



Three-layered location recommendation algorithm using spectral clustering

Alireza Moayedikia¹ · Hamed Jahani² · Md Anisur Rahman³ · Kok Leong Ong²

Received: 18 December 2023 / Revised: 6 April 2024 / Accepted: 20 April 2024
© The Author(s) 2024

Abstract

Users utilize Location-Based Social Networks (LBSNs) to check into diverse venues and share their experiences through ratings and comments. However, these platforms typically feature a considerably larger number of locations than users, resulting in a challenge known as insufficient historical data or user-location matrix sparsity. This sparsity arises because not all users can check into all available locations on a given LBSN, such as Yelp. To address this challenge, this paper proposes combining Spectral Clustering with a three-layered location recommendation model to develop a recommender system named LSC, applied to Yelp datasets. LSC leverages various information, including users' check-in data, demographics, location demographics, and users' friendship network data, to train the recommender system and generate recommendations. Evaluation of LSC's performance utilizes the Yelp dataset and several comparison metrics, such as accuracy, RMSE, and F1-score. The results demonstrate that our proposed algorithm delivers reliable and significant performance improvements across various evaluation metrics compared to competing algorithms.

Keywords Location based recommender systems · Location based services · Spectral clustering · Friendship network · Location based social networks · Yelp

1 Introduction

The rise in smartphone adoption and the evolution of GPS-equipped devices have spurred users to engage with Location-Based Social Networks (LBSNs) like Foursquare, Yelp, and similar platforms (Canturk and Karagoz 2021) (Rahimi et al. 2020) (Si et al. 2019). On these platforms, users consistently generate data by checking in at various locations and points of interest (POIs), leaving comments, and providing ratings (Iqbal, et al. 2019) (Wang, et al. 2019). The data generated by users is valuable for creating Recommendation Systems (RS) that suggest points of interest (POIs) to users. These recommendation systems

aid users in discovering interesting places based on their location and trajectory (Xiong 2020) (Jiao et al. 2019) (Tuan et al. 2017).

POI recommendation aids users in finding venues (such as sightseeing spots, cafes, bars, etc.) when they travel to new locations they haven't visited before (Wang, et al. 2019). In POI recommenders, the recommendation algorithm handles diverse user-related features including demographics, comments, ratings, and various spatio-temporal features. On LBSNs, users' check-ins to POIs are depicted in a matrix format where rows denote users and columns denote locations (Sojahrood and Taleai 2021) (Cai et al. 2019) (Mohammadi and Rasoolzadegan 2022).

A key challenge in POI recommendation is the sparsity of the user-location matrix. This indicates that there are more locations than users, and not all users check into all or most of the locations. Consequently, many entries in the user-location matrix become sparse (Manotumruksa et al. 2020) (Kolahkaj et al. 2020). The issue of sparse matrices becomes even more challenging in the presence of noisy data. Additionally, the proposed algorithms fail to address the cold start problem effectively due to their architecture. (Panda and Ray 2022) (Alves, et al. 2023). This means new

✉ Alireza Moayedikia
amoayedikia@swin.edu.au

¹ Department of Business Technology and Entrepreneurship, Swinburne University of Technology, Hawthorne, VIC, Australia

² School of Accounting, Information Systems and Supply Chain, RMIT University, Melbourne, VIC, Australia

³ La Trobe Business School, La Trobe University, Bundoora, VIC, Australia

users or locations should stay on the platform for some time to participate in the recommendation process.

To address these challenges, this paper introduces a three-layered location recommendation model that utilizes Spectral Clustering (LSC) and the friendship network to predict and recommend the next location to a user (Zhou et al. 2022) (Canturk et al. 2023) (Ma et al. 2020). Introducing LSC as a three-layered POI recommendation model instead of relying solely on the user-location matrix helps alleviate sparsity issues and cold-start problems simultaneously. In essence, the three-layered POI recommender model consists of three stacked and interconnected layers (refer to Fig. 1).

LSC clusters similar users and locations in the second and third layers, leveraging friendship information in the first layer to form a community of friends. This community incorporation enables tuning recommendations based on their visiting patterns. Given that user-location matrices are typically sparse, incorporating spatial information enhances POI recommendation accuracy, aiding in overcoming the cold-start problem.

Furthermore, considering the friendship network is beneficial because friends tend to share similar interests and visiting patterns compared to random users. Therefore, a location deemed interesting for one user is likely to be appealing to their friends as well (Forsati et al. 2014) (Zhou et al. 2021). LSC employs network representation to portray users, locations, and the interactions (i.e., visits) between them (Ge et al. 2021) (Zhang et al. 2021). In addition, LSC utilizes spectral clustering to generate clusters of users and locations. Spectral clustering is a widely adopted method for extracting insights from complex networks, capable of identifying clusters with non-convex shapes (Bai et al. 2023) (Alshammari et al. 2021) (Zhang et al. 2021).

Moreover, spectral clustering encodes pairwise similarity into an adjacency matrix, a process that inherently limits spectral clustering to second-order structures, such as undirected or directed edges connecting two nodes (Ge

et al. 2021) (Deng, et al. 2023). In summary, the proposed POI recommender algorithm (LSC), makes the following contributions thanks to embedding Spectral Clustering:

- **Noise resistance** (Farahani et al. 2023) (Behera and Nain 2023): In location-based datasets, encountering noisy data is common, necessitating the development of a noise-resistant algorithm. LSC demonstrates noise resistance by integrating spectral clustering, which is inherently more robust to noisy data compared to traditional methods like k-means. This resilience stems from spectral clustering's consideration of the global structure of the data rather than solely focusing on local neighborhoods.
- **Scalability** (Khan, et al. 2023a, b) (Sarkar et al. 2023): Recommender systems are anticipated to operate on large-scale datasets containing numerous data samples accompanied by a plethora of features. LSC leverages spectral clustering, which, in contrast to traditional clustering methods, is better equipped to handle high-dimensional data by transforming it into a lower-dimensional space. Consequently, this approach enhances scalability.
- **Cold start** (Forsati et al. 2014) (Panda and Ray 2022): Employing spectral clustering within a three-layered model helps in addressing the cold start problem. New users lacking visiting history are initially clustered based on their demographics and subsequently receive recommendations. Likewise, new locations devoid of visitor history are clustered according to their features and subsequently contribute to the recommendation process.
- **Considering friendship data** (Forsati et al. 2014) (Sheibani et al. 2023): In contrast to the majority of location-based recommender systems that rely solely on users' profiles and spatial attributes, LSC also takes into account friendship network data. This data assists in refining the list of recommended locations for a specific user. In practice, LSC compiles the final list of recommendations for a user and then prioritizes venues previously visited by that user's friends, enhancing their relevance.

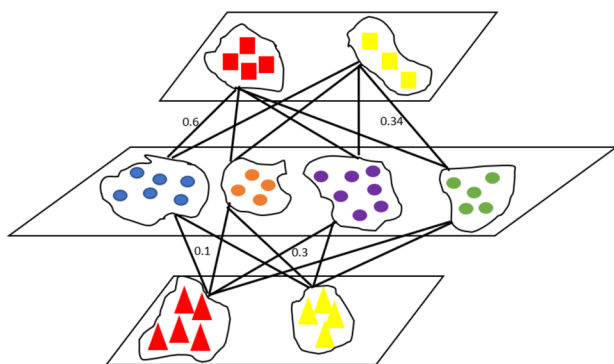


Fig. 1 Three layered graph check-in representation

Friendship network data is not the direct result (i.e., contribution) of utilising spectral clustering, because friends are created using are created through community detection algorithms (see Sect. 3).

To provide an overview on how the proposed recommender system algorithm works, LSC begins by receiving the initial networks of users and locations. It then translates these networks into a three-layer model of clusters. In this model, the top layer comprises a cluster of friendship networks, the middle layer contains the cluster of locations, and the bottom layer preserves clusters of users based on their profile similarity. Subsequently, this three-layered model is

utilized to recommend locations to both existing and new users by clustering the users and shortlisting potential locations for recommendation.

Spectral clustering enables LSC to handle complex data more efficiently. Unlike other clustering techniques such as k-means, spectral clustering excels in managing complex structured datasets. By effectively capturing complex and non-linear structures within location-based datasets, spectral clustering proves suitable for location recommendation.

Moreover, spectral clustering enhances the versatility of the LSC algorithm compared to other algorithms. This is because LSC does not impose any assumptions on the shape of clusters. In contrast, traditional clustering algorithms (e.g., k-means) assume spherical clusters, whereas spectral clustering remains agnostic to the shapes or sizes of the clusters, thus rendering the algorithm more flexible and adaptive to changes in the data. The remainder of this paper is organized as follows.

Section 2 reviews the recent research on POI recommendation. Section 3 introduces the method used for POI recommendation, followed by Sect. 4 which discusses the performance of LSC and compares it against rival algorithms before Sect. 5 concludes the paper.

2 Literature review

On location based social networks, an important feature is to recommend POI to users (Zhu, et al. 2021) (Han, et al. 2023).

POI recommendation has been extensively studied by several researchers to enhance location based social networks. Some of the existing POI recommendation algorithms fail to address the cold start problem. Dokuz and Celik (2017), present an algorithm aimed at identifying socially significant locations and analyzing users' spatial preferences. This algorithm, termed Socio-Spatially Important Locations Mining (SS-ILM), employs measures such as location density, visit lifetime, and user prevalence to quantify socially significant locations. These measures are adaptable to any social media activity dataset containing geographical and temporal information. While SS-ILM utilizes large-scale data, the authors overlook addressing how their approach tackles scalability in POI recommendation.

In another work (Rahmani et al. (2019) introduce a Behavior-based Location Recommendation (BLR) method, which forecasts users' behavior by considering their past activities alongside activities exhibited by similar users. BLR suggests a location suitable for the predicted behavior, relying on user-specific and behavior-specific spatial models. Furthermore, the authors leverage textual data sources to refine POI recommendations (Zhao et al. 2020). The authors

fail to explicitly address how their algorithm can handle the cold start problem.

Several other algorithms that propose promising methods and results have overlooked the utilization of friendship network data. Establishing trust in social recommender systems constitutes a crucial element (Manotumrukxa et al. 2020) (Forsati et al. 2014). Users tend to place more trust in recommendations associated with their friendship network compared to systematically generated recommendations, irrespective of the accuracy of the recommendation.

For instance, Missaoui, et al. (2019) propose a recommendation algorithm that analyzes users' comments on check-ins to create their profiles, identifying preferences and interests. It then compares these profiles with the formal representation of travel-related services. Using similarity metrics, the algorithm generates personalized recommendations based on users' preferences and the characteristics of travel services. This approach combines text mining with AI to provide tailored travel suggestions. Their algorithm is to some extent scalable. However, noise resistibility has not been discussed.

Gao et al. (2018) argues that traditional models often overlook implicit social influence in user modelling. To address this, they propose a novel POI recommendation method based on kernel estimation. This method utilizes a self-adaptive kernel bandwidth to model geographical influence between POIs. Additionally, trust values between users are estimated using a Gaussian radial basis kernel function-based support vector regression (SVR) model.

In contrast, some algorithms rely on the social relations of users. (Xiong (2020) utilize a latent probabilistic generative model to integrate trust relations and text mining principles, designing a POI recommendation algorithm. This algorithm is founded on the notion that optimal POI recommendations for users in specific regions can be inferred from the comments made by their friends within the same community on communication-based social networks (e.g., Facebook). In another model developed by Kefalas et al. (2018), The trust relationship has been implemented in the form of friend recommendation. The authors devised a model to construct a tripartite graph, integrating spatial and temporal dimensions into the model. This graph comprises users, locations, and sessions.

Divyaa and Pervin (2019) critique the complete dependence on trust relations and offer a contrasting viewpoint. They argue that many POI recommenders assume that users' ratings are influenced solely by their social connections in friendship networks, neglecting users' preferential similarity. Therefore, the authors introduce a collaborative filtering location recommendation algorithm that leverages the preference network concept to highlight the influence of 'Rating Bubbles'.

Wang, et al. (2019), contend that users with similar traits may not always be connected as friends in a network. Therefore, the authors propose inferring users' similarities based on their activities and behavior. They introduce a model that estimates trust between users using a network representation learning technique. This technique is based on a user co-visiting network, which suggests that two users visiting a location within a specific timeframe and geographical region likely share similar interests, thus warranting a connection between them.

Some other algorithms consider overcoming the matrix factorization problem while failing to address the presence of noisy data. Zhang et al. (2020) employ transfer learning-based artificial neural networks to initially learn users' spatial and non-spatial preferences from historical POI interactions. The model then incorporates user interactions from other domains, introducing valuable preferences into POI recommendations to tackle data sparsity issues. Similarly, Xiong (2020), utilize a probabilistic generative model to integrate factors including cross-platform textual content (extracted from user profiles on various social media platforms like Facebook), temporal effects, social communities, and geographical regions.

In another similar work, Liu and Wang (2018) introduce a POI recommendation algorithm founded on a multi-order Markov model. This model predicts users' next favorite POIs by considering not only their current location but also their previous locations. While the Markov model partially mitigates the presence of noise in the data, concerns regarding the scalability of their solution arise.

Guo et al. (2018) argue that the frequency of visits may not accurately reflect the similarity and significance between a pair of POIs. For instance, a person might visit a museum and a pharmacy six times a week. However, museums and pharmacies hold different levels of significance from a POI recommendation standpoint. The authors adopt the concept of deep learning to assess the significance of different POIs. Their proposed algorithm converts check-in data into POI preferences, which are then input into a deep learning semi-restricted Boltzmann model to estimate geographical similarity. Additionally, social influences and similarities among friends are extracted using a conditional layer. However, the authors do not integrate friendship network data to further improve their proposed algorithm. Furthermore, the cold start problem remains unaddressed.

In the next section, we introduce Layered Spectral Clustering (LSC) POI recommendation technique that is based on the novel spectral clustering method. The proposed technique resolves POI recommendation problems of trust relation and tourism data sparsity. In our proposed solution, each layer of LSC contains a set of clusters for users or locations. The LSC model is trained using past check-in data of users on Yelp, to be used for POI recommendation.

3 Three-layered spectral clustering recommender system

To resolve the challenge of data sparsity, this section proposes a three-Layered Spectral Clustering (LSC) algorithm (Khan et al. 2023a, b) for POI recommendation. First, Sect. 3.1 provides an overview of problem formulation, followed by Sects. 3.2 and 3.3 to discuss spectral clustering and LSC, respectively.

3.1 Problem formulation and solution overview

In LSC, there are L locations and U users, where users in set U have visited and rated N out of L locations, with $N \ll L$. To preserve users' check-ins, we consider a POI matrix, also known as the rating matrix shown as M , where each entry of M (i.e., m_{ij}) indicates the rating user i has provided to location j . Since users visit fewer places than those available on a given LBSN, most entries for M are vacant (i.e., sparse). Therefore, in location recommender systems, the problem is: given the sparse matrix of user ratings, how to predict users' next unvisited POI. To address this problem, LSC leverages users' check-ins, demographics, locations, and friendship networks of users to develop a three-layered spectral clustering POI recommendation, as depicted in Fig. 1.

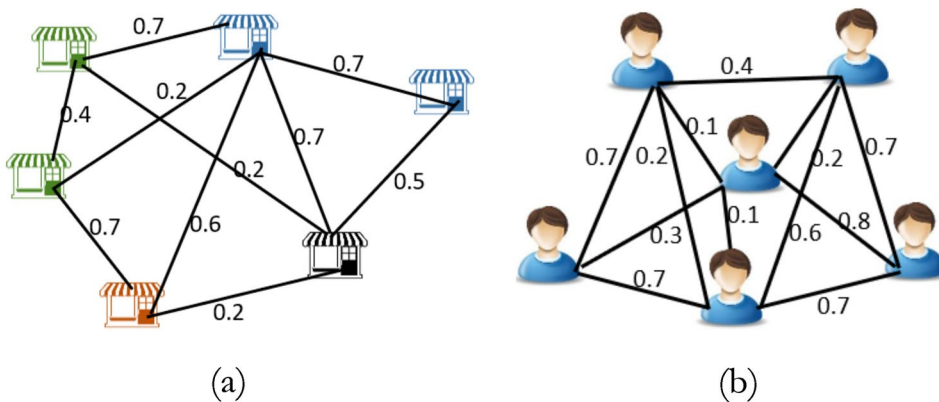
In this model, Layer 1 (at the top) comprises the network of friends of various users, Layer 2 contains the cluster of locations, and Layer 3 contains the cluster of users, grouped based on their demographic profile similarity. Clusters located on different layers are connected using weighted edges, indicating the strength between two given clusters.

For instance, an edge with strength of 0.6 between cluster l (on the second layer) and cluster u (the third layer), indicates that users in cluster u are 60 percent likely to be interested in locations inside cluster l . This model has been built on Yelp dataset, that incorporates spectral clustering inside layers two and three. Details on spectral clustering and how it is used are outlined in Sects. 3.2, 3.3.

3.2 Spectral clustering

The LSC location recommender employs spectral clustering to cluster users and locations on layers two and three. The middle layer (layer two) maintains clusters for locations, where an initial graph of locations is formed by connecting all locations to each other. Each connection is associated with a weight indicating the similarity between a pair of locations (see Fig. 2). The initial weight on each edge is determined using the Euclidean distance between two given locations. Similarly, an

Fig. 2 Initial network (graph) of **a** locations and **b** users



initial graph of users is created by connecting all users through weighted edges based on their Euclidean distance.

Spectral clustering is applied separately on each layer (i.e., locations and users) to create a cluster of locations/users. In spectral clustering, there is a graph containing vertices and edges shown with $G(V, E)$. This graph connects data points which is represented as an adjacency matrix A . Spectral clustering is a relaxation of the normalized cut problem ($Ncut$) defined as:

$$cut(B, \bar{B}) = \sum_{i \in B, j \in \bar{B}} A_{ij} \tag{1}$$

Here, \bar{B} and A_{ij} are the complement of B , and the similarity score between two given nodes of i and j , respectively. However, $Ncut$ tends to cut isolated sets rather than significant partitions since it increases with the number of edges. Consequently, $Ncut$ was modified as:

$$cut(B, \bar{B}) = \frac{cut(B, \bar{B})}{assoc(B, V)} + \frac{cut(\bar{B}, B)}{assoc(\bar{B}, V)} \tag{2}$$

Equation (2) penalizes the cut cost by the total connections from the nodes in B to all nodes V in the graph. Let y be the exact solution of $Ncut(B, \bar{B})$ with $y_i = 1$ if $i \in B$, and -1 otherwise. Then $Ncut(B, \bar{B})$ can be optimized as:

$$\min_x Ncut(B, \bar{B}) = \min_y \frac{y^T(D - A)y}{y^T D y} \tag{3}$$

where D and A are the degree and adjacency matrices respectively. To exactly solve $Ncut$, we have to look for two subsets with strong intra-connections and relatively weak weights between them. However, by relaxing y to take real values it can be minimized by solving the generalized eigenvalue system: $(D - A)y = \lambda D y$. The second smallest eigenvalue λ^L of the graph Laplacian $L = D - A$ and its corresponding eigenvector v^L , provide an approximation for

solving $Ncut$. When there is a partitioning between B and \bar{B} such that: $v_i^L = \begin{cases} \alpha, & i \in B \\ \beta, & i \in \bar{B} \end{cases}$. Then B and \bar{B} becomes the optimal $Ncut$ with a value of $Ncut(B, \bar{B}) = \lambda^L$. v^L is used to bipartition the graph then the following eigenvectors are used to partition the graph further. Clustering through graph Laplacian eigenvectors could be done iteratively (i.e., ordered by eigenvalues) or by constructing an embedding space using top eigenvectors.

The latter approach is more convenient and a well-known method for embedding space clustering using a symmetric graph Laplacian $L_{sym} = D^{-1/2} A D^{-1/2}$ where D and A are degrees and similarity matrices respectively. L_{sym} top eigenvectors constitute an embedding space in which points that are strongly connected will fall close to each other making clusters detectable by k -means. When it comes to spectral clustering, it is all about quantifying similarities. Ideally, points in the same cluster are linked by large weights so they can fall close in the embedding space. A naive approach of assigning weights would be through Euclidean distance. However, this is not a practical choice since it only considers first-order relationships. In first-order relationships, edges are drawn based on information from a pair of points only. A more practical approach would be considering second-order relationships where edges are drawn based on information from the neighbours. Section 3.3 describes how Spectral clustering has been used for POI recommendation.

3.3 The use of Spectral clustering for user POI recommendation

LSC is a POI recommender system designed to address data sparsity challenges in location recommendation. As depicted in Fig. 1, clusters on the top layer are formed using existing social connections among users (e.g., friendships), while the bottom layer contains clusters of users created based on their profile similarities. The middle layer also comprises clusters of locations, constructed based on the similarity of location

profiles (e.g., check-ins, ratings, category). LSC calculates the correlation (i.e., similarity strength) between every pair of clusters across layers 1 and 2, and layers 2 and 3, utilizing Mutual Information (MI) as shown in Eq. (4). The degree of correlation between a pair of clusters signifies the suitability of a cluster of locations for a cluster of users.

$$MI(C_i, C_j) = - \sum_p P(C_i) \sum_i P(c_i^j | C_i) \log_2 \left(P(c_i^j | C_i) \right) \tag{4}$$

where, C_i and C_j are two given clusters, c_i^j is the j th member of cluster i , and c_j^i is the i th member of cluster j . The concept is to tally the occurrences of a specific pair (i.e., a location and user) appearing together in the dataset, indicating that the user has checked into that location. LSC comprises two parts: initialization and utilization, as elaborated in the subsequent section.

3.3.1 Initialisation step

In the initialization step, as outlined in Algorithm 1, LSC constructs a network of locations by connecting all nodes and assigning a similarity degree to each edge (refer to Fig. 2). This network is then utilized with spectral clustering to identify clusters of locations (Lines 1–4). Similarly, spectral clustering is applied to the network of users to establish their similarity clusters. Users with friendship connections are grouped into the same cluster to form friendship clusters (Lines 5 and 6). Following the training

of clusters in the three-layered model, the subsequent step involves calculating the correlation among clusters in both intra and inter layers. Intra-layer correlation pertains to clusters located on the same layer, indicating the similarity between two given sets of clusters (Lines 10–12 in Algorithm 1). On the other hand, inter-layer correlation pertains to clusters situated on different layers, denoting the likelihood of users in one cluster showing interest in locations within adjacent clusters. This calculation is performed using Eq. (4) (Lines 14–17 in Algorithm 1). Upon initialization, the three-layer model is then utilized for POI recommendation in the utilization phase.

3.3.2 Utilisation step

During the utilization phase, as illustrated in Fig. 3 and Algorithm 2, the process of recommending locations to users unfolds in three steps. In the first step, an active user—defined as an online user seeking a POI—is clustered. This clustering hinges on social connections (appearing on the top layer) and the user's profile (appearing on the bottom layer). Here, a user profile encompasses demographic information and check-in sequences, alongside the scores and comments provided by a user for various places. Concurrently, clustering social connections aims to discern a cluster of friends and followers on the first layer.

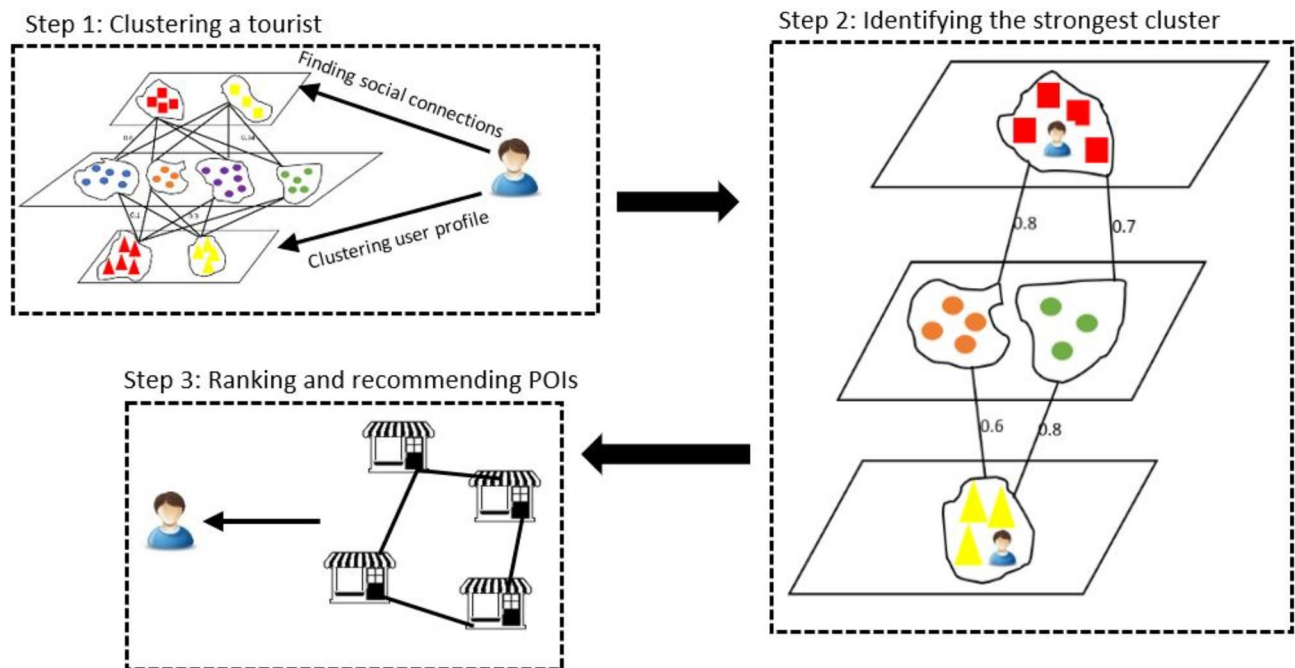


Fig. 3 Steps involved in LSC's utilisation phase

Algorithm 1: Initialisation of Location-based Spectral Graph Clustering

Input:
 M_S : Sparse user-rating matrix
 C : User check-in matrix (created using only training data)

Output:
 $LSGC$: a trained three-layer cluster model for location recommendation
 M'_S : Factorized (completed) user-rating matrix

Algorithm:

- 1 $N_L \leftarrow$ Randomly create an initial network of locations
- 2 $N_F \leftarrow$ Network of users' friendship
- 3 $N_S \leftarrow$ Network of users' similarity

- 4 $C_F^S \leftarrow$ Create a cluster of user friends based on N_F
- 5 $C_S^S \leftarrow$ Apply spectral clustering on N_S
- 6 $C_L^S \leftarrow$ Apply spectral clustering on N_L

- 7 $L_1 \leftarrow$ place N_S on the first layer of LSC
- 8 $L_2 \leftarrow$ place N_L on the middle layer of LSC
- 9 $L_3 \leftarrow$ place N_F on the third layer of LSC

- 10 $COR_{L_1} \leftarrow$ Calculate strength between every pair of clusters on L_1
- 11 $COR_{L_2} \leftarrow$ Calculate strength between every pair of clusters on L_2
- 12 $COR_{L_3} \leftarrow$ Calculate strength between every pair of clusters on L_3

- 14 **for** every cluster i on layer L_1
- 15 $COR_{L_1}^{L_2} \leftarrow$ measure correlation strength between i th cluster and j th cluster on L_2

- 16 **for** every cluster i on layer L_3
- 17 $COR_{L_3}^{L_2} \leftarrow$ measure correlation strength between i th cluster and j th cluster on L_2

- 18 LSC \leftarrow aggregate ($COR_{L_1}, COR_{L_2}, COR_{L_3}, COR_{L_3}^{L_2}, COR_{L_1}^{L_2}$)

In the second step, using the clusters identified in the first step, all locations within clusters showing the strongest connectivity to the user's cluster are extracted. This entails selecting locations from clusters that meet the strength threshold—where their strength to the user cluster exceeds a predefined threshold. Following this, in the third step, the extracted results are ranked based on similarity strength and recommended to the active user. The ranking process is governed by Eq. (5).

$$score_{l_i} = S_c * sim(c^i, v_p) \tag{5}$$

where S_c is a cluster of locations, with strength higher than a given threshold, c^i is a location in the cluster S_c and v_p is a place previously visited by the active user. If a user has no previous visiting history, then $sim(c^i, v_p) = 1$ and therefore score of locations will be calculated only based on connectivity strength (i.e., S_c).

Algorithm 2: Utilisation of Location-based Spectral Graph Clustering

Input:

LSC : Trained three-layer model
 U_n : Vector of user profile and check-in history
 t : threshold to include location clusters for recommendation

Output:

L_R : Set of recommended locations

Algorithm:

$U_i^c \leftarrow$ Cluster user U_n
 $L_i^c \leftarrow$ Identify clusters with strengths larger than t and extract locations in cluster with strongest connection to U_i^c

$U_f^c \leftarrow$ Identify the cluster friends for user U_n
 $L_f^c \leftarrow$ Identify clusters with strengths larger than t and extract locations in cluster with strongest connection to U_f^c

$$L_R \leftarrow L_i^c \cap L_f^c$$

3.3.3 Resolving cold start issue

LSC's three-layer model provides a solution to the cold start problem by suggesting locations to users with no check-in history. This process involves clustering users based on their demographic information and utilizing a cluster of locations with the highest strength to extract potential recommendations. These recommended locations are then ranked using Eq. (5). Similarly, for new locations without user check-ins, clustering based on their location profile allows them to be presented as

candidate recommendations to users. In Algorithm 3, LSC addresses data sparsity by predicting users' ratings for unvisited locations. This entails three steps: first, extracting the list of unvisited locations for each user; second, clustering users based on demographic and profile information; and third, clustering the unvisited locations, computing the average of existing location ratings in the corresponding cluster, and considering this average as the rating the user may provide for the unvisited location.

Fig. 4 Location clustering evaluation using Davis metric

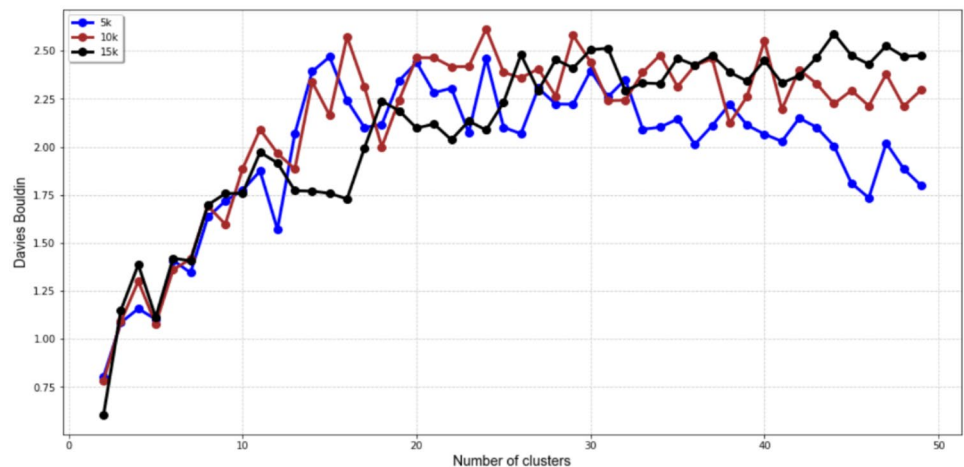


Fig. 5 Users clustering evaluation using Silhouette metric

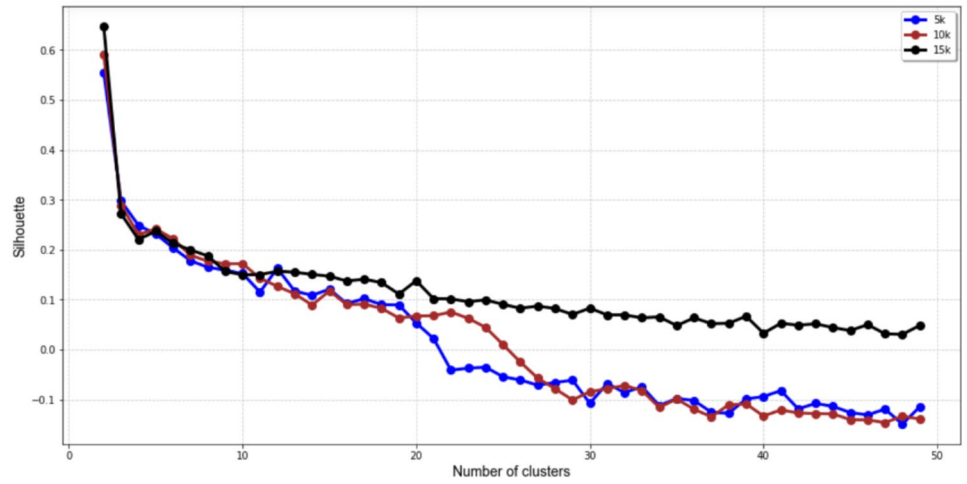


Fig. 6 Location clustering evaluation using Silhouette metric

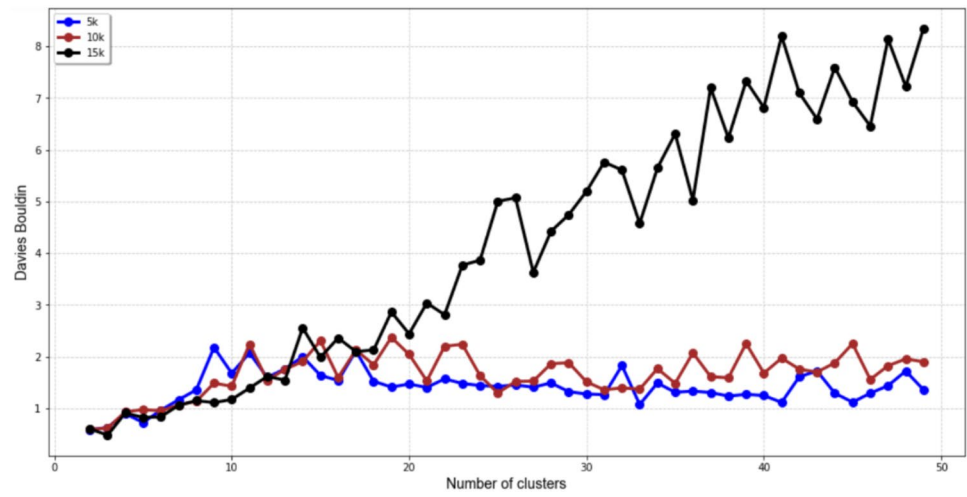
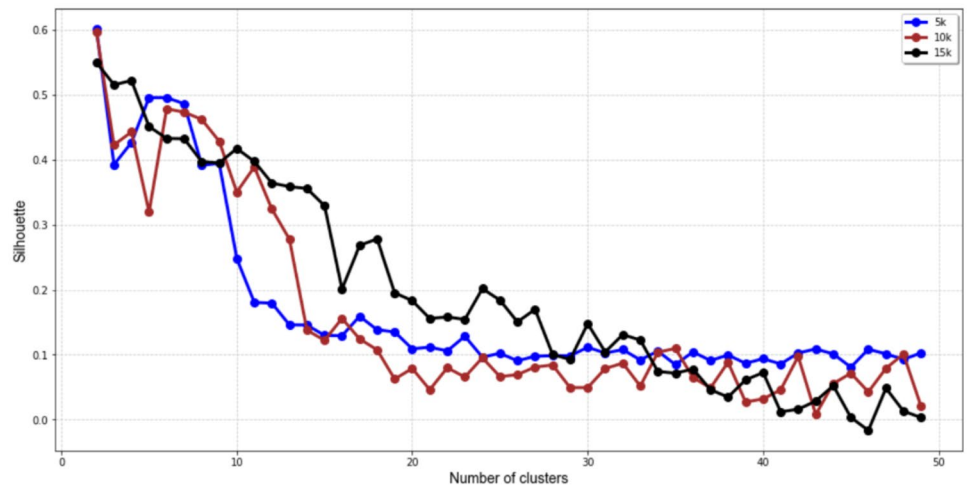


Fig. 7 Users clustering evaluation using Davis metric



Algorithm 3: resolving data sparsity in LSC

Input:

LSC : Trained three-layer model
 U : list of users
 L : list of all locations
 M : user-rating matrix

Output:

Predicted ratings

Algorithm:

```

for each user  $i$  in set  $U$ 
   $L_i \leftarrow$  extract unvisited locations in matrix  $M$  for  $U_i$ 
   $U_i^C \leftarrow$  Cluster user  $U_i$ 
  for each unvisited location  $j$  in  $L_i$ 
     $L_i^C \leftarrow$  cluster location  $L_i^j$ 
     $avg_i^C \leftarrow$  take the average of locations in cluster  $L_i^C$ 
     $score_i^j \leftarrow avg_i^C$ 
  end foreach
end foreach

```

4 Experiments and discussions

This section evaluates the performance of LSC and compares it against other rival algorithms using the Yelp dataset in terms of Root Mean Square Error (RMSE), Accuracy, and F1-score. The Yelp dataset is publicly available and chosen for this evaluation due to its suitability, as it provides information about users' friendship network, location

profiles (e.g., opening hours, on-spot car park availability), and users' profiles (e.g., age, gender, location). Unlike datasets such as Foursquare and Gowalla, which only offer information on users' check-ins (where and when), the Yelp dataset aligns with the requirements of LSC. The experiments are conducted using three different settings for the number of users: 5 k, 10 k, and 15 k. This variation aims to examine the impact of varying user sizes on LSC location

Table 1 LSC performance in terms of accuracy and F1 score

	Mutual Infor	Minimum Score	Without friendship network			With friendship network		
			Accuracy	F1 score	# users	Accuracy	F1 score	# users
0.5	0.5	0.5	0.955133	0.914366	4062	0.9864	0.975504	3593
	0.6	0.6	0.943	0.888628	4062	0.9864	0.975504	3411
	0.7	0.7	0.891	0.76294	4049	0.985533	0.973903	2631
	0.8	0.8	0.870867	0.706293	4030	0.984267	0.971553	2329
0.6	0.5	0.5	0.955133	0.914366	4062	0.9864	0.975504	3593
	0.6	0.6	0.943	0.888628	4062	0.9864	0.975504	3411
	0.7	0.7	0.891	0.76294	4049	0.985533	0.973903	2631
	0.8	0.8	0.870867	0.706293	4030	0.984267	0.971553	2329
0.7	0.5	0.5	0.956333	0.916847	838	0.771467	0.32837	3611
	0.6	0.6	0.944467	0.891804	832	0.771067	0.326403	3433
	0.7	0.7	0.9306	0.861033	766	0.766667	0.304452	3225
	0.8	0.8	0.886867	0.751719	369	0.7402	0.159223	2569
0.8	0.5	0.5	0.956333	0.916847	880	0.774267	0.342013	3611
	0.6	0.6	0.944467	0.891804	880	0.774267	0.342013	3433
	0.7	0.7	0.9306	0.861033	825	0.7706	0.324101	3225
	0.8	0.8	0.886867	0.751719	654	0.7592	0.265854	2569

recommendation. LSC utilizes friendship data to generate user communities on Layer 1. Thus, a community detection algorithm available in Python (known as CDlib) is applied to the friendship data to generate communities.

In the next section, the results obtained by spectral clustering are presented, with the aim of determining the optimal number of clusters for location and user clustering. Section 4.2 evaluates LSC recommendation performance in terms of accuracy and F1-score in two settings: with and without consideration of the friendship network. Additionally, Sect. 4.3 utilizes RMSE to compare LSC against state-of-the-art algorithms proposed for location recommendation. The execution time for LSC is discussed in Sect. 4.4.

4.1 Cluster tuning

LSC utilizes spectral clustering to create clusters of users and locations on layers two and three of the three-layered model, respectively. To assess the scalability of the model, we extracted datasets containing 5 k, 10 k, and 15 k users for experimentation, as illustrated in Figs. 4–5. We employed the elbow rule to determine the optimal number of clusters. As observed in Figs. 6 and 5, as the number of users increases from 5 to 10 k and then 15 k, the Silhouette measure decreases. This decline occurs because more user information leads to better training of the spectral clustering algorithm, resulting in improved outcomes. Similarly, in Figs. 4 and 7, the Davies Bouldin Index (DB) rises. However, in both the DB and Silhouette metrics, beyond a certain number of clusters (i.e., 20 clusters), the magnitude of change in these metrics becomes insignificant. Hence, the best number of clusters is 20 for locations and users.

4.2 Recommendation performance

Users on Location-Based Social Networks (LBSNs) search for suitable places to check-in. Upon checking in, they rate their experience and provide comments. This section's experiment evaluates LSC's performance in recommending places in two settings: with and without consideration of the friendship network. In Table 1, MI represents the threshold used to select a subset of candidate clusters (refer to Fig. 3). The "Score" denotes the satisfaction score (i.e., rating) provided by users for a location. For instance, a score of 0.5 suggests that candidate locations with a score of 5 out of 10 are recommended. Users check into various places and rate the venues to express their satisfaction. However, satisfaction criteria vary among users. While some users might be satisfied with venues rated an average of 3 out of 5, others may prefer venues with a rating of 4.5 or higher. One of the goals of location recommendation in LSC is to suggest places that users are highly likely to find satisfying.

Therefore, the "Score" column indicates the minimum threshold for considering a user satisfied. The "# users" column indicates the number of users whose average rating equals or exceeds the value in the "Score" column.

As indicated in Table 4, increasing the Mutual Information (MI) threshold results in fewer users being short-listed. LSC evaluates these users to assess their similarity with the current online user and makes recommendations based on the locations visited by other similar users (refer to Sect. 3.3). Consequently, LSC's criteria for shortlisting clusters of users become stricter, leading to a reduced number of selected users. Incorporating friendship information has led to improvements in accuracy and F1-score measures. This enhancement is attributed to LSC's utilization of friendship network information to refine the results. Specifically, LSC considers the locations visited by a user's friends and includes them in the list of candidate locations if they meet the minimum score requirement.

4.3 Comparisons

This section compares the performance of LSC against five other rival algorithms. The selection of articles for comparison is based on their relevance to location recommendation, novelty, and whether they utilize friendship networks. Priority is given to articles that use the Yelp dataset as their primary dataset to maintain consistency with LSC's evaluation criteria. A brief explanation for each compared algorithm has been given below.

- **Local Geographical based Logistic Matrix Factorization (LGLMF):** This algorithm proposed for POI Recommendation (Rahmani et al. 2019). LGLMF stands as an efficient geographical model which takes into account the user's primary region of activity along with the significance of each location within that region. This local geographical model is then integrated into the Logistic Matrix Factorization to enhance the precision of the POI recommendation. LGLMF utilizes geographical data to encompass both the user's individual geographical profile and the popularity of a location in terms of geography. It seamlessly incorporates this geographical model into matrix factorization methodologies.
- **Joint Geographical and Temporal Modelling based on Matrix Factorization (STACP):** This is another algorithm proposed for Point-of-Interest Recommendation (Rahmani et al. 2020). The authors emphasize the necessity of integrating contextual information, including geographical and temporal influences, to enhance POI recommendation and tackle the data sparsity issue. Therefore, they delve into the spatio-temporal activities of users to create a more precise model of their behavior.

Table 2 RMSE comparisons across various location recommenders

User/location	0.5	0.6	0.7	0.8	0.9	1
LSC algorithm with(without) friendship						
0.5	0.14(0.37)	0.14(0.37)	0.14(0.37)	0.14(0.37)	0.14(0.37)	0.14(0.37)
0.6	0.14(0.37)	0.14(0.37)	0.14(0.37)	0.14(0.37)	0.14(0.37)	0.14(0.37)
0.7	0.14(0.44)	0.14(0.44)	0.14(0.44)	0.14(0.44)	0.14(0.44)	0.14(0.44)
0.8	0.14(0.44)	0.14(0.44)	0.14(0.44)	0.14(0.44)	0.14(0.44)	0.14(0.44)
0.9	0.14(0.44)	0.14(0.44)	0.14(0.44)	0.14(0.44)	0.14(0.44)	0.14(0.44)
1	0.14(0.48)	0.14(0.48)	0.14(0.48)	0.14(0.48)	0.14(0.48)	0.14(0.48)
LGLMF						
0.5	0.34	0.34	0.34	0.34	0.34	0.34
0.6	0.34	0.34	0.34	0.34	0.34	0.34
0.7	0.34	0.34	0.34	0.34	0.34	0.34
0.8	0.33	0.33	0.33	0.33	0.33	0.33
0.9	0.31	0.31	0.31	0.31	0.31	0.31
1	0.31	0.31	0.31	0.31	0.31	0.31
STACP						
0.5	0.34	0.34	0.34	0.34	0.34	0.34
0.6	0.33	0.33	0.33	0.33	0.33	0.33
0.7	0.33	0.33	0.33	0.33	0.33	0.33
0.8	0.33	0.33	0.33	0.33	0.33	0.33
0.9	0.32	0.32	0.32	0.32	0.32	0.32
1	0.31	0.31	0.31	0.31	0.31	0.31
Gaussian						
0.5	0.34	0.34	0.34	0.34	0.34	0.34
0.6	0.34	0.34	0.34	0.34	0.34	0.34
0.7	0.33	0.33	0.33	0.33	0.33	0.33
0.8	0.33	0.33	0.33	0.33	0.33	0.33
0.9	0.31	0.31	0.31	0.31	0.31	0.31
1	0.31	0.31	0.31	0.31	0.31	0.31
Looker (Missaoui, et al. 2019)						
0.5	0.24	0.24	0.24	0.24	0.24	0.24
0.6	0.24	0.24	0.24	0.24	0.24	0.24
0.7	0.23	0.23	0.23	0.23	0.23	0.23
0.8	0.23	0.23	0.23	0.23	0.23	0.23
0.9	0.21	0.21	0.21	0.21	0.21	0.21
1	0.21	0.21	0.21	0.21	0.21	0.21
Gradient Descent (Forsati et al. 2014)						
0.5	0.14	0.12	0.12	0.12	0.12	0.12
0.6	0.11	0.11	0.12	0.11	0.11	0.11
0.7	0.1	0.12	0.11	0.1	0.1	0.1
0.8	0.11	0.11	0.1	0.11	0.11	0.11
0.9	0.1	0.11	0.1	0.1	0.1	0.1
1	0.1	0.11	0.1	0.1	0.1	0.1

- Gaussian based location recommendation (Cheng et al. 2012), fuses matrix factorization with geographical and social influence for POI recommendation in LBSNs. The proposed algorithm captures geographical influence by modelling the probability of a user's check-in at a location using a Multi-centre Gaussian Model (MGM). It then incorporates social information and integrates geographical influence into a generalized matrix factorization framework.
- LOOKER (Missaoui, et al. 2019): This algorithm employs content-based filtering (CBF) based on a multi-layer user profile. Each layer represents different categories of travel-related services (e.g., restaurants, hotels, points of interest), modelled using language models

Table 3 LSC execution time (minutes) MI=0.5, Score=0.5

% of location	% of User					
	0.5		0.7		0.95	
	With	Without	With	Without	With	Without
25	17	7	25	17	34	19
50	20	10	29	20	38	22
75	23	15	32	22	43	24
100	25	19	36	26	47	27

Table 4 LSC execution time (minutes) MI=0.8, Score=0.8

% of Location	% of User					
	0.5		0.7		0.95	
	With	Without	With	Without	With	Without
25	26	12	25	22	25	21
50	27	15	27	23	30	26
75	28	17	29	24	39	29
100	30	20	31	26	47	34

Table 5 LSC execution time in terms of minutes compared with rival algorithms (considering all locations rounded up to the closest whole number)

Algorithms	Ratio of users utilised in experiments					
	0.45	0.55	0.65	0.75	0.85	0.95
LSC (without)	18	20	23	26	32	35
LSC (with)	26	30	31	31	40	45
LGLMF	30	30	33	34	41	43
STACP	32	32	33	34	45	45
Gaussian	30	30	33	34	41	43
LOOKER	22	22	23	27	30	31
Gradient Descent	18	20	23	26	32	35

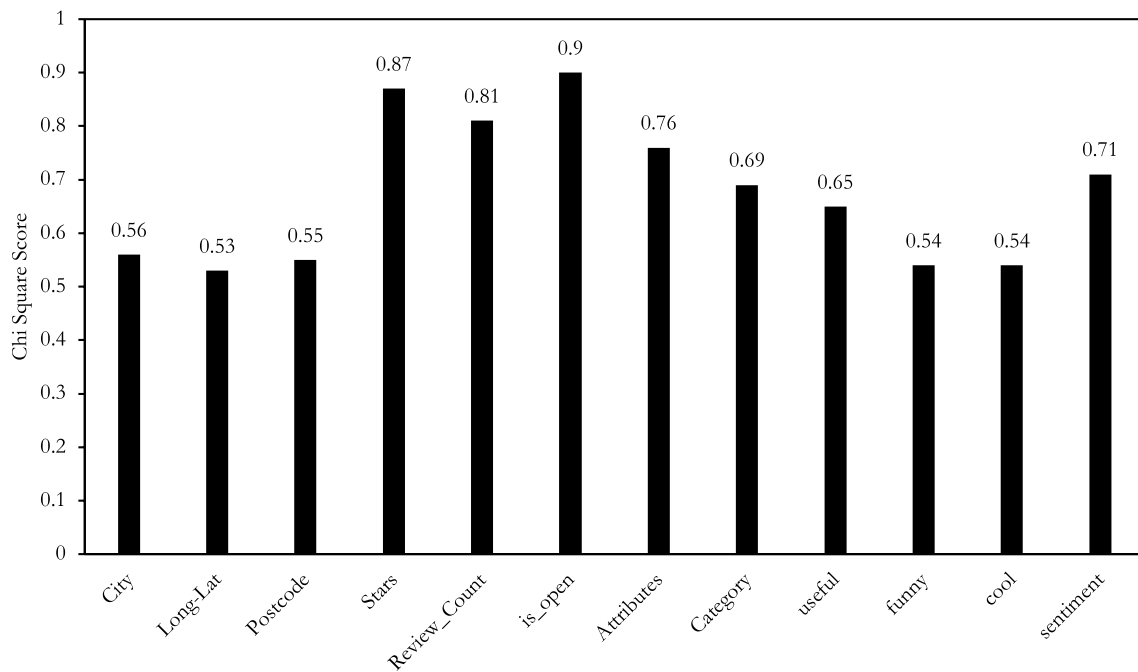
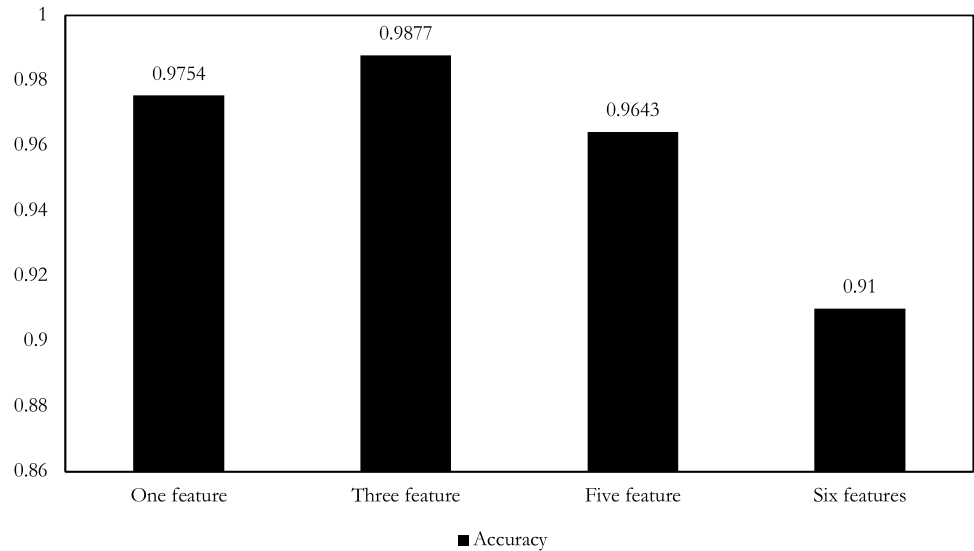
Table 6 Statistical non-parametric sign test

	LGLMF	STACP	Gaussian	LOOKER	Gradient	Z-ref
LSC (friendship)	6.17	6.17	6.17	6.17	-6.09	1.96

defined based on captured User-Generated Content (UGC). This approach enables the inference of travelers' interests and opinions regarding the available items.

- Gradient Descent based POI recommender (Forsati et al. 2014): This model is a matrix factorization-based approach for recommendation in social rating networks. It effectively integrates both trust and distrust relationships to enhance the quality of recommendations and alleviate issues related to data sparsity and cold-start problems.

The results for LSC in Table 2 are presented as X(Y), where X denotes the RMSE value and Y represents the mutual information value between a pair of clusters. LSC, in both settings of 'with' and 'without' friendship networks, has demonstrated superior performance compared to rival algorithms. In evaluating these algorithms, the one with the lowest RMSE is considered the best. Comparisons are conducted across different rates of user participation and available locations to understand how varying data availability impacts LSC's location recommendation compared to rival algorithms.

Fig. 8 Ablation study conducted on Yelp dataset**Fig. 9** Chi-Square scores on spatial features (normalized values)

LSC has exhibited superior performance compared to rival algorithms, with its performance being comparable to the Gradient Descent algorithm. This superiority can be attributed to the three-layered model incorporating spectral clustering. Users and their past visiting patterns are effectively linked through clusters and mutual information between user-location clusters and users within friendship communities. Moreover, spectral clustering proves beneficial as it extracts insights from complex networks by identifying non-convex shape clusters and encoding pairwise similarity into an adjacency matrix. Notably, inclusion of friendship

information has led to a reduction in the error rate. This is because friends are likely to exhibit similar behavior and traits, thus making recommendations based on friends' visits more likely to be favorable.

4.4 Empirical analysis

This section reports the result of several empirical analyses to better understand LSC's performance. These empirical analyses discuss the execution time of LSC and rival algorithms (Sect. 4.1), report the results of a non-parametric

statistical test (Sect. 4.2) and finally discuss an ablation study to investigate how various spatial features can affect LSC's performance (Sect. 4.3).

4.4.1 Execution time

This section presents the execution time for LSC location recommendation, focusing on the time required to build the three-layered training model (i.e., overhead time). In Tables 3, 4, the first column denotes the ratio of selected locations, while the remaining columns represent the ratio of users utilized for constructing the three-layered model. Columns labeled as 'with' or 'without' refer to the results obtained when considering and not considering the friendship network, respectively.

The results displayed in Tables 3, 4 reveal that increasing the number of locations, from 25 to 100, leads to a slight increase in execution time (approximately four minutes). However, a notable increase in execution time is observed when augmenting the number of users from 50 to 95 percent (i.e., the last two columns). This indicates that the number of users has a greater impact on the algorithm's execution time compared to the number of locations. This is attributed to the fact that within the three-layered model, the first and third layers directly or indirectly contain information about users (or their friendship network), whereas only one layer (the middle layer) holds information about locations.

Table 5 compares the overhead execution time of LSC with/without friendship against its rival algorithms (with $MI = 0.65$ and $Score = 0.8$ for LSC). The results indicate that LSC is comparable to the rival algorithms in terms of execution time.

4.4.2 Statistical test

We conducted a statistical non-parametric sign test (Triola 2001) on the RMSE values presented in Table 2 to demonstrate the statistical significance of LSC over the baseline techniques. We performed a right-tailed sign test for the significance level $\alpha = 0.05$, i.e., 95% significance level. We present the test statistics (z-value) for LSC compared with the baseline techniques in Table 6. The z-ref value is shown in the last column of the table. To be considered significantly better than the baseline techniques, LSC would need to have a z-value greater than the z-ref value. For $\alpha = 0.05$, the z-ref value is 1.96. From Table 6, we can see that LSC's results are significantly better than LGLMF, STACP, Gaussian and Looker for a significance level $\alpha = 0.05$, i.e., 95% significance, except for Gradient Descent.

4.4.3 Ablation study

This section discusses the impact of various spatial features on recommendation accuracy. In the dataset, we utilize 12

spatial features in total. To execute the algorithm, we determine the best setting for Mutual Information and minimum score based on the results reported in Table 1. Hence, we use LSC with a friendship network where MI and minimum score are both set to 0.6.

In this ablation study, we systematically exclude a given number of spatial features at a time to assess their impact on recommendation accuracy, i.e., whether recommendation improves or deteriorates. This process is outlined in Fig. 8. We calculate the chi-square value for each spatial feature and then exclude them from the set of features based on their importance to re-run LSC and obtain accuracy results. Figure 8 illustrates the removal of features based on chi-square results depicted in Fig. 9. The features are sorted in descending order, and then the top one, three, five, and six features are excluded from the spatial features. As evident in Fig. 8, removing features has varying impacts on recommendation accuracy, with some features improving accuracy while others deteriorate it.

5 Conclusion

Location-based recommender systems play a crucial role in assisting users throughout their tourism journey, from planning and booking to exploring new places. However, recommending suitable locations from a vast pool can be challenging. This study addresses this challenge by introducing a novel location recommender system tailored for the Yelp dataset, leveraging spectral clustering and a three-layered recommendation model. Spectral clustering proves invaluable for uncovering complex network structures and encoding pairwise similarities, forming the foundation of our approach. The proposed algorithm, termed LSC, trains a three-layered model using Yelp data, employing a community detection algorithm to organize users' friends into communities and clustering to group users and locations effectively.

Experimental evaluations demonstrate LSC's effectiveness, showcasing improvements in recommendation accuracy, F1-Score, and RMSE. Comparison with five state-of-the-art recommender systems further highlights the superiority of LSC, especially with the incorporation of the friendship layer. Future research directions include exploring alternative clustering algorithms for both user and location layers, testing various community detection algorithms for the friendship network layer, and incorporating Twitter data to enhance recommendations based on user tweets. Additionally, extending LSC to handle power-law patterns in friendship networks could further enhance recommendation quality by addressing imbalanced friendship data.

Author contributions A.M.: Conceptualisation, write up, experimental design and analysis H.J.: Conceptualisation, write up, experimental design and analysis M.A.R.: Write up, and proofreading K.L.O.: Write up, and proofreading.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions.

Data availability We have used publicly available Yelp dataset and will be provided upon request.

Declarations

Conflict of interest The authors declare no competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Alshammari M, Stavrakakis J, Takatsuka M (2021) Refining a k-nearest neighbor graph for a computationally efficient spectral clustering. *Pattern Recogn* 114:107869
- Alves P, Martins H, Saraiva P, Carneiro J, Novais P, Marreiros G (2023) Group recommender systems for tourism: how does personality predict preferences for attractions, travel motivations, preferences and concerns? *Marreiros*, g 33(5):1141–1210
- Bai L, Qi M, Liang J (2023) Spectral clustering with robust self-learning constraints. *Artif Intell* 320:103924
- Behera G, Nain N (2023) The state-of-the-art and challenges on recommendation system's: principle, techniques and evaluation strategy. *SN Comp Sci* 4(5):677
- Cai W, Wang Y, Lv R, Jin Q (2019) An efficient location recommendation scheme based on clustering and data fusion. *Comput Electr Eng* 77:289–299
- Canturk D, Karagoz P (2021) SgWalk: location recommendation by user subgraph-based graph embedding. *IEEE Access* 9:134858–134873
- Canturk D, Karagoz P, Kim S, Toroslu I (2023) Trust-aware location recommendation in location-based social networks: a graph-based approach. *Expert Syst Appl* 213:119048
- Cheng C, Yang H, King I, Lyu M (2012) Fused matrix factorization with geographical and social influence in location-based social networks. *AAAI Conf Artif Intell* 26:17–23
- Deng J, Huang D, Ding Y, Zhu Y, Jing B, Zhang B (2023) Subsampling spectral clustering for stochastic block models in large-scale networks. *Comput Stat Data Anal* 189:107835
- Divyaa L, Pervin N (2019) Towards generating scalable personalized recommendations: integrating social trust, social bias, and geo-spatial clustering. *Decis Support Syst* 122:113066
- Dokuz A, Celik M (2017) discovering socially important locations of social media users. *Expert Syst Appl* 86:113–124
- Farahani M, Torkestani J, Rahmani M (2023) Dynamic user profile for adaptive personalized recommender system using learning automata. *Multimed Tools Appl*. <https://doi.org/10.1007/s11042-023-17339-w>
- Forsati R, Mahdavi M, Shamsfard M, Sarwat M (2014) Matrix factorization with explicit trust and distrust side information for improved social recommendation. *ACM Trans Info Syst (TOIS)* 32(4):1–38
- Gao R, Li J, Li X, Song C, Zhou Y (2018) A personalized point-of-interest recommendation model via fusion of geo-social information. *Neurocomputing* 273:159–170
- Ge Y, Peng P, Lu H (2021) Mixed-order spectral clustering for complex networks. *Pattern Recogn* 117:107964
- Guo J, Zhang W, Fan W, Li W (2018) Combining geographical and social influences with deep learning for personalized point-of-interest recommendation. *J Manag Inf Syst* 34(4):1121–1153
- Han L, Luo W, Yang A, Zheng Y, Lu R, Lai J, Cheng Y (2023) Fully privacy-preserving location recommendation in outsourced environments. *Ad Hoc Netw* 141:103077
- Iqbal M, Ghazanfar M, Sattar A, Maqsood M, Khan S, Mehmood I, Baik S (2019) Kernel context recommender system (KCR): a scalable context-aware recommender system algorithm. *IEEE Access* 7:24719–24737
- Jiao X, Xiao Y, Zheng W, Wang H, Hsu C (2019) A novel next new point-of-interest recommendation system based on simulated user travel decision-making process. *Futur Gener Comput Syst* 100:982–993
- Kefalas P, Symeonidis P, Manolopoulos Y (2018) Recommendations based on a heterogeneous spatio-temporal social network. *World Wide Web* 21:345–371
- Khan I, Sadad A, Ali G, ElAffendi M, Khan R, Sadad T (2023a) NPR-LBN: next point of interest recommendation using large bipartite networks with edge and cloud computing. *J Cloud Comp* 12(1):54
- Khan S, Khan O, Azam N, Ullah I (2023b) Improved spectral clustering using three-way decisions. *Inf Sci* 641:119113
- Kolahkaj M, Harounabadi A, Nikravanshalmani A, Chinipardaz R (2020) A hybrid context-aware approach for e-tourism package recommendation based on asymmetric similarity measurement and sequential pattern mining. *Electron Commer Res Appl* 42:100978
- Liu S, Wang L (2018) A self-adaptive point-of-interest recommendation algorithm based on a multi-order markov model. *Futur Gener Comput Syst* 89:506–514
- Ma Y, Mao J, Ba Z, Li G (2020) Location recommendation by combining geographical, categorical, and social preferences with location popularity. *Inf Process Manage* 57(4):102251
- Manotumrukta J, Macdonald C, Ounis I (2020) A contextual recurrent collaborative filtering framework for modelling sequences of venue checkins. *Inf Process Manage* 57(6):102092
- Missaoui S, Kassem F, Viviani M, Agostini A, Faiz R, Pasi G (2019) LOOKER: a mobile, personalized recommender system in the tourism domain based on social media user-generated content. *Pers Ubiquit Comput* 23:181–197
- Mohammadi N, Rasoolzadegan A (2022) A two-stage location-sensitive and user preference-aware recommendation system. *Expert Syst Appl* 191:116188
- Panda D, Ray S (2022) Approaches and algorithms to mitigate cold start problems in recommender systems: a systematic literature review. *J Intell Info Syst* 59(2):341–366
- Rahimi S, Far B, Wang X (2020) Behavior-based location recommendation on location-based social networks. *GeoInformatica* 24:477–504

- Sarkar J, Majumder A, Panigrahi C, Roy S, Pati B (2023) Tourism recommendation system: a survey and future research directions. *Multimed Tool Appl* 82(6):8983–9027
- Sheibani S, Shakeri H, Sheibani R (2023) Four-dimensional trust propagation model for improving the accuracy of recommender systems. *J Supercomput* 79(15):16793–16820
- Si Y, Zhang F, Liu W (2019) An adaptive point-of-interest recommendation method for location-based social networks based on user activity and spatial features. *Knowl-Based Syst* 163:267–282
- Sojahrood Z, Taleai M (2021) A POI group recommendation method in location-based social networks based on user influence. *Expert Syst Appl* 171:114593
- Tuan C, Hung C, Wu Z (2017) Collaborative location recommendations with dynamic time periods. *Pervasive Mob Comput* 35:1–14
- Wang W, Chen J, Wang J, Chen J, Liu J, Gong Z (2019) Trust-enhanced collaborative filtering for personalized point of interests recommendation. *IEEE Trans Industr Inf* 16(9):6124–6132
- Xiong XQ (2020) A point-of-interest suggestion algorithm in multi-source geo-social networks. *Eng Appl Artif Intell* 88:103374
- Zhang H, Wei S, Hu X, Li Y, Xu J (2020) On accurate POI recommendation via transfer learning. *Distrib Parallel Database* 38:585–599
- Zhang X, Liu H, Wu X, Zhang X, Liu X (2021) Spectral embedding network for attributed graph clustering. *Neural Netw* 142:388–396
- Zhao G, Lou P, Qian X, Hou X (2020) Personalized location recommendation by fusing sentimental and spatial context. *Knowl-Based Syst* 196:105849
- Zhou Y, Yang G, Yan B, Cai Y, Zhu Z (2022) Point-of-interest recommendation model considering strength of user relationship for location-based social networks. *Expert Syst Appl* 199:117147
- Zhu J, Han L, Gou Z, Yang Y, Yuan X, Li J, Li S (2021) A robust personalized location recommendation based on ensemble learning. *Expert Syst Appl* 167:114065
- Rahmani H, Aliannejadi M, Ahmadian S, Baratchi M, Afsharchi M, Crestani F (2019) LGLMF: local geographical based logistic matrix factorization model for poi recommendation. *Asia Information Retrieval Symposium*, (pp. 66–78)
- Rahmani H, Aliannejadi M, Baratchi M, Crestani F (2020) Joint geographical and temporal modeling based on matrix factorization for point-of-interest recommendation. In: *Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020 Proceedings, Part I*. Springer International Publishing: Cham (pp. 205–219)
- Triola M (2001) *Elementary Statistics*. (Vol. 8). Addison Wesley Longman
- Zhou C, Peng J, Ma Y, Jiang Q (2021) A Privacy-preserving location recommendation scheme without trustworthy entity. In: *2021 IEEE 20th International conference on trust, security and privacy in computing and communications (TrustCom)*.(pp. 444–451)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.