A Recommender System Research Based on Location-Based Social Networks

Jianmin Wang[1], Ruhuo Tan[2(✉)], Ri-Peng Zhang[2], and Fang You[1]

[1] School of Arts and Media, Tongji University, Shanghai, China
{wangjianmin,youfang}@tongji.edu.cn
[2] School of Information Science and Technology, Sun Yat-Sen University, Guangzhou, China
{891567977,pk_mati}@qq.com

Abstract. Nowadays, with the rapid development of Location-Based Social Networks, information presents a trend of explosive growth. In order to locate the valuable information in tremendous amounts of location-based service data and prosperi O2O business through LBS, recommender system based on location-based service was presented. This paper takes Sina Microblog LBS data as research object. By analyzing the features of the crawled data and the existing problems of current LBS recommender systems, we present Region-density-based Clustering (RC) recommendation algorithm. For optimization, this paper also presents another algorithm called Distance-and-Category-based Clustering (DCC). This algorithm is mainly about clustering spots base on their distance similarity and category similarity. If two spots are nearby and both category attributes are similar, they will be more likely to gathered into a cluster. Finally, this paper also proposed the visualization method of the LBSNs recommender system.

Keywords: Location-based service · Sina microblog · Cluster · Location recommendation · Visualization

1 Introduction

With the rapid development of mobile Internet and mobile terminal location technology in recent years, location-based service (LBS), which has been applied to various applications, is becoming the standard configuration of mobile Internet applications. All kinds of LBS applications, which can be mainly categorized as entertainment, social networking, life service, and business service, are merged into social networking services (SNS), thereby changing the way people socialize. For instance, the traditional relationship of sociality is based on friends or some kind of stable social relation constructed by interests. However, through the LBS, we can find a user who is engaged in the same activity, in the same place, and at the same time. As a result, we build a new social relation based on location. Moreover, as a dimension of information filter, location can be used to improve the validity and accuracy of a user's fetching and sharing of information. Furthermore, relying on their significant relationship and interest spectra, location-based social networks (LBSNs) construct an online to offline (O2O) pattern [1]

G. Meiselwitz (Ed.): SCSM 2016, LNCS 9742, pp. 81–90, 2016.
DOI: 10.1007/978-3-319-39910-2_8

through releasing text, images, videos, and audios that possess location information. The O2O pattern improves the propagandist strength of a businessman as well as makes finding valuable locations easy for a user.

2 Background

As LBSNs develop, information overload appears inevitable because of the promotion of user-based and location information. To solve such a problem, a recommender system based on location service is proposed and is gradually becoming a popular subject of current studies. In current mainstream LBSNs, a user's location preferences are mainly measured by the number of check-ins. For example, Berjani and Strufe [2] established a user-site score matrix by crawling Gowalla's data, after which they gain results by using orthogonal matrix decomposition. Ying et al. [3] established UPOI-Mine, a recommender system driven by a prediction model based on a regression tree. Ye et al. [4] combined the geographic feature of social relation and location, and proposed the geo-measured friend-based collaborative filtering algorithm. While the above methods are only concerned with check-in records and ignore the semantic information between locations, which may easily lead to a problem of identifying similar locations, Lee and Chung [5] used semantic information of sites to compute the similarity of users for the first time. However, they neglected the effect of distance. Currently, research on LBSNs is mainly based on collaborative filtering [6], but the large data in LBSNs will result in sparsity of check-in matrix. Thus, Leung et al. [7] used community location model (CLM) to show the relations among users, activities, and locations. They used community-based agglomerative-divisive clustering to cluster CLM and reduce the sparsity. In addition, for a better universal method of reducing the dimension of user check-in matrix, Zhou et al. [8] proposed the use of PLSA topic model to determine the implicit subject between users and locations.

In LBSNs, each user's check-in information of location becomes easy to obtain because of the openness of such information. By analyzing different users' check-in behaviors, we can recommend users with similar location preferences and possible interesting locations, which makes finding valuable user and information in massive data effective for users as well as increases O2O business through location recommendation. SNS data are characterized by a large order of magnitude, authenticity, and instantaneity. Therefore, we believe it can reflect users' behavior effectively.

3 RC Algorithm

In this paper, clustering all the sites is an important part of the location recommendation algorithm because it can reduce the dimension and improve the recommendation accuracy. Clustering, which is mainly based on the similarity of data objects, involves dividing data objects with similar characteristics based on certain rules into different subsets, which are usually called clusters or groups. The goals of clustering are to make object similarity in a cluster as high as possible and make the object similarity among clusters as low as possible [9]. The common clustering algorithm can be divided into

partitioning methods [10], hierarchical methods [11], density-based methods [12], grid-based methods [13] and model-based methods.

The RC algorithm clusters sites based on the domain density according to the aggregation effect of geography so that it can reduce the dimension and avoid user similarity reduction because of check-in error. Subsequently, it calculates the signed location for each cluster from the clustering result and transfers a cluster that contains multiple locations into an abstract signed location with the geographic coordinate and hierarchical category. Then, it transfers the check-in vector of a user relative to the location into the check-in vector of the signed location of the location cluster and calculates the similarity among users. Finally, it selects users with Top-K similarity as a user recommendation, after which we use the recommended users for collaborative filtering calculation based on location preferences. We obtain the Top-N score locations as the location recommendation when we have finished adding the preference of the target user relative to this class of location to the calculation of unvisited sites.

Here i would like to introduce user similarity calculation based on hierarchical category tree. As shown in Fig. 1 we can construct a hierarchical tree based on the location cluster by transferring the category property to the location cluster. The bottom of the tree is the location cluster layer, while CategorySmall, CategoryMedium, and CategoryLarge are the small, middle, and large categories, respectively, to which the location cluster belongs. A high hierarchy corresponds to a large grain size and more location clusters. Constructing the hierarchical category tree improves the calculation of the similarity among users. For instance, user u_a always checked in location cluster c_1, whose signed location is Commercial Street of Beijinglu, while user u_b always checked in the

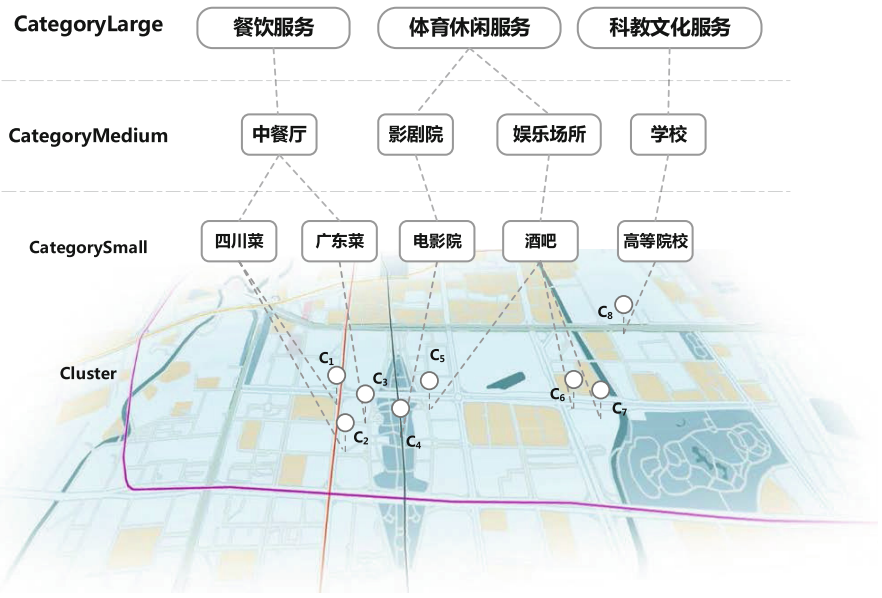

Fig. 1. Hierarchical categories tree based on location cluster

location cluster c_2, whose signed location is Commercial Street of Shangxiajiu; the abovementioned two users will have no similarity because they have not checked in the same location cluster if we consider the cluster layer only. However, if we consider the small category property of c_1 and c_2, then we will find that both c_1 and c_2 belong to the characteristic commercial walking street. Thus, users u_a and u_b possess a similarity in the CategorySmall layer. For the other hierarchy, if users do not possess a common check-in category in a hierarchy, then we could consider a higher hierarchy. If users have a similarity in the lower hierarchy, then their similarity is higher. For example, users u_a and u_b have common check-in location clusters in the cluster layer, while users u_a and u_c do not possess a similarity in the same layer. But if users u_a and u_c have a common check-in category in the CategorySmall layer, then the similarity between users u_a and u_b is higher than the similarity between users u_a and u_c. Therefore, to reduce the weight of hierarchy in the user similarity calculation, the hierarchy with a higher grain size should be multiplied by the corresponding coefficient.

4 DCC Algorithm

The RC algorithm contains different kinds of locations after clustering, and the category property of the signed location would cover the category property of other locations when using the signed location conversion algorithm. Therefore, the final check-in vector of the user to the location cluster cannot represent the true preference of the user in each category. To maintain the user's original location category preference when clustering the location, we propose the DCC algorithm. DCC algorithm clusters the locations that are near each other or have similar categories to improve the similarity among locations in the cluster. As shown in Figs. 2 and 3, while the RC algorithm can find only conglomerated location clusters, which are not overlapped in the geographical space, the DCC algorithm can find the crossed location clusters, thereby obtaining higher flexibility in location clustering.

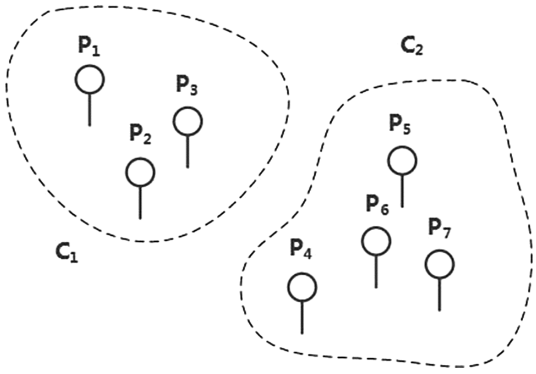

Fig. 2. location cluster result of RC algorithm

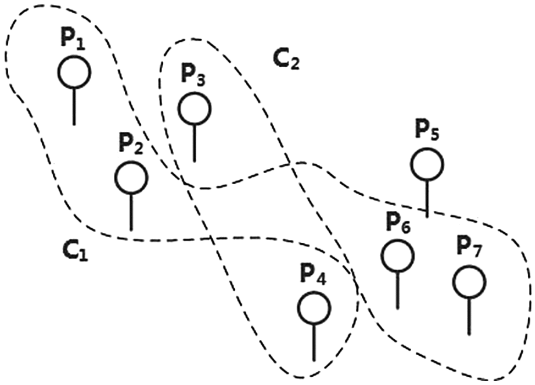

Fig. 3. Location cluster result of DCC algorithm

4.1 Similarity Based on Distance and Category

(1) Similarity in location distance
Locations have geographical position attributes, and they are represented by two points in a two-dimensional orthogonal coordinate system. In the two-dimensional space, we can use the Euclidean distance to measure the distance between two locations. However, we should use spherical distance formula to calculate the actual distance of two locations because Earth is a sphere. If location p_i's geographic coordinate is (log_i, lat_i), location p_j's geographic coordinate is (log_j, lat_j), and R = 6370856 (meters) is the rough radius of Earth, then the distance formula of the two locations can be shown as:

$$Dis(p_i, p_j) = R \cdot \cos^{-1}(\sin(lat_i) \cdot \sin(lat_j) + \cos(lat_i) \cdot \cos(lat_j) \cdot \cos(lon_i - lon_j)) \quad (1)$$

According to the geographical agglomeration effect, the locations with closer distances will have higher similarities. Thus, location distance similarity is inversely related to their distance.

$$Sim_{distance}(p_i, p_j) = \frac{1}{1 + Dis(p_i, p_j)} \quad (2)$$

(2) Similarity in location category
Aside from the similarity in distance, locations have similarity in category. According to Fig. 1, for independent trees "catering service" and "sports and leisure service," calculating the similarity between the c_3 in "catering service" and c_4 in "sports and leisure service" is possible. However, it is not different from the common classification recommendation if we cannot calculate the similarity among different large categories. As a solution, we defined that it has similarities with different degrees with each location in the location model based on distance and category. Therefore, we add a root layer upon the CategoryLarge layer so that all the nodes in the

hierarchical category tree possess a common ancestor node, that is, the root category. As a result, the hierarchical category tree becomes a connected graph.

Intuitively, the category similarity of the two locations can be transferred into the distance between the two leaf nodes in the hierarchical category tree, and a higher distance, which is indicated by the number of edges between the two leaf nodes, corresponds to lower similarity. However, we cannot obtain a satisfactory accuracy if we use only the edge number between the two leaf nodes as the measurement standard of the similarity. For instance, the two leaf nodes, (cinema) and (Cantonese restaurant) have an edge number of eight. The nodes (cinema) and (colleges and universities) also have an edge number of eight, but obviously, we are more likely to have dinner in a nearby Cantonese restaurant after watching a movie. Therefore, to show this kind of similarity, we set different weights for each edge.

(1) Division of the category of edges
First, we divide the edges from the cluster layer to the CategoryLarge layer into two categories: one is the $e_{cluster}$, the edges from the cluster layer to the CategorySmall layer, which is a known value; the others are the $e_{categorysmall}$, $e_{categorymedium}$, and $e_{categorylarge}$, which correspond to each hierarchy in CategorySmall, and this kind of edges should be constructed from low to high.

(2) Construction of the weights of $e_{categorysmall}$
In this paper, we use the dataset crawled from Sina microblog to construct $e_{categorysmall}$. We count the total number of check-ins of every category in the CategorySmall layer, and we label it $CheckinNum_j$

$$e_{categorysmall} = \frac{1}{1 + \log_{10} CheckinNum_j} \tag{3}$$

We use the log function to reduce the effect on the calculation of similarity because of significant data diversity. We add one to avoid a zero denominator. The $e_{categorysmall}$ is inversely related to the check-in number, that is, a higher check-in number corresponds to lower weights of its edges. Therefore, clustering is simplified because of the close distance to other categories.

(3) Construction of the weights of $e_{categorymedium}$ and $e_{categorylarge}$
We take the average of the edge weights of the lower class that is connected directly to the edge as the weights of $e_{categorymedium}$ and $e_{categorylarge}$.

$$w_{categorymedium} = \frac{\sum w_{categorysmall}}{n_{categorysmall}} \tag{4}$$

$$w_{categorylarge} = \frac{\sum w_{categorymedium}}{n_{categorymedium}} \tag{5}$$

(4) Construction of the weights of $e_{cluster}$

We define the weights of $e_{cluster}$ as half of the minimum value in $w_{categorysmall}$.

$$w_{cluster} = 0.5 \times \min(w_{categorysmall}) \tag{6}$$

(5) Category similarity calculation among the locations
After finishing the construction of the weights of each hierarchy's edge, we calculate the category similarity among the locations in the dataset. We define the category similarity by the following formula because it is inversely related to distance.

$$Sim_{category}(p_i, p_j) = \frac{1}{\sum w_{cluster} + \sum w_{category}} \tag{7}$$

(3) Total similarity of locations
Location similarity based on distance and category essentially involves clustering related locations through the balance of distance and category. If two locations p_a and p_b have a close range but they belong to two large categories with significant difference, then they cannot be clustered. By contrast, if the above locations have a far range but they belong to the same small category, then they are likely to be clustered.

$$Sim(p_i, p_j) = \alpha \cdot Sim_{category}(p_i, p_j) \cdot Sim_{distance}(p_i, p_j) \tag{8}$$

4.2 Location Clustering Based on Affinity Propagation (AP) Algorithm

We will obtain a location similarity matrix *S* after calculating the similarity of N locations. This section will cluster the location similarity matrix through the AP algorithm and select the clustering center automatically through the information delivered by the locations. Compared with the DBSCAN algorithm, the AP algorithm effectively confirms the parameters of the clustering algorithm, and we could obtain a satisfactory result if we use only the default value. In addition, selecting a signed location after clustering to ensure a practical clustering effect is unnecessary. Furthermore, the AP algorithm can handle a massive dataset in a short time and obtain an ideal result.

We add the similarity of distance and category into the definition of similarity among locations, after which we use the AP algorithm to select the most representative location as the typical point of clustering. Therefore, the nodes in the result set may have a close range or similar category. In conclusion, the RC algorithm has a better effect on location clustering than the algorithm based on space density.

4.3 Calculation of User Similarity and Recommendation Result

After clustering the location, we transfer the user check-in vector, calculate the location preference of each user, and calculate the Top-K similar users of each user. Finally, we calculate the prediction score of the unvisited location according to the location

preference of the target recommendatory user and show the locations with Top-N prediction scores to the target user as the final recommendation.

5 Experimental Results and visualization

5.1 Evaluation criteria

We use an offline experiment method to evaluate the effect of the algorithm in this paper. The data for the experiment is crawled through the Sina microblog API. For each target user, we use 80 % of their check-in record as the training set and the remaining 20 % for the test set, which is used to verify the effect of the recommendation. In addition, we select precision and recall, which are the most common index in the evaluation of the recommender system performance.

$$Precision = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |R(u)|} \tag{9}$$

$$Recall = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |T(u)|} \tag{10}$$

Here, $R(u)$ is the location recommendation based on the user's check-in record in the training set, and $T(u)$ is the user's check-in record in the test set.

5.2 Visualization

To strengthen the ability to display the multidimensional information of this recommender system and improve the intuition of the recommendation, we created a visualization model for the user recommendation that is based on force-directed algorithm, and a visualization for location recommendation is implemented.

Fig. 4. Visualization of historic check-in record

We concentrated on elaborating the design of the recommendatory algorithm. Thus, we display only the interfaces of historic check-in record (Fig. 4), DCC user recommendation (Fig. 5), and DCC location recommendation (Fig. 6).

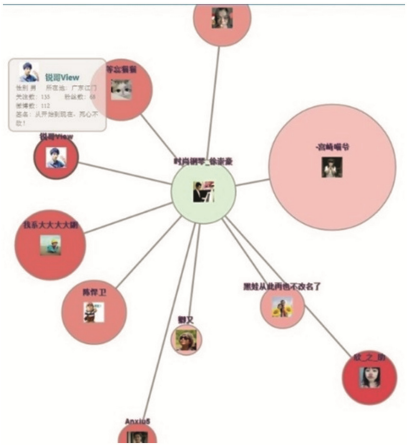

Fig. 5. Visualization of DCC user recommendation

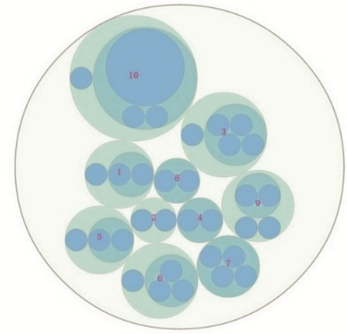

Fig. 6. Visualization of DCC location recommendation

6 Conclusion and Future Work

Social networking based on location service is a network platform with multidimensional information. It reflects extensive information about an individual in society to facilitate a thorough understanding of problems and laws in our daily life through the study of LBSNs.

Except for distance and category, we did not include additional feature information in the business of location into location clustering, such as the law of check-in time,

location tag, and user comments. We expect to enhance the effect of location clustering by including additional feature information in future research.

Acknowledgements. This work was supported by the National Natural Science Foundation of China under Grant Nos. 61073132 and 60776796; the Fundamental Research Funds for the Central Universities (101gpy33); Special Project on the Integration of Industry, Education and Research of Guangdong Province (No. 2012B091000062); Project 985 of Innovation Base for Journalism & Communication in the All-media Era, Sun Yat-sen University.

References

1. Chen, Y.: Analysis in SoloMo application pattern of mobile internet. Telecommun. Sci. **03**, 18–22 (2012)
2. Berjani, B., Strufe, T.: A recommendation system for spots in location-based online social networks. In: Proceedings of the 4th Workshop on Social Network Systems, p. 4. ACM (2011)
3. Ying, J.J.-C., Lu, E.H.-C., Kuo, W.-N., et al.: Urban point-of-interest recommendation by mining user check-in behaviors. In: Proceedings of the ACM SIGKDD International Workshop on Urban Computing, pp. 63–70. ACM (2012)
4. Ye, M., Yin, P., Lee, W.-C., et al.: Exploiting geographical influence for collaborative point-of-interest recommendation. In: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 325–334. ACM (2011)
5. Lee, M.-J., Chung, C.-W.: A user similarity calculation based on the location for social network services. In: Yu, J.X., Kim, M.H., Unland, R. (eds.) DASFAA 2011, Part I. LNCS, vol. 6587, pp. 38–52. Springer, Heidelberg (2011)
6. Goldberg, D., Nichols, D., Oki, B.M., et al.: Using collaborative filtering to weave an information tapestry. Commun. ACM **35**(12), 61–70 (1992)
7. Leung, K.W.T., Lee, D.L., Lee, W.C.: CLR: a collaborative location recommendation framework based on co-clustering. In: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 305–314. ACM (2011)
8. Zhou, D., Wang, B., Rahimi, S.M., Wang, X.: A study of recommending locations on location-based social network by collaborative filtering. In: Kosseim, L., Inkpen, D. (eds.) Canadian AI 2012. LNCS, vol. 7310, pp. 255–266. Springer, Heidelberg (2012)
9. De Sa, J.M.: Pattern Recognition: Concepts, Methods, and Applications. Springer, Berlin (2001)
10. Cheu, E.Y., Keongg, C.,Zhou, Z.: On the two-level hybrid clustering algorithm. In: International Conference on Artificial Intelligence in Science and Technology, pp. 138–142 (2004)
11. Fred, A., Leitao, J.M.: Partitional vs hierarchical clustering using a minimum grammar complexity approach. In: Amin, A., Pudil, P., Ferri, F., Iñesta, J.M. (eds.) SPR 2000 and SSPR 2000. LNCS, vol. 1876, pp. 193–202. Springer, Heidelberg (2000)
12. Nanni, M., Pedreschi, D.: Time-focused clustering of trajectories of moving objects. J. Intell. Inf. Syst. **27**(3), 267–289 (2006)
13. Yanchang, Z., Junde, S.: GDILC: a grid-based density-isoline clustering algorithm. In: 2001 International Conferences on Info-tech and Info-net, 2001 Proceedings ICII 2001-Beijing, pp. 140–145. IEEE (2001)