



ARTICLE

User Role Discovery and Optimization Method Based on K-means++ and Reinforcement Learning in Mobile Applications

Yuanbang Li^{*}, Wengang Zhou, Chi Xu and Yuchun Shi

School of Computer Science and Technology, Zhoukou Normal University, Zhoukou, 466001, China

^{*}Corresponding Author: Yuanbang Li. Email: lybang@whu.edu.cn

Received: 07 October 2021 Accepted: 30 November 2021

ABSTRACT

With the widespread use of mobile phones, users can share their location and activity anytime, anywhere, as a form of check-in data. These data reflect user features. Long-term stability and a set of user-shared features can be abstracted as user roles. This role is closely related to the users' social background, occupation, and living habits. This study makes four main contributions to the literature. First, user feature models from different views for each user are constructed from the analysis of the check-in data. Second, the K-means algorithm is used to discover user roles from user features. Third, a reinforcement learning algorithm is proposed to strengthen the clustering effect of user roles and improve the stability of the clustering result. Finally, experiments are used to verify the validity of the method. The results show that the method can improve the effect of clustering by 1.5~2 times, and improve the stability of the cluster results about 2~3 times of the original. This method is the first time to apply reinforcement learning to the optimization of user roles in mobile applications, which enhances the clustering effect and improves the stability of the automatic method when discovering user roles.

KEYWORDS

User role discovery; user role optimization; K-means++; reinforcement learning

1 Introduction

According to the 47th Statistical Report on the Development of China's Internet by the China Internet Network Information Center [1], as of December 2020, the number of Chinese netizen reached 989 million, of which 986 million were mobile netizen, accounting for 99.7%. With the widespread use of mobile phones, mobile applications have achieved rapid development because users can share their interests and freely send various types of information [2].

Massive amounts of check-in data are generated in the usage of mobile applications by users. These data include a wealth of user knowledge. Analyzing these large amounts of data and vector them as user features are interesting research directions [3].

In mobile applications, users' behaviors may be diverse and change significantly over time. However, eventually, these behaviors will have certain rules, and these are often shared by a group of users. This stable, common behavioral rule can be abstracted into the user roles. Although the



definitions of roles are different in different disciplines, the core concept is the same, that is, a role is a set of user groups that share similar behavior patterns. User roles are closely related to the user's social background, occupation, and living habits [4].

The following are the contributions of this paper.

- (1) An algorithm was designed to realize the vectorization of user features from user behavior data.
- (2) A method based on the K-means++ algorithm is proposed to discover user roles through the analysis of user features.
- (3) A method based on reinforcement learning is proposed to improve the stability of the user-role discovery method.
- (4) Experiments were conducted to verify the validity of the method.

The subsequent chapters are arranged as follows: [Chapter 2](#) introduces the related work. [Chapter 3](#) introduces the preliminaries of the method. [Chapter 4](#) introduces the process of the method. [Chapter 5](#) experimentally verifies the effectiveness of the method. Finally, [Chapter 6](#) summarizes the paper.

2 Related Works

Role analysis has become an important analysis method and has been widely used in many applications, such as mobile applications. Yu [5] defined roles from an abstract level: a role is an abstract representation of the behaviors of social actors in a specific professional background or field of work. Zhang et al. [6] defined roles specifically in virtual communities: a role is a behavior model and behavior norm that meets the requirements of the virtual community generated by users under psychological motivation factors, which is different from social roles. They further noted that, a role is not the expectation of people's behaviors in the real society but the expansion and extension of social roles.

The role discovery methods can be divided into the following categories: expert experience-based, survey-based, and mathematical analysis-based methods.

- Expert experience-based method

This is a method in which domain experts define user roles in a specific field based on personal experience. For example, Liu-Thompkins et al. [7] divided users in computational advertising into three roles: creator, metavoicer, and propagator, and analyzed the opportunities, challenges, and future research directions of each role.

- Survey-based method

This method obtains user roles through a survey. According to the different objects of the survey, this method can be divided into literature-based survey methods and user-based survey methods. Literature-based methods such as systematic literature review was used by Hacker et al. [8] to investigate user roles in enterprise social networks, and 10 user roles, including initiator, debater, sharer, coordinator, seeker, helper, expert, net-worker, linker, and observer were finally obtained. The user-based survey method, such as the GLAM digital participation survey activity launched at the University of Oxford in the United Kingdom, through user surveys on users of gardens, libraries, and museums, is divided into five roles: content producers, compliant learners, and cultural consumers, topic lovers, and education practitioners [9].

- **Mathematical analysis-based method**

This method uses various mathematical methods or analysis techniques, such as clustering algorithms, statistical analysis, and log analysis, to discover user roles through data analysis and summary. For example, Hacker et al. [10] used the K-means clustering algorithm to analyze corporate social network data. Through the analysis of the clustering results, users are divided into nine roles: power users, conversation starters, well-connected helpers, focused information sharers, sporadic users, task coordinators, offline experts, chat users, and team members. Guo et al. [11] used social network analysis and clustering analysis techniques to analyze user data in professional virtual communities and divided users into six roles: planning instructors, active designers, multifaceted persons, communicators, passive designers, and observers.

Expert's personal domain knowledge and social background affect the results of expert experience-based methods, and the results summarized by different experts may be quite different. Additionally, when the field is complex and involves many users and frequent user activities, experts find it difficult to sum up user roles.

The survey-based method does not require complicated equipment and experiments and is relatively easy to implement. It is a feasible method for collecting first-hand data to summarize user roles. However, this method is labor-intensive and difficult to reuse; if the research field or the user changes, the survey process should be repeated. The researcher's skills and experience also affect the results. In mobile applications, the openness and diversity of users also increase the difficulty of using survey methods.

In mobile applications, users are transparent to managers, and user behavior data are rich. These characteristics are suitable for mathematical analysis-based methods. Therefore, many studies have been conducted using this method to discover user roles [10–13]. However, current research focuses only on the discovery of roles without considering the stability of roles and corresponding user groups. Experiments show that for the same data, running the K-means algorithm multiple times has greater randomness in the division of the user's role. Therefore, based on using the K-means algorithm to discover the user's role, this study uses the reinforcement learning method to learn the user role division, which improves the stability of the division.

3 Preliminaries

3.1 Definitions

The definitions used in the method are as follows:

Definition 1: User context set (UCS)

The USC is a set of attributes that can be used to describe the user and context for user activities.

$$UCS = \{UC^i | 0 \leq i < |UCS|\} \quad (1)$$

Definition 2: View set (VS)

VS is a set of different perspectives that system analysts and managers use to observe user activity patterns according to their interests.

$$VS = \{V^j | 0 \leq j < |VS|\} \quad (2)$$

Definition 3: User context view preference set (UFS)

The UCVFS is the set of user preferences from a contextual perspective determined from different perspectives under different scenarios.

$$UFS = \{UF^{ij} | 0 \leq i < |UCS|, 0 \leq j < |VS|\} \quad (3)$$

where:

$$UF^{ij} = \begin{pmatrix} uf^{ij}_{11} & \dots & uf^{ij}_{1|V^j|} \\ \vdots & \ddots & \vdots \\ uf^{ij}_{|UC^i|1} & \dots & uf^{ij}_{|UC^i||V^j|} \end{pmatrix} \quad (4)$$

3.2 Basis of Reinforcement Learning

Reinforcement learning theory is derived from the psychology and neuroscience of animal behavior and provides an explanation of how agents optimize their control of the environment [14]. In an unknown and complex environment, the agent behavior and resulting consequences are usually unpredictable. The process of reinforcement learning does not directly tell the agent which behavior to perform but constantly tries to make the agent perform various behaviors in the environment. Further, it calculates the instant reward resulting from a single behavior on the environment by defining a reward function. The instant reward is used to identify the agent whose behavior is the best in the current environment. The larger the instant reward, the better is the behavior. However, in a complex environment, the optimal solution for each behavior may not reach the overall optimal solution. Therefore, the value function must be defined. The value function is used to measure the long-term value of an agent's behavior.

The components of reinforcement learning used in this study include the following:

State set S: The state set refers to the set of all possible states of the agent from the beginning of learning to completion. It is marked as S and contains several states s, that is, $s \in S$.

Behavior set A: The behavior set refers to the set of all behaviors in the learning process of the agent, marked as A, and contains several behaviors a, that is, $a \in A$.

State behaviors set A_s : State behavior set refers to the set of possible behaviors of the agent in a specific states, marked as A_s and contains several possible behaviors a_s , $a_s \in A_s$. **Instant reward $Reward_s^a$:** Instant reward refers to the reward generated when the agent adopts behavior a in a specific state s, which is marked as $Reward_s^a$.

Long-term value V: Long-term value refers to the sum of the value generated by the agent after adopting a series of behaviors, marked as V.

4 Method Proposed to Realize User Role Discovery and Optimization

4.1 UFS Construction

Quantifying user characteristics by analyzing user activity data is the basis of using automatic methods to analyze user features and discover user roles. In this study, we first implemented vectorization from user activity data to user features. According to Definition 3, the UFS is a set of cardinalities $|VS| * |UCS|$, where each set element is a matrix of $|UC^i|$ times $|V^j|$ dimensions. The UFS algorithm is presented in Algorithm 1.

The input of the algorithm includes DS, UCS, and VS, and the output is the UFS. First, the check-in data of user u are initialized from DS (Line 1). For each context UC^i in the UCS; each perspective V^j in VS is iterated. For this process, UF^{ij} is first initialized to a zero matrix; the row and column are the cardinality of UC^i and V^j (Row 4). Iteration occurs over the data record d of user U , initializing the row and column subscripts of the matrix corresponding to each record to 0 (Rows 5–7). The corresponding user context value uc is obtained, and the value v is viewed. The subscript uc , that is, uc_serial_num , and the subscript v , that is, v_serial_num (Lines 8–17), are calculated. The value of the matrix corresponding to the position of the row and column subscripts was added (Line 18). Finally, UF^{ij} was added to the UFS (Line 20).

Algorithm 1: UFS construction algorithm

Input: DS (Data Set), UCS, VS

Output: UFS

```

1: Initialize data of user  $u$  DS_u from DS
2: for  $UC^i$  in UCS
3:   for  $V^j$  in VS
4:     Initialize  $UF^{ij} = O(|UC^i|*|V^j|)$ 
5:     for  $d$  in DS_u
6:        $uc\_serial\_num = 0$ 
7:        $vs\_serial\_num = 0$ 
8:       for  $uc$  in  $UC^i$ 
9:         for  $v$  in  $V^j$ 
10:          if ( $d.UC^i\_value == uc \ \&\& \ d.V^j\_value == v$ )
11:             $uc\_serial\_num = GetSerial\_Number(uc)$ 
12:             $vs\_serial\_num = GetSerial\_Number(v)$ 
13:            break
14:          endif
15:        endfor
16:      break
17:    endfor
18:     $UCVF^{ij}[uc\_serial\_num][vs\_serial\_num] ++$ 
19:  endfor
20:  UFS.add( $UF^{ij}$ )
21: end for
22: end for
23: return UCVFS

```

4.2 Overview of the Method

The concept behind this method is that the UFS established in the previous section is the learning object. First, the K-means++ algorithm is used to cluster the user features and discover user roles. On this basis, the idea of reinforcement learning is used to define the appropriate immediate reward function and long-term value function through the learning of the subordination between users and roles. Through continuous iterative optimization, an ideal solution set was obtained. An overview of this method is shown in Fig. 1.

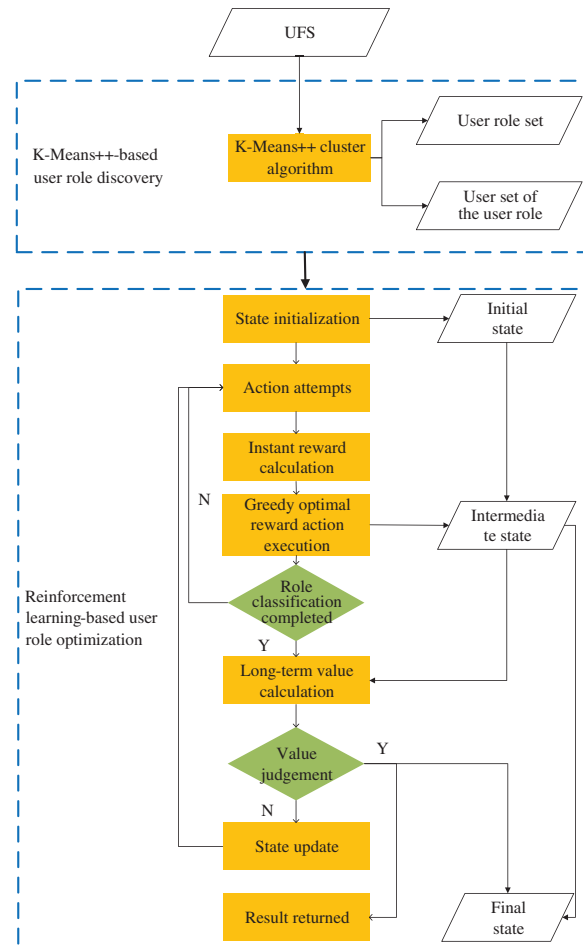


Figure 1: Overview of the method

4.3 K-Means++-Based User Role Discovery

A variety of clustering methods were investigated to discover user role effectively, including: 1) DBSCAN (Density-Based Spatial Clustering Application with Noise) method; 2) AHC (Agglomerative Hierarchical Clustering) method; 3) K-Means++ method. And user's time-root category feature in data_1, which is detail described in Section 5.2 is used to validate the effectiveness of these method.

- DBSCAN method

DBSCAN method is a density-based non-parametric clustering algorithm [15]: given a set of points in a space, the closely arranged points, that is, the neighboring points, are clustered together.

Two parameters are needed in the method, one is the radius of the search named "eps", and the other is the minimum number of points in the range named "min_samples". Outliers may appear during the clustering process because they are too far away from their neighbor points and do not belong to any category. Therefore, it is necessary to analyze the parameter combination

(eps, min_samples) to ensure the rationality of the number of categories and the effective data ratio. The specific analysis of the case data is shown in Figs. 2 and 3.

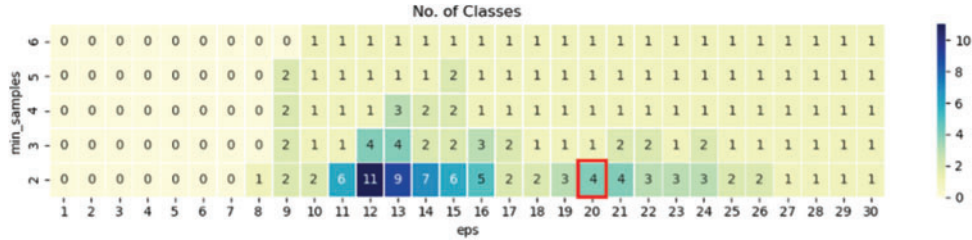


Figure 2: Number of categories of DBSCAN method

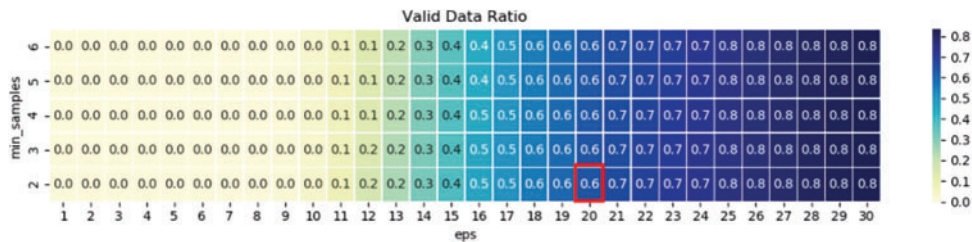


Figure 3: Effective data ratio of DBSCAN method

As shown in Fig. 2, when the radius is between 9 and 26, the classification number is greater than 2. When the value of parameter combination (eps, min_samples) is (12, 2), the number of clusters reaches the maximum, which is 11 categories, however, as shown in Fig. 3, under this parameter, the effective data ratio is only 0.2. Through a comprehensive analysis of the combination of the two parameters, we take (20, 2) as the final combination parameter, under this parameter, the number of clusters is 4, and the effective data ratio is 0.6, as shown in the small red box in the figures.

- AHC method

Hierarchical clustering obtains a cluster tree by condensing the data at different levels, which consist of agglomerative hierarchical clustering and split hierarchical clustering. In the agglomerative hierarchical clustering method [16], each record is a cluster at the beginning. Then the proximity between the clusters is calculated and the initial clusters are merged until it finally reaches the set number of categories. In this study, the AHC method was used to construct a hierarchical clustering tree of the user’s time-root category features in data_1, as shown in Fig. 4. It can be found from the figure that the data is reasonably divided when the number of clusters is 6, as shown by the yellow horizontal line in the figure.

- K-means++ method

K-means algorithm divides the data into K clusters and minimize the distance between each sample and the center of its class. It has been widely used in practice because of its easy implementation and fast convergence speed [17]. K-means algorithm is sensitive to the selection of the initial clustering center. Improper selection of the initial clustering center affects the clustering effect. The k-means++ algorithm optimized the k-means algorithm, its basic idea is to make the distance between the initial center points as far as possible, that is, the center points are in the

“mutually exclusive” state [18]. The value of K can be determined by calculating the root mean square error (RMSE) within the cluster. If the error decreases before K reaches a certain value, and then slows down, forming an obvious “elbow,” then the K value is reasonable. Taking user’s time-root category features in data_1 as an example, the specific changes of mean validation score according to K are shown in Fig. 5. The figure shows that when the number of clusters is 9, there is an obvious elbow, as marked in red in the picture.

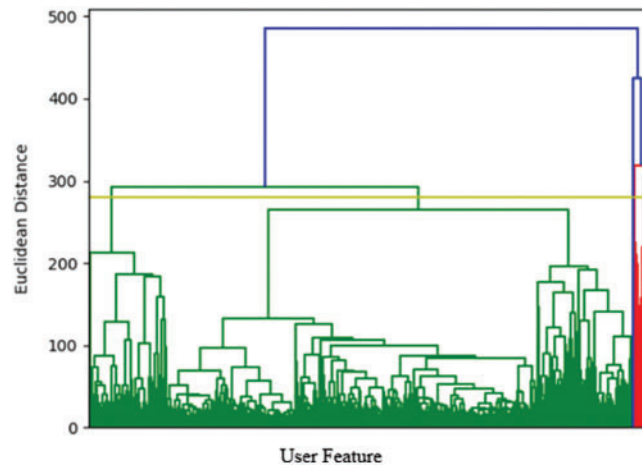


Figure 4: Agglomerative hierarchical clustering tree

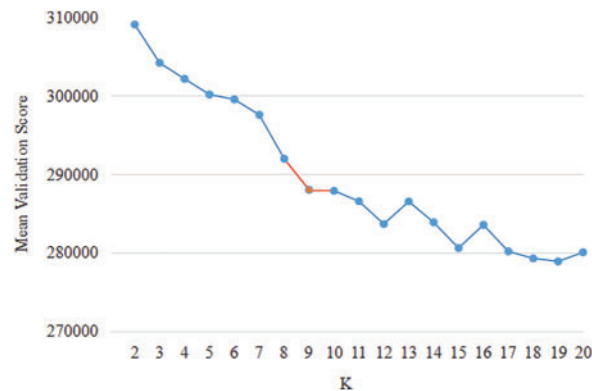


Figure 5: Change trend of mean validation score according to K

Applicability of the three clustering methods was analyzed from five aspects: data coverage, number of categories, number of users for each category, standard deviation of number of users, and minimum number of users for each category. The result is shown in Table 1.

In data coverage aspect, the effective data used by the DBSCAN method accounting for 64.7%, corresponding to 100% of AHC and K-means++ method. In the number of categories aspect, the number of the K-means++ is 9, corresponding to 4 and 6 of AHC and K-means++ method, and a further analysis of the classification results found that the categories obtained by DBSCAN and HCA method can be found in the K-means++ algorithm with similar feature

categories. The number of users for each category is listed in Column 4, and the standard deviation of number of users is calculated in Column 5, in which K-means++ algorithm is the smallest. In the minimum number of users for each category aspect, the K-means++ algorithm is the largest, as shown in Column 6 of the table. Based on the analysis above, K-means++ algorithm was finally selected as the clustering algorithm in this research.

Table 1: Applicability analysis table for clustering method

Method	DC	NC	NU	SD	MN
DBSCAN	64.7%	4	[695, 2, 2, 2]	600.2	2
AHC	100%	6	[895, 150, 20, 15, 2, 1]	792.9	1
K-means++	100%	9	[438, 225, 210, 74, 55, 46, 21, 7, 7]	409.2	7

Note: DC: data coverage, NC: number of categories, NU: number of users for each category, SD: standard deviation of number of users, MN: minimum number of users for each category.

4.4 Reinforcement Learning Based User Role Optimization

The K-means++ algorithm can discover user roles and divide the roles to which the user belongs, but because the initial K centers of the algorithm are random, and the order of input will also impact the clustering results, the clustering results have randomness. Therefore, in this method, the idea of reinforcement learning is used to define a reasonable real-time reward and long-term value calculation function to optimize the clustering results through many rounds of continuous attempts and learning; finally, a satisfactory solution is obtained.

To describe the user role optimization method based on reinforcement learning, the concept of silhouette coefficient is introduced, which is a classic index used to measure the quality of clustering [19].

- Silhouette Coefficient

The centroid of clustering is the core concept used to calculate the silhouette coefficient. Let R be a cluster; then, its centroid is the mean vector of all vectors of the cluster, and the calculation is shown in Eq. (5).

$$Centroid(R) = \frac{\sum_{r_i \in R} r_i}{|R|} \quad (5)$$

For each user feature UF_i , the calculation process for the silhouette coefficient is as follows:

- (1) Calculate the maximum distance between the user feature UF_i and other user features in the role, marked as $TRMax(UF_i)$.
- (2) Calculate the minimum distance from the user feature UF_i to the centroid of other characters, marked as $ORMin(UF_i)$.
- (3) Calculate the silhouette coefficient according to $TRMax(UF_i)$ and $ORMin(UF_i)$, as shown in Eq. (6).

$$Se(UF_i) = \frac{ORMin(UF_i) - TRMax(UF_i)}{Max(ORMin(UF_i), TRMax(UF_i))} \quad (6)$$

The value range of the silhouette coefficient $Se(UF_i)$ was between -1 and 1 . $Se(UF_i)$ can be used to measure the clustering effect of UF_i for a single user feature. The larger the value, the better the clustering quality. To measure the overall clustering effect of all user features, the average contour coefficient of the user features is used. The larger the average contour coefficient, the better the overall clustering quality. Suppose that the user set $u = \{U_1, U_2, U_i, \dots, U_m\}$, where $m = |u|$ and the user feature corresponding to user U_i is UF_i . The calculation of the average silhouette coefficient is shown in [formula \(7\)](#).

$$Se(UF) = \frac{1}{m} \sum_{i=1}^m Se(UF_i) \quad (7)$$

After introducing the concept of the silhouette coefficient, a user role optimization method based on reinforcement learning is proposed, which consist of the following 9 steps.

1) Data preparation

The input data used in state initialization include user set U , feature set UF , role set $ROLE$, and role user list set $ROLE_USERLIST$. The specific definitions of the data are as follows:

User set $U = \{U_1, \dots, U_i, \dots, U_m\}$, indicating all users, where $m = |U|$.

Feature set $UF = \{UF_1, \dots, UF_i, \dots, UF_m\}$, where the user feature corresponding to user U_i is UF_i .

The role set $role = \{R_1, \dots, R_i, \dots, R_n\}$, which represents the role abstracted from the feature, where $n = |role|$. The construction process of the role set is shown in [Section 4.3](#).

The role user list set $ROLE_USERLIST = \{ROLE_USERLIST_1, \dots, ROLE_USERLIST_i, \dots, ROLE_USERLIST_n\}$, represents the user list corresponding to each role, and $U = \bigcup_{i=1}^n ROLE_USERLIST_i$ is established.

2) State initialization

The purpose of reinforcement learning is to obtain a satisfactory solution for the user's role. Therefore, the system state is described as the subordinate matrix of the user and its role. If the initial state is S^0 , then the row number of S^0 is $|U|$, and the column number is $|ROLE|$. The S^0 initialization algorithm is presented in Algorithm 2.

Algorithm 2: State initialization algorithm

Input: $U, ROLE_USERLIST$

Output: S^0

- 1: Initialize $S^0 = O(|U|*|ROLE|)$
 - 2: for U_i in U
 - 3: for $ROLE_USERLIST_j$ in $ROLE_USERLIST$
 - 4: if $(U_i \in ROLE_USERLIST_j)$
 - 5: $S^0_{[i-1][j-1]} = \alpha$
 - 6: endif
 - 7: endfor
 - 8: endfor
 - 9: return S^0
-

3) Action attempts

Take user U_i from user set U and try to assign him to each role R_j in the ROLE.

4) Instant reward calculation

Using formula (6) to calculate the instant reward when U_i is assigned to each role R_j , the candidate instant reward set $REWARD = \{Se(UF_i)_1, Se(UF_i)_2, Se(UF_i)_j, \dots, Se(UF_i)_n\}$, where $n = |ROLE|$.

5) Greedy optimal reward action execution

Using the idea of the greedy algorithm, we take the current local optimal solution, that is, the role corresponding to the maximum value in the REWARD set, marked as R_j . The implementation process includes the following steps:

- a. Assign U_i to role R_j , and update centroid R_j using formula (5);
- b. Update the user set U , execute $U = U - U_i$;
- c. Update the state S^n to S^{n+1} , using formula (8)

$$S^{n+1}[i-1][j-1] = (1-\beta) * S^n[i-1][j-1] + \beta * Se(UF_i)_j \quad (8)$$

In formula (8), β is a constant, and its value is between [0,1], which is used to adjust the influence of the immediate reward obtained from this greedy strategy on the overall state. The greater the value of β , the greater is the influence of the immediate reward on the state.

6) Repeat Steps (3)–(5) until the end of one round of execution, that is, all users are assigned

7) Long-term value calculation

In this study, we believe that after the process of class reinforcement learning, the long-term value needs to achieve two goals. One is good role partition, and the specific indicators are measured by the average silhouette coefficient, which is calculated using formula (7). The other is reduced randomness of role partition, so that the partition results tend to be stable; that is, after a new round of allocation, users should still belong to the same role. It is assumed that the user list set of the role before allocation is $ROLE_USERLIST = \{ROLE_USER_1, \dots, ROLE_USER_i, \dots, ROLE_USER_n\}$, after a new round of partition, the role user list set is $ROLE_USERLIST' = \{ROLE_USER'_1, \dots, ROLE_USER'_i, \dots, ROLE_USER'_n\}$. The calculation of randomness is shown in formulas (9) and (10).

$$In(ROLE_i) = ROLE_USER_i \cap ROLE_USER'_i \quad (9)$$

$$Rand(ROLE) = \sum_{i=1}^n (|ROLE_USER_i - In(ROLE_i)| + |ROLE_USER'_i - In(ROLE_i)|) \quad (10)$$

8) Long-term value judgment

To determine the long-term value, if the two targets average silhouette coefficients $Se(UF) > \gamma$ and $Rand(ROLE) < \delta$ cannot be satisfied, both initialize the user set to all users and repeat Steps (3)–(8); if the two targets are satisfied, turn to Step (9). The constant γ represents the lowest

value of the acceptable average silhouette coefficient, and δ represents the maximum value of the randomness of the acceptable role partition.

9) **Ends the algorithm and return the results**

5 Experiments

5.1 Research Questions

According to the research motivation, the research questions (RQ) are as follows:

RQ1: What is the effect of automatically discovering user roles from user features using the K-means++ algorithm?

RQ2: What is the effect of using reinforcement learning to optimize user role discovery?

RQ3: Compared with existing user role discovery methods, what are the advantages of the method in this study?

5.2 Date Sets of Experiment

The data set is constructed based on the Foursquare user check-in data in existing research [20,21]. The data set includes three independent sets, which are labeled as data_1, data_2, and data_3. data_1 uses the check-in data of New York City in the study [20]. To verify that the method is not data sensitive, data_2 and data_3 are randomly generated from [21]. The statistical information of the data set is shown in Table 2, and an example of the data in the data set is shown in Table 3. The dataset can be downloaded directly at the following URL: <https://pan.baidu.com/s/1zibTOM9dKaSkLoap3bh97Q>, and the password is 1234.

Table 2: Data statistics table of the datasets

	User number	POI number	Check-in times
data_1	1083	38333	227428
data_2	5000	359036	1472935
data_3	8000	509440	2253379

Note: POI: Point of Interest.

Table 3: Data sample table of the datasets

U	P	PC	PCN	LO	LA	W	Y	M	D	T
1	4d*	4b*	American Restaurant	40.7*	-73.9*	Sat	2012	Apr	07	17:42:24
49	42*	4a*	Railway Station	40.7*	-73.9*	Wed	2012	Apr	04	12:11:28
...
712	4c*	4f*	Neighborhood	40.7*	-73.9*	Mon	2012	Nov	05	23:48:22

Note: U = user; P = POI; PC = POI category; PCN = POI category name; LO = longitude; LA = latitude; W = week; Y = year; M = month; D = day; T = time. P and PC in the data are represented by a string with a length of 24; for concise and intuitive, only the first two characters are given in the table, and the rest are replaced by *. The longitude and latitude in the data are accurate to 15 decimal places after the decimal point, for the concise and intuitive, two decimal places are given, and the rest are replaced by *.

To obtain an intuitive understanding of user characteristics from an abstract level, the root category data item is added to the data set. The corresponding relation between the POI category and POI root category is established based on the hierarchical category relationship tree on the Foursquare official website. On the website, nine types of root categories exist, which are arts and entertainment, college and university, food, outdoors and recreation, professional and other places, residence, shop and service, travel and transport, and event.

5.3 Results and Analysis

Through data analysis, the user context set $UCS = \{UC^t, UC^d\}$, where t represents time and d represents distance. In this study, the time was segmented in hours, so $|UC^t| = 24$. The distance from the user’s home to the POI is divided into 4 levels [22]: within 1 km, between 1 and 10 km, between 10 and 30 km, and more than 30 km. Therefore, $|UC^d| = 4$.

The view set $VS = \{V^r\}$, where r represents the root category of POI. Nine root categories of POI exist in the dataset, so $|V^r| = 9$.

Based on the analysis above, $UFS = \{UF^{time-root\ category}, UF^{distance-root\ category}\}$. $UF^{time-root\ category}$ is a $24 * 9$ matrix and $UF^{distance-root\ category}$ is a $4 * 9$ matrix. The construction of these matrices is shown in Algorithm 1.

According to the research questions, the results and analysis are as follows:

RQ1: What is the effect of automatically discovering user roles from user features using the K-means++ algorithm?

The K-means++ algorithm is used to cluster user features. The results are presented in Table 4.

Table 4: User roles generated from UFS

UFS	Role name
$UF^{time-root\ category}$	Noon outdoor activists, Urban practitioners, Night food lover, Early morning food lover, Lunch-time food lover, Afternoon at home food lover, Evening outdoor activists, Early morning revelers, Afternoon outdoor enthusiasts
$UF^{distance-root\ category}$	Long-distance public transportation riders, Long-distance food lovers, Surrounding professionals, Surrounding food lover, Nearby outdoor activists, Nearby food lover, Long-distance outdoor activists, Long-distance professionals, Surrounding outdoor activists

It can be found from the table that clustering different user features can describe the life habits of a user set from different perspectives, thereby providing a basis for analyzing users’ social attributes. Taking the $UF^{time-root\ category}$ as an example, a schematic of the user roles is shown in Figs. 6–14.

As shown in Fig. 6, users in this role sign in very frequently around noon, and sign in a lot in outdoor activities, shops and services, travel, and transportation. This shows that users like to be active around noon and like to use transportation and engage in daily activities such as outdoor activities or shopping.

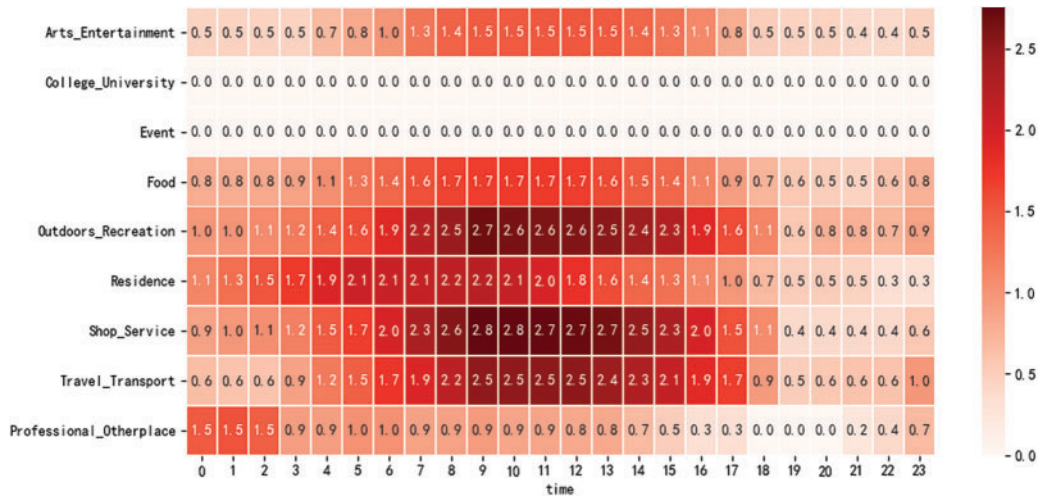


Figure 6: Noon outdoor activists

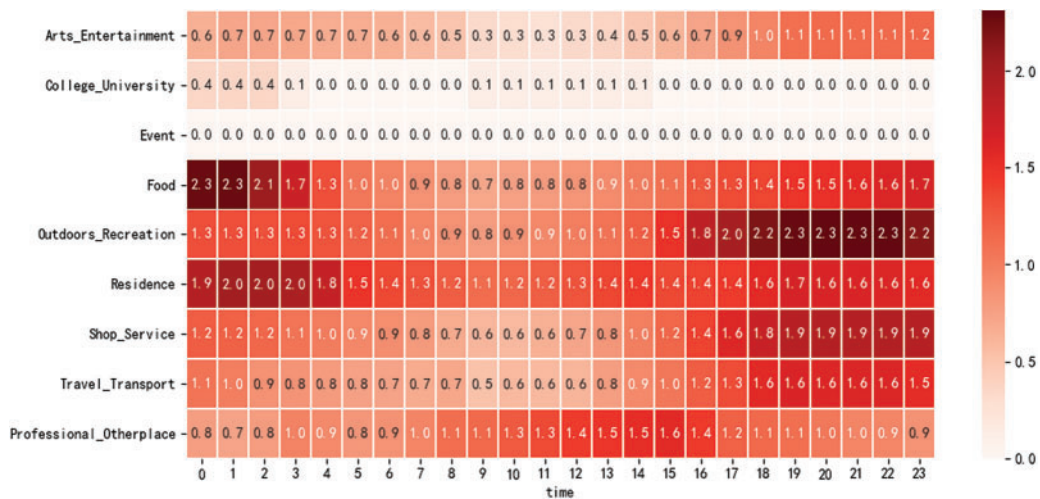


Figure 7: Urban practitioners

As shown in Fig. 7 from approximately 08:00 a.m. to approximately 16:00 p.m., users in this role check in more occupations in the root category of professional and other places, indicating that the user should be at work; from the end of work at around 17:00 p.m. to 23:00 p.m., users are more active, and activities are mainly concentrated in outdoor and recreation, shops and services, and travel and transport root categories. From 00:00 a.m. to 02:00 a.m., users' outdoor and shopping activities are significantly reduced, whereas there is more food and housing, indicating that users have been busy for a day and ate some supper before rest.

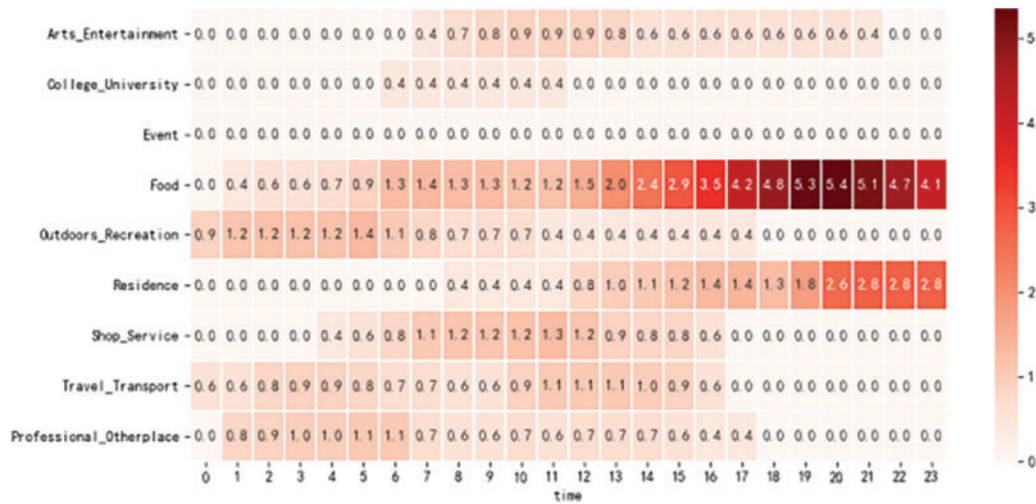


Figure 8: Night food lover

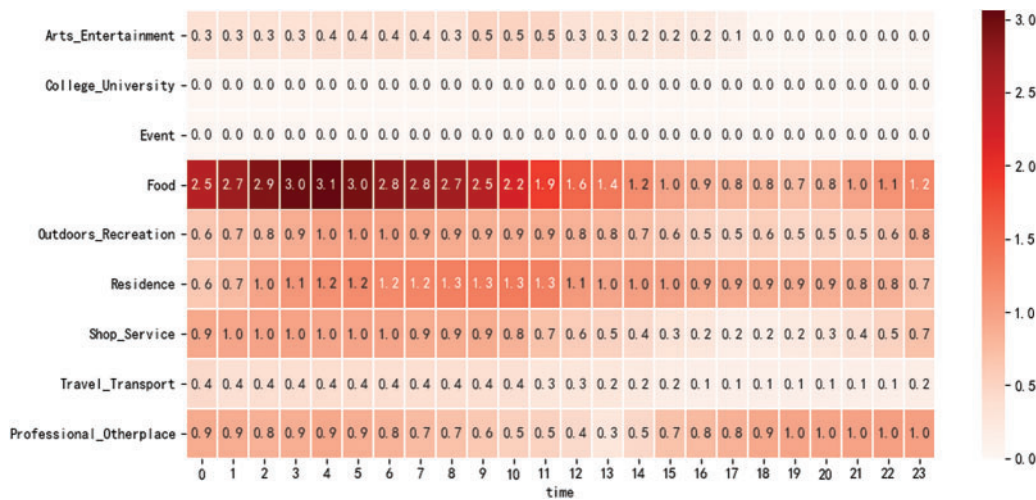


Figure 9: Early morning food lover

Fig. 8 shows that night food lovers are particularly fond of fine food, and check in activities are mainly concentrated between 17:00 p.m. and 23:00 p.m. Starting from 00:00 a.m., checking in drops sharply, indicating that users in this role do not like to stay up late.

As shown in Fig. 9, early morning food lovers have the most activity in the food root category, and their check in activity began to increase significantly after 00:00 a.m., indicating that users are interested in urban nightlife.

As shown in Fig. 10, lunch-time food lovers have the highest activity in the food root category, and the time is concentrated around noon. It was also found that the users' activity frequency in the morning was significantly higher than that in the afternoon in the professional and other root categories, suggesting that users would often be busy working in the morning and then enjoy a meal break at noon.

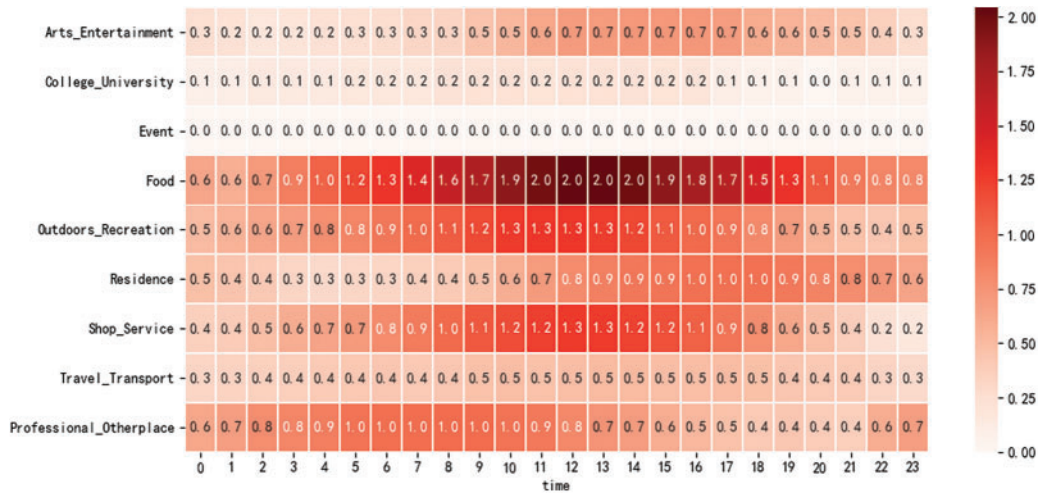


Figure 10: Lunch-time food lover

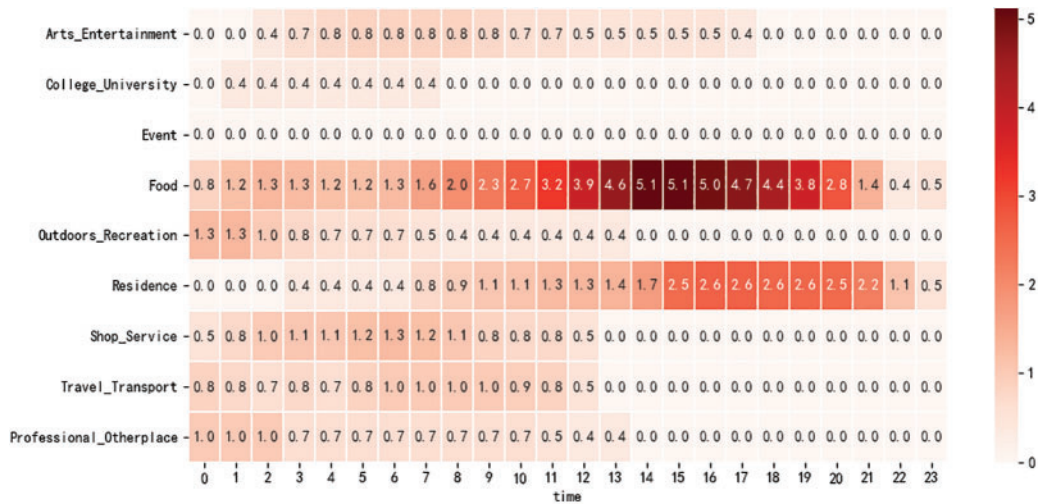


Figure 11: Afternoon at home food lover

As shown in Fig. 11, the users in this role had the highest activity in the food root category, with the time being concentrated in the afternoon, and they had significantly more activity in the residential category in the afternoon, indicating that users in this role like stay at home and enjoy fine food in the afternoon.

As shown in Fig. 12, check-in activities of users in this role are mainly concentrated between 17:00 and 22:00 p.m., and the frequency of activity drops significantly from 23:00 p.m. The main categories of activities are concentrated in shops and services, outdoor and recreation, and travel and transportation, indicating that users would always use public transportation and enjoy shopping and outdoor activities.

As shown in Fig. 13 check in activities of users in this role are concentrated after 00:00 a.m. The root categories of these check in activities include shops and services, outdoor and recreation, and travel and transportation, indicating that users are avid nightlife enthusiasts and like many kinds of outdoor activities such as shopping, outdoor activities, and recreation.

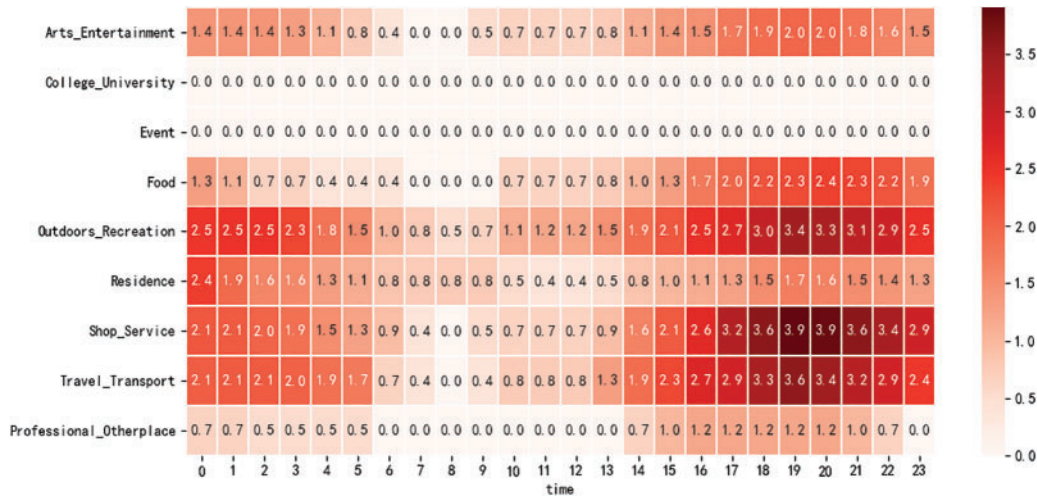


Figure 12: Evening outdoor activists

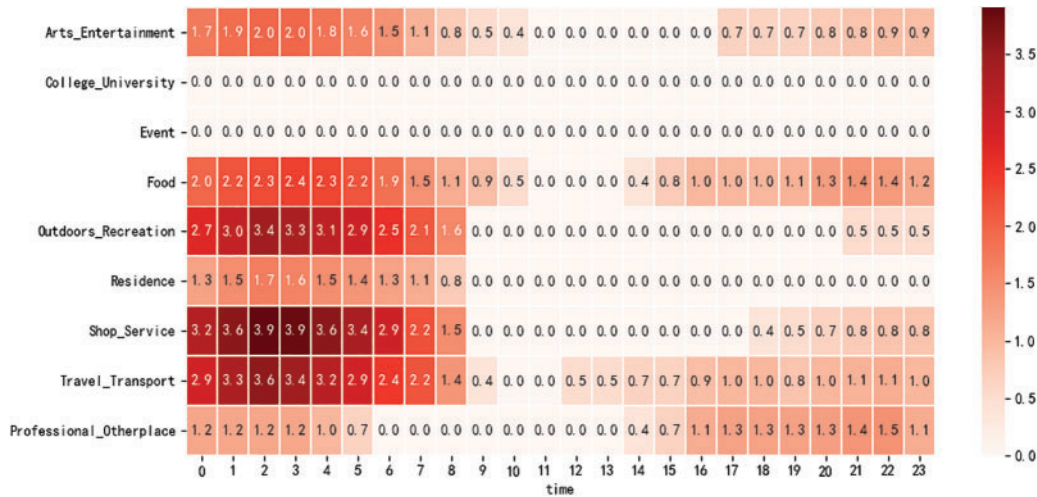


Figure 13: Early morning revelers

Fig. 14 shows that users in this role are frequently active from 12:00 p.m. to 18:00 p.m., and the activities are mainly in shop and services, outdoor and recreation, and travel and transportation, indicates that users often take public transportation and like shopping and outdoor activities.

RQ2: What is the effect of using reinforcement learning to optimize user role discovery?

- Parameter setting

The values of γ and δ are first set according to a set of experiments, where γ is the threshold value of the silhouette coefficient, and δ is the threshold of randomness. Further, the two parameters are defined in the process of long-term value judgment. The experimental result of the γ ' value is shown in Table 5, and that of δ ' value is shown in Table 6. The data in the table are the running time of the experiment, and the unit is minutes. “/” indicates that the algorithm runs for more than 24 h without convergence.

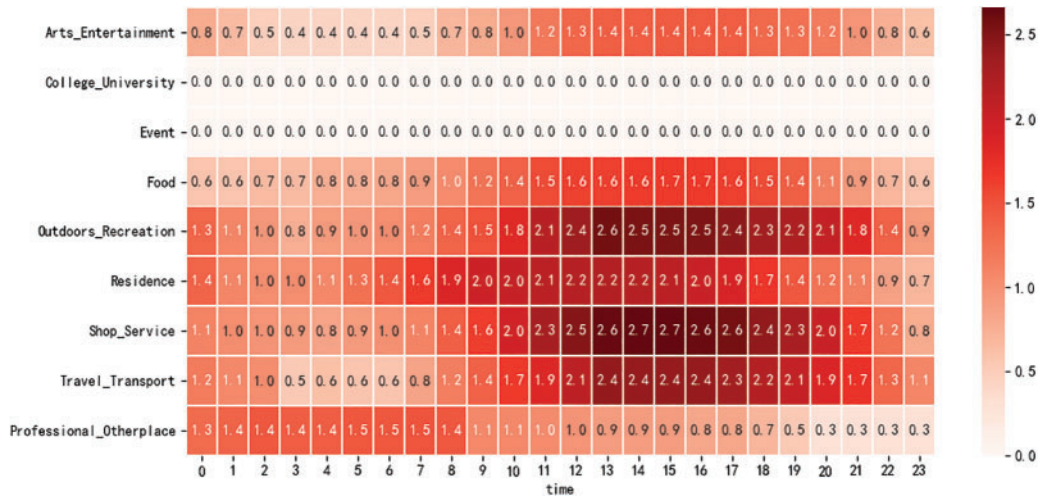


Figure 14: Afternoon outdoor enthusiasts

Table 5: Threshold value table of γ

	$\gamma = 0.5$	$\gamma = 0.6$	$\gamma = 0.7$	$\gamma = 0.8$	$\gamma = 0.9$
UF time-root category	132.9	311.2	1376.4	/	/
UF distance-root category	142.1	193.7	1058.3	/	/

Table 6: Threshold value table of δ

	$\delta = U /5$	$\delta = U /10$	$\delta = U /20$	$\delta = U /50$	$\delta = U /100$
UCVF time-root category	36.7	217.5	1421.8	/	/
UCVF distance-root category	41.3	337.2	1261.5	/	/

The experimental results in Table 5 show that when the value of γ is 0.9, or 0.8, the algorithm cannot converge in the specified time. When the value of γ is 0.7, the algorithm converges in the effective time. Therefore, the final value of parameter γ in the experiment is 0.7.

U in Table 6 represents the number of users in the data set. The experimental results show that when the value of δ is $|U|/100$, that is, the randomness of the cluster cannot be greater than 1%, or the value of δ is $|U|/50$, the algorithm cannot converge within the specified time. When the value of δ is $|U|/20$, the algorithm converges; therefore, the parameter δ is $|U|/20$.

• Experiment result

To verify the effectiveness of the method, the K-means++ algorithm is run N times for each user feature, and the average silhouette coefficient of each cluster result, as well as the randomness between each two cluster results are calculated using formulas (7), (9), and (10). The reinforcement process for each cluster result was then conducted. After the reinforcement process, the average silhouette coefficient of each cluster result and the randomness between each two cluster results were calculated and compared with those before reinforcement. In this experiment, the value of N was set to three, and the specific experimental results are shown in Tables 7 and 8.

Table 7: Comparison table of reinforcement effect of average silhouette coefficient

			SC_K ₁	SC_K ₁ ^E	SC_K ₂	SC_K ₂ ^E	SC_K ₃	SC_K ₃ ^E
data_1	UCVF	time-root category	0.299	0.719	0.327	0.716	0.352	0.714
	UCVF	distance-root category	0.314	0.772	0.311	0.782	0.411	0.726
data_2	UCVF	time-root category	0.293	0.704	0.365	0.728	0.362	0.762
	UCVF	distance-root category	0.351	0.763	0.271	0.719	0.279	0.715
data_3	UCVF	time-root category	0.417	0.724	0.368	0.716	0.436	0.722
	UCVF	distance-root category	0.566	0.739	0.391	0.728	0.511	0.714

Table 8: Comparison table of reinforcement effect of stability

		User	Rand _{k1k2}	Rand _{k1k2}	Rand _{k1k3}	Rand _{k1k3}	Rand _{k2k3}	Rand _{k2k3}	
		(P)	(P)	(P)	(P)	(P)	(P)	(P)	
data_1	UCVF	time-root category	1083	252 (23.3%)	48 (4.4%)	268 (24.7%)	46 (4.2%)	242 (22.3%)	52 (4.8%)
	UCVF	distance-root category	1083	196 (18.1%)	38 (3.5%)	274 (25.3%)	42 (3.9%)	218 (20.1%)	48 (4.4%)
data_2	UCVF	time-root category	5000	1040 (20.8%)	238 (4.8%)	986 (19.7%)	208 (4.2%)	944 (18.9%)	240 (4.8%)
	UCVF	distance-root category	5000	996 (19.9%)	242 (4.8%)	964 (19.3%)	226 (4.5%)	1048 (21.0%)	214 (4.3%)
data_3	UCVF	time-root category	8000	1168 (14.6%)	318 (4.0%)	1080 (13.5%)	374 (4.7%)	1236 (15.5%)	373 (4.7%)
	UCVF	distance-root category	8000	1242 (15.5%)	326 (4.1%)	1286 (16.1%)	360 (4.5%)	1364 (17.1%)	366 (4.6%)

Note: P: Percent.

In Table 7, SC_K₁, SC_K₂, and SC_K₃ are the average silhouette coefficients for each run of the K-means++ algorithm. SC_K₁^E, SC_K₂^E, and SC_K₃^E are the average silhouette coefficients after the enhancement of each clustering result. The results of the experiment show that the average silhouette coefficient has been improved after using the reinforcement process of the clustering results, which means that the reinforcement process is effective.

In Table 8, Rand_{k1k2}, Rand_{k1k3}, and Rand_{k2k3} refer to the randomness between the two cluster results. SC_K₁^E, SC_K₂^E, and SC_K₃^E are the randomness between the two cluster results after the reinforcement process. The stability used in this paper refers to the stability of the user list belongs to each role, that is, changes of the user list for each role when running automatic algorithm multiple times, the smaller the change value, the stronger the stability is. We adopt the criterion proposed by Cohen [23] to validate the stability of the user list. Cohen using statistical method to analyze user behavior, and drew a conclusion that “when the difference value is less than one-tenth of the average value, the experimental result is stable”. The experimental results in Table 8 show that after the reinforcement learning process, the change value between each of the two cluster results was reduced to less than 10 percent.

In summary, the experimental results show that after the reinforcement learning process, not only was the average silhouette coefficient of clustering improved but the stability between each two clustering results was also improved. Therefore, the reinforcement learning process is effective for the optimization of the user role discovery.

RQ3: Compared with existing user role discovery methods, what are the advantages of the method in this study?

- Baselines

Role discovery methods mainly include the following categories: expert experience-based, survey-based, and mathematical analysis-based methods. The studies shown in [Table 8](#) were chosen as baselines after a review of the most recent research in each method category.

- Comparison dimension

Automation [24] and re-usability [25] of methods play an important role in improving the efficiency of software development. Whether the result is stable or influenced by personal experience have an important impact on the re-usability of the method. Therefore, four comparison dimension is defined in this study to validate the effective of the method, which are automation, personal experience influence, re-usability, and result stability.

- Result analysis

The method proposed in this study is compared with the baseline methods. The comparison results are presented in [Table 9](#).

As shown in the table, the expert experience-based method is a no automated process of manual judgment, which is strongly influenced by the personal experience of experts. With the complexity of the scene, the difficulty of manual judgment will be greatly increased, the method is difficult to reuse, and the result is unstable [7]. Therefore, this method has not been widely used in role discovery.

Survey-based methods can discover the role of users through a survey of the literature [8] or users [9]. Although the results obtained by this method are stable, the method is not automatic. It requires a large amount of human resources to investigate the literature and users. Moreover, when the application scenarios or user groups change, the method is difficult to reuse directly, and the research activities need to be repeated.

Table 9: Comparison results table of the methods

Method	Related works	Automation	Personal experience influence	Re-usability	Result stability
Expert experience-based method	[7]	No	Yes	Hard	No
Survey-based method	[8,9]	No	No	Hard	Yes
Mathematical analysis-based method	[10–13]	Yes	No	Easy	No
This method	This paper	Yes	No	Easy	Yes

Mathematical analysis is a general term that refers to methods for discovering user roles using clustering analysis, statistical analysis, social network analysis, and other mathematical algorithms. Because it is difficult to communicate with users directly in mobile applications, and user activity data are rich, this type of method has been widely used [10–13]. This type of method can realize automation and easy re-usability, and the running results are not affected by personal experience.

However, the current research does not consider the stability of the classification results. From the experiments in this study, which run the K-means++ algorithm numerous times, the user set in the same role is quite different, which can reach even more than 20% of the total number of users, as shown in Table 8.

The method proposed in this paper is easily automated and reusable, and the running results are not affected by personal experience. The method uses reinforcement learning to improve the stability of the cluster results.

6 Conclusions

The discovery of user roles from user behavior data in mobile applications is necessary for an in-depth and comprehensive understanding of users. A user role discovery and optimization method based on K-means++ and reinforcement learning were proposed in this paper. First, an algorithm was proposed to vector user features from user behavior data. Second, a method based on the K-means++ algorithm was used to discover user roles through the clustering of user features. On this basis, a method based on reinforcement learning was proposed to optimize the clustering results to improve the stability of the results. A set of experiments were conducted. The results show that the method can improve the effect of clustering by 1.5~2 times, and the stability of the cluster results are improved about 3 times of the original, which indicates the effectiveness of the method.

Funding Statement: This work is supported by the National Natural Science Foundation of China under Grant No. U1504602.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. China Internet Information Center (CNNIC) (2020). 45nd Statistical Report on the Development of China's Internet. http://www.cac.gov.cn/2020-04/27/c_1589535470378587.htm/2021-1-29.
2. Yu, Z. (2011). *Location-based social networks: Users*, pp. 243–276. Springer, New York.
3. He, J., Li, X., Liao, L., Wang, M. (2018). Inferring continuous latent preference on transition intervals for next point-of-interest recommendation. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 741–756. Springer, Cham.
4. Hacker, J. V., Bodendorf, F., Lorenz, P., Carayannis, E., Heisig, P. (2017). A framework to identify knowledge actor roles in enterprise social networks. *Journal of Knowledge Management*, 21(4), 817–838. DOI 10.1108/JKM-10-2016-0443.
5. Yu, S. K. (1995). *Modelling strategic relationships for process reengineering (Ph.D. Thesis)*. University of Toronto, Department of Computer Science.
6. Zhang, W., Zhu, J. (2020). Review of user roles in professional virtual communities. *Journal of Modern Information*, 40(7), 167–177.
7. Liu-Thompkins, Y., Maslowska, E., Ren, Y., Kim, H. (2020). Creating, metavoicing, and propagating: A road map for understanding user roles in computational advertising. *Journal of Advertising*, 49(4), 1–17. DOI 10.1080/00913367.2020.1795758.
8. Hacker, J. V., Bodendorf, F., Lorenz, P., Carayannis, E., Heisig, P. (2017). A framework to identify knowledge actor roles in enterprise social networks. *Journal of Knowledge Management*, 21, 817–838. DOI 10.1108/JKM-10-2016-0443.

9. Oxford University (2021). Digital Engagement in GLAM (Gardens, Libraries & Museums). <https://glam.web.ox.ac.uk/digital-engagement-glam #collapse1060891>.
10. Hacker, J., Riemer, K. (2020). Identification of user roles in enterprise social networks: Method development and application. *Business & Information Systems Engineering*, 63(4), 367–387. DOI 10.1007/s12599-020-00648-x.
11. Guo, W., Zheng, Q., An, W., Peng, W. (2017). User roles and contributions during the new product development process in collaborative innovation communities. *Applied Ergonomics*, 63(10), 106–114. DOI 10.1016/j.apergo.2017.04.013.
12. Toral, S. L., Martinez-Torres, M. R., Barrero, F. (2010). Analysis of virtual communities supporting OSS projects using social network analysis. *Information and Software Technology*, 52(4), 296–303. DOI 10.1016/j.infsof.2009.10.007.
13. Fueller, J., Hutter, K., Hautz, J., Matzler, K. (2014). User roles and contributions in innovation contest communities. *Journal of Management Information Systems*, 31(1), 273–307. DOI 10.2753/MIS0742-1222310111.
14. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J. et al. (2019). Human-level control through deep reinforcement learning. *Nature*, 518(7540), 529–533. DOI 10.1038/nature14236.
15. Li, J., Ba, J. (2020). Improved DBSCAN algorithm based on MCMC. *Computer Engineering and Design*, 41(1), 122–127.
16. Das, S., Christopher, J., Apan, A., Choudhury, M. R., Chapman, S. et al. (2021). UAV-thermal imaging and agglomerative hierarchical clustering techniques to evaluate and rank physiological performance of wheat genotypes on sodic soil. *ISPRS Journal of Photogrammetry and Remote Sensing*, 173, 221–237. DOI 10.1016/j.isprs.2021.01.014.
17. Steinley, D. (2011). K-means clustering: A half-century synthesis. *British Journal of Mathematical & Statistical Psychology*, 59(1), 1–34. DOI 10.1348/000711005X48266.
18. Manu, A., Ragesh, J., Arindam, P. (2015). K-means++ under approximation stability. *Theoretical Computer Science*, 588, 37–51. DOI 10.1016/j.tcs.2015.04.030.
19. Annibale, P., Bogdan, D., Rocco, O. (2013). How to effectively use topic models for software engineering tasks? An approach based on genetic algorithms. *35th International Conference on Software Engineering*, pp. 522–531. San Francisco, CA.
20. Yang, D., Zhang, D., Zheng, V. W., Yu, Z. (2015). Modelling user activity preference by leveraging user spatial temporal characteristics in LBSNs. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(1), 129–142. DOI 10.1109/TSMC.2014.2327053.
21. Yang, D., Zhang, D., Qu, B. (2016). Participatory cultural mapping based on collective behavior data in location-based social networks. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 7(3), 1–23.
22. Kuo, M. H., Chen, L. C., Liang, C. W. (2009). Building and evaluating a location-based service recommendation system with a preference adjustment mechanism. *Expert Systems with Applications*, 36(2), 3543–3554. DOI 10.1016/j.eswa.2008.02.014.
23. Cohen, J. (2013). *Statistical power analysis for the behavioral sciences*. Pittsburgh, Pennsylvania, USA: Academic Press. https://www.researchgate.net/publication/336869743_Asynchronous_Methods_for_Model-Based_Reinforcement_Learning.
24. Dikmen, M., Li, Y., Ho, G. (2020). The burden of communication: Effects of automation support and automation transparency on team performance. *IEEE International Conference on Systems, Man, and Cybernetics*, pp. 2227–2231. Toronto, Ontario, Canada, IEEE.
25. Liu, C. (2021). Research on software development mode and application of reusable framework based on neural network algorithm. *6th International Conference on Communication and Electronics Systems*, pp. 1586–1589. Coimbatore, India.