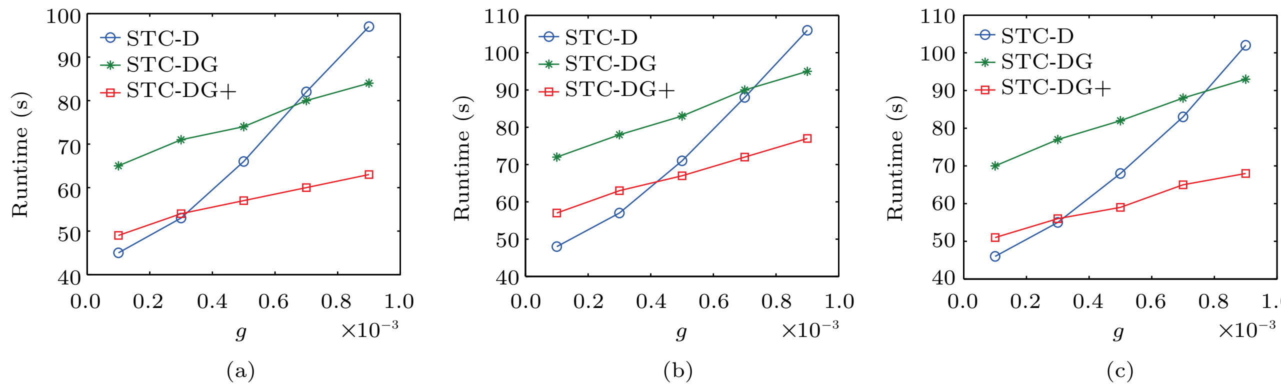


Fig.9. Efficiency evaluation based on the dataset of 2005. (a) Jaccard metric. (b) Cosine metric. (c) Dice metric.

than STC-DG. The small grid granularity leads to the increasing number of grids. A large number of grids result in many cluster combinations in STC-DG. When grid granularity is small, STC-D performs better than STC-DG because of the large cost of clusters combinations in STC-DG.

Next, we will evaluate impacts of other parameters on the efficiency. Fig.10(a) shows that running time decreases when $\widehat{\varphi}$ increases. When $\widehat{\varphi}$ is large, more POI sets whose sizes are smaller than $\widehat{\varphi}$ will be ignored, which directly affects the time cost. Fig.10(b) shows that the running time decreases when $\widehat{w_a}$ increases. When $\widehat{w_a}$ increases, each pair of POIs in an FOPG must have more similar street addresses, which leads to a smaller size of a cluster. The smaller clusters contribute to reducing the time cost.

Finally, we evaluate the performance of three pruning rules, i.e., length pruning, prefix pruning and bounds pruning. Fig.10(c) shows the efficiency of three pruning rules under different $\widehat{w_a}$. When $\widehat{w_a}$ is small, the length pruning and the prefix pruning rules are not good and they even reduce the efficiency because of the extra overhead. With the increasing of $\widehat{w_a}$, more dissimilar pairs can be pruned ahead of time. Hence, the two rules perform well when $\widehat{w_a}$ is large. Contrary to the length pruning and the prefix pruning, the bounds pruning shows great performance when $\widehat{w_a}$ is low and the bounds pruning is not good when $\widehat{w_a}$ increases due to the extra computation of bounds.

7 Conclusions

This paper studied a new problem of discovering FOPGs. To the best of our knowledge, we are the first to discover FOPGs for spatial keyword recommendation. We proposed a two-step solution for discovering FOPGs. In the first step, we designed two algorithms, i.e., STC-D and STC-DG. Besides, we proposed three pruning rules to improve the efficiency of STC-DG. In the second step, we proposed OPGs-LDA model to discover functions of OPGs for further recommendation.

To evaluate the feasibility of our solutions, we conducted extensive experiments on two real-world datasets. The experimental results demonstrated the effectiveness and efficiency of the proposed algorithms. Besides, we analyzed the effect of parameters on effec-

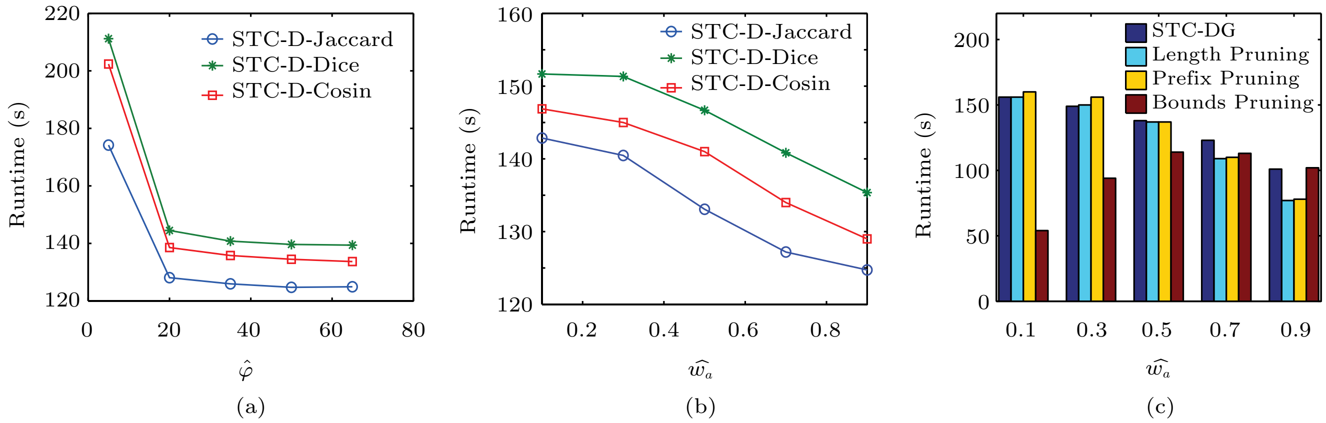


Fig.10. Parameters effect on efficiency. (a) Effect of $\hat{\varphi}$. (b) Effect of $\widehat{w_a}$. (c) Effect of pruning rules.

tiveness and efficiency of algorithms.

The information of POIs will change all the time. Besides, new POIs will be added and old POIs will be deleted. Once changes happen, it is expensive to discover FOPGs from the scratch. Hence, we will study effective update algorithms in the future work.

References

[1] Yin H, Zhou X, Cui B, Wang H, Zheng K, Nguyen Q V H. Adapting to user interest drift for POI recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 2016, 28(10): 2566-2581.

[2] Wen Y, Cho K, Peng W, Yeo J, Hwang S. KSTR: Keyword-aware skyline travel route recommendation. In *Proc. the 15th IEEE International Conference on Data Mining*, November 2015, pp.449-458.

[3] Cao X, Cong G, Jensen S C, Ooi B C. Collective spatial keyword querying. In *Proc. the 2011 ACM SIGMOD International Conference on Management of Data*, June 2011, pp. 373-384.

[4] Yuan Q, Cong G, Sun A. Graph-based point-of-interest recommendation with geographical and temporal influences. In *Proc. the 23rd ACM International Conference on Information and Knowledge Management*, November 2014, pp.659-668.

[5] Chen W, Zhao L, Xu J, Liu G, Zheng K, Zhou X. Trip oriented search on activity trajectory. *Journal of Computer Science and Technology*, 2015, 30(4): 745-761.

[6] Yin H, Zhou X, Shao Y, Wang H, Sadiq S. Joint modeling of user check-in behaviors for point-of-interest recommendation. In *Proc. the 24th ACM International Conference on Information and Knowledge Management*, October 2015, pp.1631-1640.

[7] Yin H, Wang W, Wang H, Chen L, Zhou X. Spatial-aware hierarchical collaborative deep learning for POI recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 2017, 29(11): 2537-2551.

[8] Guo Y, Xu C, Song H, Wang X. Understanding users' budgets for recommendation with hierarchical Poisson factorization. In *Proc. the 26th International Joint Conference on Artificial Intelligence*, August 2017, pp.1781-1787.

[9] Guo T, Cao X, Cong G. Efficient algorithms for answering the m-closest keywords query. In *Proc. the 2015 ACM SIGMOD International Conference on Management of Data*, June 2015, pp.405-418.

[10] Li W, Cao J, Guan J, Yiu M L, Zhou S. Efficient retrieval of bounded-cost informative routes. *IEEE Transactions on Knowledge and Data Engineering*, 2017, 29(10): 2182-2196.

[11] Gao Y, Zhao J, Zheng B, Chen G. Efficient collective spatial keyword query processing on road networks. *IEEE Transactions on Intelligent Transportation Systems*, 2016, 17(2): 469-480.

[12] Ester M, Kriegel H, Sander J, Xu X. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. the 2nd International Conference on Knowledge Discovery and Data Mining*, August 1996, pp.226-231.

[13] Gan J, Tao Y. DBSCAN revisited: Mis-claim, un-fixability, and approximation. In *Proc. the 2015 ACM SIGMOD International Conference on Management of Data*, June 2015, pp.519-530.

[14] Jiang Y, Li G, Feng J, Li W. String similarity joins: An experimental evaluation. *Proceedings of the VLDB Endowment*, 2014, 7(8): 625-636.

[15] Yuan J, Zheng Y, Xie X. Discovering regions of different functions in a city using human mobility and POIs. In *Proc. the 18th ACMSIGKDD International Conference on Knowledge Discovery and Data mining*, August 2012, pp.186-194.

[16] Feng K, Cong G, Bhowmick S S, Peng W, Miao C. Towards best region search for data exploration. In *Proc. the 2016 International Conference on Management of Data*, July 2016, pp.1055-1070.

[17] Yin H, Cui B, Chen L, Hu Z, Zhang C. Modeling location-based user rating profiles for personalized recommendation. *ACM Transactions on Knowledge Discovery from Data*, 2015, 9(3): Article No. 9.

[18] Zhu W, Peng W, Chen L, Zheng K, Zhou X. Modeling user mobility for location promotion in location-based social networks. In *Proc. the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 2015, pp.1573-1582.

[19] Yin H, Hu Z, Zhou X, Wang H, Zheng K, Nguyen Q V H, Sadiq S. Discovering interpretable geo-social communities for user behavior prediction. In *Proc. the 32nd IEEE International Conference on Data Engineering*, May 2016, pp.942-953.

[20] Tong Y, Chen L, Zhou Z, JagadishH V, Shou L, Lv W. SLADE: A smart large-scale task decomposer in crowdsourcing. *IEEE Transactions on Knowledge and Data Engineering*. doi:10.1109/TKDE.2018.2797962. (preprint)

[21] Tong Y, She J, Ding B, Wang L, Chen L. Online mobile micro-task allocation in spatial crowdsourcing. In *Proc. the 32nd IEEE International Conference on Data Engineering*, May 2016, pp.49-60.

[22] Liu J, Yu G, Sun H. Subject-oriented top-k hot region queries in spatial dataset. In *Proc. the 20th ACM International Conference on Information and Knowledge Management*, October 2011, pp.2409-2412.

[23] Choi D, Chung C, Tao Y. A scalable algorithm for maximizing range sum in spatial databases. *Proceedings of the VLDB Endowment*, 2012, 5(11): 1088-1099.

[24] Cao X, Cong G, Jensen C S, Yiu M L. Retrieving regions of interest for user exploration. *Proceedings of the VLDB Endowment*, 2014, 7(9): 733-744.

[25] Han J, Kamber M, Pei J. Data Mining: Concepts and Techniques (Southeast Asia edition). Morgan Kaufmann, 2006.

[26] Chen L, Cong G, Jensen C S, Wu D. Spatial keyword query processing: An experimental evaluation. *Proceedings of the VLDB Endowment*, 2013, 6(3): 217-228.

[27] Long C, Wong R C W, Wang K, Fu A W C. Collective spatial keyword queries: A distance owner-driven approach. In *Proc. the 2013 ACM SIGMOD International Conference on Management of Data*, June 2013, pp.689-700.

[28] Christoforaki M, He J R, Dimopoulos C, Markowetz A, Suel T. Text vs. space: Efficient geo-search query processing. In *Proc. the 20th ACM International Conference on Information and Knowledge Management*, October 2011, pp.423-432.

[29] Zhang J, Chow C, Li Y. LORE: Exploiting sequential influence for location recommendations. In *Proc. the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, November 2014, pp.103-112.

[30] Gao H, Tang J, Hu X, Liu H. Exploring temporal effects for location recommendation on location-based social networks. In *Proc. the 7th ACM Conference on Recommender Systems*, October 2013, pp.93-100.

[31] Wang W, Yin H, Chen L, Sun Y, Sadiq S, Zhou X. Geo-SAGE: A geographical sparse additive generative model for spatial item recommendation. In *Proc. the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 2015, pp.1255-1264.

[32] Yin H, Cui B. Spatio-Temporal Recommendation in Social Media. Springer, 2016.

[33] Xie M, Yin H, Wang H, Xu F, Chen W, Wang S. Learning graph-based POI embedding for location-based recommendation. In *Proc. the 25th ACM International Conference on Information and Knowledge Management*, October 2016, pp.15-24.

[34] Yin H, Cui B, Zhou X, Wang W, Huang Z, Sadiq S. Joint modeling of user check-in behaviors for real-time point-of-interest recommendation. *ACM Transactions on Information Systems*, 2016, 35(2): Article No. 11.

[35] Bergroth L, Hakonen H, Raita T. A survey of longest common subsequence algorithms. In *Proc. the 7th International Symposium on String Processing and Information Retrieval*, September 2000, pp.39-48.

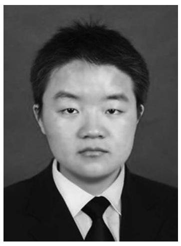
[36] Bouros P, Ge S, Mamoulis N. Spatio-textual similarity joins. *Proceedings of the VLDB Endowment*, 2012, 6(1): 1-12.

[37] Leskovec J, Rajaraman A, Ullman J D. Mining of Massive Datasets (2nd edition). Cambridge University Press, 2014.

[38] Blei D M, Ng A Y, Jordan M I. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 2003, 3: 993-1022.

[39] Blei D M. Probabilistic topic models. *Communications of the ACM*, 2012, 55(4): 77-84.

Yan-Xia Xu received her B.S. degree in computer science from Soochow University, Suzhou, in 2015. She is currently a M.S. candidate in Soochow University, Suzhou. Her research interests include data mining, spatial-temporal database and graph computing.



Wei Chen is currently a Ph.D. candidate in the School of Computer Science and Technology, Soochow University, Suzhou. His research interests include data mining and spatial-temporal database.

Jia-Jie Xu is an associate professor at the School of Computer Science and Technology, Soochow University, Suzhou. He got his Ph.D. and M.S. degrees from the Swinburne University of Technology, Victoria, and the University of Queensland, Brisbane, in 2011 and 2006 respectively. Before joining Soochow University in 2013, he worked as an assistant professor in the Institute of Software, Chinese Academy of Sciences, Beijing. His research interests mainly include spatio-temporal database systems, big data analytics and workflow systems.

Zhi-Xu Li is an associate professor in the School of Computer Science and Technology at Soochow University, Suzhou. He worked as a research fellow at King Abdullah University of Science and Technology, Thuwal. He received his Ph.D. degree in computer science from the University of Queensland, Brisbane, in 2013, and his B.S. and M.S. degrees in computer science from Renmin University of China, Beijing, in 2006 and 2009 respectively. His research interests include data cleaning, big data applications, information extraction and retrieval, machine learning, deep learning, knowledge graph and crowdsourcing.

Guan-Feng Liu is an associate professor at the School of Computer Science and Technology, Soochow University, Suzhou. He got his Ph.D. degree in computer science from Macquarie University, New South Wales, in 2013, his M.Eng. degree in computer software and theory from Qingdao University (QDU), Qingdao, in 2008, and his B.Eng. degree in computer science and technology from Qingdao University of Science and Technology (QUST), Qingdao, in 2005. His research interests include service computing, trust computing, and social network mining.

Lei Zhao is a professor in the School of Computer Science and Technology at Soochow University, Suzhou. He received his Ph.D. degree in computer science from Soochow University, Suzhou, in 2006. His research focuses on graph databases, social media analysis, query outsourcing, parallel and distributed computing. His recent research is to analyze large graph database in an effective, efficient, and secure way.