



HMGR: A Hybrid Model for Geolocation Recommendation

Ruwang Wen¹, Zhengxiang Cheng¹, Weixuan Mao¹, Zhuolin Mei^{1,2,3}(✉),
Jiaoli Shi^{1,2,3}, and Xiao Cheng¹

¹ School of Computer and Big Data Science, Jiujiang University, Jiujiang, Jiangxi, China
meizhuolin@126.com

² Jiujiang Key Laboratory of Network and Information Security, Jiujiang, Jiangxi, China

³ Institute of Information Security, Jiujiang University, Jiujiang, Jiangxi, China

Abstract. The Geosocial networks integrate geographical location information into traditional social networks, bridging the gap between people's real-life experiences and the virtual world. As a significant application of geosocial networks, location recommendation suggests places that individuals may find interesting, offering valuable references for their travels and greatly enhancing their lives. Consequently, the challenge of recommending relevant locations to users from a vast pool of geographic options has become a prominent topic in academic research. Collaborative filtering algorithms stand as one of the classic solutions in the field of recommendations. While they partially address the problem of information overload, they often encounter a common obstacle known as the cold start problem. To overcome this issue, this study makes two primary contributions: Firstly, it proposes the utilization of Markov chains to mitigate the cold start problem. Secondly, it introduces a hybrid recommendation model called HMGR for location-based recommendations, which effectively enhances the accuracy of suggestions. We evaluate the efficacy of the Markov chain and HMGR model through extensive experimentation. The results demonstrate that the implementation of Markov chains successfully alleviates the cold start problem, and our HMGR model significantly improves the precision of recommendations.

Keywords: Geographical Location · Collaborative Filtering · Markov Chain · Cold Start

1 Introduction

In the context of information overload [1, 2], users are faced with an overwhelming amount of information and options, making it difficult to accurately find the desired information or make the best choices. In such a situation, both probabilistic relational database hypothesis querying [3] and recommendation systems can help users more effectively obtain useful information and address uncertainty. However, recommendation systems have clear advantages in dealing with information overload. They have the capability to provide personalized recommendations based on users' historical behavior and interests. This means that users will receive content that matches their preferences

and needs, greatly enhancing the efficiency and satisfaction of information retrieval. Recommendation systems can leverage image processing techniques [4, 5] to extract content features from images that users post on social media. Furthermore, they can fully utilize users' historical data for personalized recommendations. By analyzing users' behavior data, such as click history, purchase records, search history, comments, and favorites, recommendation systems can gain deep insights into users' preferences, habits, and needs, further increasing user satisfaction and click-through rates.

However, recommendation systems face a significant challenge in practical applications, known as the cold-start problem [6]. The cold-start problem refers to the lack of sufficient historical data for new users or new items, making it difficult for the recommendation system to accurately predict their preferences. In real-world applications, new users and items are inevitable and may occur frequently. If the recommendation system cannot effectively address the cold-start problem, these new users and items will not receive personalized recommendations, which can lower user experience and the utility of the recommendation system.

Additionally, recommendation accuracy is of utmost importance for the value of a recommendation system. Accurate recommendations not only increase user trust in the recommendation system but also significantly improve user satisfaction and click-through rates. On the contrary, poor recommendation accuracy may lead to user dissatisfaction and even prompt users to discontinue using the recommendation service, resulting in potential economic losses for businesses. Therefore, improving recommendation accuracy is a key goal in recommendation system research, especially in highly competitive market environments, where precise recommendations are essential for gaining a competitive advantage. While some new solutions have shown promising results in addressing the cold-start problem and improving accuracy in recent years, there are still limitations that need to be further addressed.

The core idea of content-based recommendation systems [7] is to utilize item content information for personalized recommendations, which builds upon the advancements in information filtering technology. By employing machine learning techniques, these systems can extract user preferences from the descriptive features of the items, thereby alleviating the cold-start problem without relying solely on user ratings. However, it is worth noting that many existing content-based recommendation methods [8–10] face challenges in uncovering users' latent interests and encounter difficulties in accurately extracting item features, which can ultimately lead to lower recommendation accuracy.

In order to uncover a user's potential interests, a collaborative filtering recommendation [11] was proposed. This method assumes that users with similar interests may have similar preferences toward similar items. The core idea is to use a neighbor-based recommendation algorithm that leverages similarity measures between users or items and historical behavior data [12]. However, traditional collaborative filtering recommendation systems are facing challenges due to the exponential growth of data from various applications and services, leading to issues like the cold-start problem [6]. To address the cold-start problem and enhance recommendations, researchers have explored utilizing user social relationships for recommendation [13–15]. Friend recommendations can increase trust and help alleviate the cold-start problem. However, the collaborative filtering algorithm is vulnerable to data bias, as it tends to favor popular items that receive

more attention and evaluations. This bias can lead to recommendation results that are biased towards these popular items, overlooking individualized preferences.

To enhance recommendation efficiency, hybrid recommendation [16] approaches have been proposed [17–19]. The main objective of hybrid recommendation algorithms is to leverage the strengths of various recommendation algorithms while mitigating their limitations. By integrating different approaches, hybrid recommendations can effectively address the cold-start problem and provide diverse recommendations. However, it is important to acknowledge that current hybrid recommendation methods may encounter challenges such as increased complexity, longer recommendation times, and difficulties in achieving a balanced integration. Despite these challenges, hybrid recommendation techniques have demonstrated promising results in improving overall recommendation performance.

Considering the limitations of previous approaches, we propose a hybrid model for geolocation recommendation, namely HMGR. In our model, we first assess the user's historical data volume. If the user is identified as new, the system employs a Markov chain for personalized recommendations. Conversely, for existing users, collaborative filtering is utilized for recommendation generation. Through extensive simulation experiments, we have observed that HMGR effectively addresses the challenges associated with the cold-start problem and significantly improves the accuracy of recommendation results.

We summarize our proposed HMGR model based on the following three key aspects.

- (1) We mitigate the cold start problem in geolocation-based recommendations by utilizing Markov chains.
- (2) We propose a hybrid recommendation model called HMGR for geolocation-based recommendations, which combines the use of Markov chains and collaborative filtering.
- (3) Conduct extensive experiments to evaluate the efficiency and effectiveness of the HMGR model, along with a thorough analysis.

The remaining sections of this paper are structured as follows: Sect. 2 provides an overview of related research. Section 3 introduces the essential preliminary knowledge. Section 4 describes the construction of our proposed model. In Sect. 5, we present the experimental evaluation. Finally, Sect. 6 summarizes the findings and conclusions of this study.

2 Related Work

Traditional recommendation algorithms can be classified into three main categories: content-based recommendation algorithms [7], collaborative filtering recommendation algorithms [11], and hybrid recommendation algorithms [16].

2.1 Content-Based Recommendation

Content-based recommendation, as the name implies, is a method that recommends items based on their similarity in terms of content. It can be considered one of the founding approaches in recommendation algorithms, as it relies on the premise that items with

similar content to a user's previously preferred items are likely to be of interest [20]. The concept of content-based recommendation originated from research in information retrieval [7]. With the rapid advancement of information retrieval and the widespread adoption of applications like email, content-based recommendation has found extensive application in the field. Content-based recommendation primarily involves the description of item content features and user profiles (such as interests and preferences). It efficiently filters and selects more valuable information, offering several advantages such as high recommendation efficiency, not requiring user evaluations or additional information, and mitigating the cold-start problem associated with new items [8–10].

For example, literature [9] proposes a content-based recommendation system model is proposed. This model tackles the cold start problem by employing content-based methods, which involves making recommendations for new users or new items during the recommendation process. Moreover, the model takes into account the incorporation of features like security, reliability, and transparency in career recommendation, assisting students in making informed career choices. However, despite these advancements, these studies still encounter challenges, such as limited diversity in recommended content and complexities in extracting multiple item features.

2.2 Collaborative Filtering Recommendation

In order to address the issue of extracting multiple-item features, collaborative filtering recommendation has emerged as an alternative approach [21]. The core principle of collaborative filtering is similarity, which involves categorizing users and items based on their similarity and making recommendations accordingly. Collaborative filtering recommendation has found widespread application in various domains, including e-commerce [22], session recommendation [23], and article recommendation [24]. In collaborative filtering, two extensively studied methods are user-based collaborative filtering and item-based collaborative filtering [25]. User-based approaches predict user ratings based on the similarity of rating behaviors among users, while item-based approaches predict user ratings based on the similarity between predicted items and the actual items chosen by users. Due to the distinct principles of these two methods, their performances vary in different application scenarios. User-based recommendations tend to be more socialized, reflecting the popularity of items within the interest group to which the user belongs, while item-based recommendations are more personalized, reflecting the user's individual interest preferences.

In collaborative filtering recommendation, the cold-start problem is also present, and many researchers have proposed several solutions to address it. For example, Zhang et al. [26] identified the significant impact of different datasets, user attributes, the number of nearest neighbors, and the number of items on recommendation results. They developed an optimized user-based collaborative filtering recommendation system that tackles the issue of varying user rating scales by standardizing the original user data. By incorporating weighted user attributes and linear combinations with user rankings, they enhanced the overall user similarity. Experimental results demonstrated that this algorithm successfully mitigated the influence of cold-start problems and provided accurate recommendations.

2.3 Hybrid Recommendation

Due to the inherent limitations of individual recommendation algorithms, a hybrid recommendation has emerged as an approach to improve the overall recommendation performance by combining different models to complement their shortcomings [18, 19, 27]. Hybrid recommendation models blend two or more recommendation algorithms, thereby mitigating issues such as user cold-start and item cold-start, and overcoming the limitations of single algorithms.

For example, Zhang et al. [27] proposed a hybrid recommendation algorithm based on collaborative filtering and video genetics. The algorithm first constructs a user-item matrix, calculates user similarity, and performs clustering using k-means to generate a recommendation list. By analyzing the genetic structure of videos and combining style preferences and regional preferences, genetic preferences are formed. The weights of these genetic preferences are determined through linear regression. The objects are then ranked based on their degree of genetic preference, and the top-ranked objects are selected as the final recommendations. The algorithm combines the recommendation results from collaborative filtering and video genetics by assigning weights to each recommendation. However, they encounter issues such as high complexity and computational overhead.

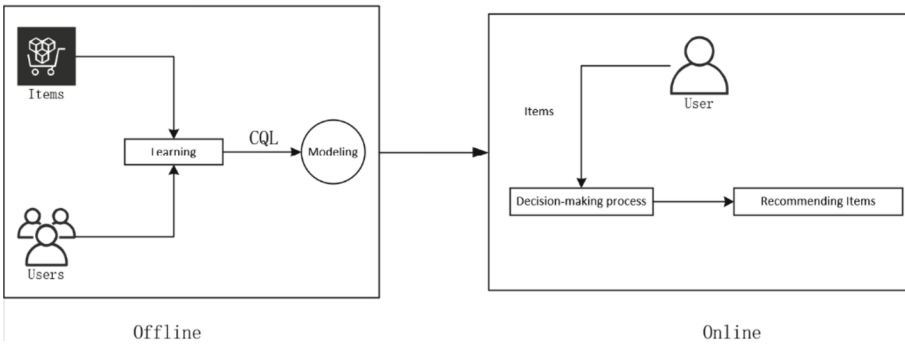


Fig. 1. System model

3 Preliminaries

Markov chain, named after the Russian mathematician Markov, is a mathematical model that represents a stochastic process. It is used to study the behavior and predict the future states of a system based on the current state, assuming that the future states only depend on the present state and are independent of the past states. This modeling technique is widely used in various fields, including probability theory, statistics, physics, computer science, and economics, to analyze and understand the dynamics of complex systems with probabilistic transitions.

Definition 1: Markov Chain. A Markov chain is a stochastic process that satisfies the property that the state of the system at time $t + 1$ depends only on the state at time t . It is

characterized by the Markov property, which states that the future states are independent of the past states given the present state. In other words, the state transitions in a Markov chain are memoryless, and the probability of transitioning to a future state depends solely on the current state.

4 Construction of HMGR

Definition 2: Critical Queue Length (CQL). Once the user geolocation queue length reaches a specific threshold, the accuracy of the collaborative filtering recommendation algorithm begins to outperform that of the Markov chain recommendation.

The system model is depicted in Fig. 1. Initially, the system learns from a large dataset of user-profiles and check-in histories obtained from offline data. During the learning phase, two algorithms, namely the Markov chain algorithm and collaborative filtering recommendation algorithm, are applied to generate recommendations for users. By comparing the accuracy of these two algorithms, the system aims to determine the threshold point CQL where the Markov chain algorithm's accuracy starts to become lower than that of the collaborative filtering recommendation system. Subsequently, the system models this threshold point CQL and incorporates it into the decision-making process. When an online user interacts with the system and provides their check-in history, the system compares the length of the user's check-in history with the value of CQL. Based on this comparison, the system distinguishes between new users and existing users. If the length of the user's check-in history is less than or equal to CQL, the system identifies the user as a new user and employs the Markov chain algorithm to recommend items. Conversely, if the length of the user's check-in history exceeds CQL, the system recognizes the user as an existing user and utilizes the collaborative filtering recommendation algorithm to provide personalized recommendations.

4.1 Markov Chain in HMGR

Markov chains rely primarily on the current location of the user and the probabilities of transitioning between locations, rather than relying heavily on extensive historical data. In cold start scenarios, where there is limited data for personalized user modeling, Markov chains can utilize the available data to predict the user's next potential location and provide reasonably accurate recommendations. Therefore, Markov chains can be used to recommend geographical locations. Specifically, the predictive power of Markov chains can be leveraged to offer recommendations for new users.

Below, we present the pertinent definitions of Markov chains in our HMGR model.

Definition 3: State Pair. A state pair refers to the transition from one geographical location state to another. It represents the change in the geographical location, where the transition from state l_i to state l_j is denoted as $l_i \rightarrow l_j$.

Definition 4: State Transition Probability. In the context of geolocation, a transition from one geolocation state to another is referred to as a geolocation state transition. This process captures the relationship between state transitions and time as geolocation states change over time. The likelihood of a geolocation transitioning from one state to another

over a specific time period is quantified as the state transition probability, as illustrated in Formula 1.

$$P_{ij} = \frac{C(l_i \rightarrow l_j)}{C} \quad (1)$$

where $C(l_i \rightarrow l_j)$ represents the number of occurrences in which the geographical location transitions from state i to state j in the user's interaction with the geographical location queue, while C represents the total number of transitions for state l_i .

Definition 5: Transition Probability Matrix. In the context of user geographical location states, where there are n possible states denoted by $\{l_1, l_2, l_3, \dots, l_n\}$, the transition probability from state l_i to state l_j is represented as p_{ij} . These transition probabilities are combined to form a transition probability matrix, as shown in Formula 2.

$$P = \begin{pmatrix} p_{11} & \cdots & p_{1n} \\ \vdots & \ddots & \vdots \\ p_{m1} & \cdots & p_{mn} \end{pmatrix} \quad (2)$$

Typically, a standard Markov chain model can be represented using a triplet. However, in this paper, we employ a modified Markov chain model that is represented using a quadruplet, as depicted in Formula 3.

$$MC < L, P, M_t, Q > \quad (3)$$

where L represents the set of states in the model, denoted as $L = \{l_1, l_2, l_3, \dots, l_n\}$, where n represents the number of states. P represents the state transition matrix, denoted as $P = [p_{ij}]_{n \times n}$. The element p_{ij} represents the probability of transitioning from state l_i at time t to state l_j at time $t + 1$. M_t represents the probability distribution of a user's state at time t , denoted as $M_t = \{m_1, m_2, m_3, \dots, m_n\}$, where m_i is the probability of being at l_i . Q represents the set of geographical locations that the user has not visited before, denoted as $Q = \{q_1, q_2, q_3, \dots, q_m\}$, where m represents the set of unvisited locations.

The performance of a Markov chain recommendation system relies on the accurate calculation of state transition probabilities. If the state definitions are not precise or there are errors in the calculation of transition probabilities, it may lead to a decline in the system's performance. Therefore, the calculation of state transition probabilities is crucial. In our HMGR model, we use a typical Markov chain that utilizes the state transition matrix based on the current geographical location at time t to predict the user's geographical location at time $t + 1$. This approach helps us address the cold start problem, where limited data for personalized user modeling is available. By computing the probability distribution M_t for the user's current geographical location, we can use Formula 4 to calculate the state transition probability distribution M_{t+1} for the user at time $t + 1$.

$$M_{t+1} = M_t \times P \quad (4)$$

In our HMGR model, the Markov chain begins by collecting the user's historical geographical location data and organizing it into a time series. Each specific location

within the sequence is treated as a state. By analyzing the user’s historical sequence of geographical locations, we calculate the probabilities of transitioning between adjacent locations, capturing the likelihood of moving from one location to another. It’s worth noting that our approach utilizes a one-step Markov chain, where each geographic location transition occurs in a single step from one location to another. This simplification allows us to effectively model and predict user behavior in the context of geographic location recommendations. Finally, we compute the probability distribution M_{t+1} and compare it to the set Q of geographical locations the user has not visited yet, thereby predicting the user’s geographic location in the next time step.

The following Example 1 demonstrates the recommendation process using Markov chains in HMGR.

Table 1. User behavior trajectory.

User	Trajectory
u_1	$l_1 \rightarrow l_2 \rightarrow l_3 \rightarrow l_1 \rightarrow l_3$
u_2	$l_2 \rightarrow l_3$
u_3	$l_1 \rightarrow l_2 \rightarrow l_1$

Example 1: As shown in Table 1. In the table, u_1, u_2, u_3 represents three users, $L = \{l_1, l_2, l_3\}$ represents three sets of geographic locations, and $l_1 \rightarrow l_2, l_2 \rightarrow l_3, l_3 \rightarrow l_1, l_1 \rightarrow l_3$ represents the state pairs. According to Formula 1, we can calculate $p_{12} = 2/3, p_{13} = 1/3, p_{21} = 1/3, p_{23} = 2/3, p_{31} = 1$ and $p_{32} = 0$. Therefore, we

obtain the state transition matrix $P = \begin{pmatrix} 0 & 2/3 & 1/3 \\ 1/3 & 0 & 2/3 \\ 1 & 0 & 0 \end{pmatrix}$.

Here, we set the probability of transitioning to the current location as zero. After obtaining the transition probability matrix, we can calculate the initial state probability distribution for the geographic location of user u_2 at time t . Given that the geographic location of user u_2 at time t is l_3 , the initial state probability distribution of the user’s state as $M_t = (0 \ 0 \ 1)$. Next, we can calculate the geographic location state transition probabilities for $M_{t+1} = (1 \ 0 \ 0)$ using Formula 4.

Upon analyzing the results, it becomes evident that the HMGR model employs a Markov chain to predict user u_2 ’s level of interest in different geographical locations. The model indicates an interest level of 1 for location l_1 , 0 for location l_2 , and 0 for location l_3 . Furthermore, when compared to the set $Q = \{l_1\}$, the prediction suggests that the most probable destination for user u_2 in the next stage is l_1 .

4.2 Collaborative Filtering in HMGR

In cases where users have limited historical data, Markov chains can provide relatively accurate recommendations. Markov chains primarily rely on the current location of the user and the transition probabilities between locations, rather than relying heavily on

extensive historical data. This enables Markov chains to predict the user's next potential location using the available data and offer reasonably accurate recommendations. However, the Markov chain model may not fully capture user preferences and personalized needs, as it only considers transition probabilities and lacks direct consideration of user interests and other contextual information. On the other hand, collaborative filtering algorithms can leverage user similarities and abundant historical behavior data to predict locations that users may like. Therefore, when users have a rich amount of historical data, collaborative filtering recommendations usually achieve higher recommendation accuracy.

In traditional collaborative filtering, there is a bias in the recommendation results due to imbalanced user interactions with geographical locations. The interactions tend to favor highly active users and popular locations, leading to a biased recommendation. The overwhelming interactions from active users and popular places overshadow other potentially relevant locations. This bias undermines the diversity of recommendations, making the recommendation system resemble more of a search engine rather than a personalized system.

A User-IIF algorithm [28] has been proposed in the literature to effectively address the issue of popularity bias, where popular items tend to become even more popular while less popular items are overlooked. In the context of geographical location recommendation, it is assumed that users exhibit similar behavior towards different locations. In our HMGR model, we have made slight modifications to its definition in order to mitigate the impact of popular locations. The algorithm introduces a modification to the user similarity calculation. We have further refined its definition to eliminate the influence of popular geographical locations. The calculation formula is presented as Formula 5.

$$w_{ij} = \frac{\sum_{l \in N(i) \cap N(j)} \frac{1}{\ln(1+N(l))}}{\sqrt{N(i)} \times N(j)} \quad (5)$$

where $N(i)$ represents the set of geographical locations that user i has provided positive feedback on. $N(j)$ represents the set of geographical locations that user j has provided positive feedback on. l represents the set of geographical locations for which both user i and user j have provided positive feedback. $N(l)$ represents the set of users who have given positive feedback on the geographical locations in set l .

After obtaining the similarity between users, the collaborative filtering recommendation algorithm recommends the geographical locations visited by the K most similar users to the target user. This recommendation is determined using Formula 6.

$$Score(u, i) = \sum_{v \in S(u, k) \cap N(i)} w_{uv} \times r_{vi} \quad (6)$$

where $S(u, k)$ represents the top k users who have the closest interests to user u . $N(i)$ is the set of users who have interacted with geographical location i . w_{uv} denotes the similarity of interests between user u and user v . r_{vi} represents the interest of user v in geographical location i . Since single-action implicit feedback data is used, all r_{vi} values are considered as 1.

The recommendation process of collaborative filtering is as follows: Firstly, collect the historical behavior data of users regarding geographical locations. Then, calculate

the similarity between users to select the top K users who are most similar to the target user. These selected users are referred to as candidate users. Finally, based on the degree of association between the candidate users and the target user, considering factors such as locations already viewed by the target user, sort and filter the recommendation results to obtain the final recommendations.

The following Example 2 demonstrates the recommendation process using collaborative filtering in HMGR.

Example 2: We continue to use the data from Table 1 to predict the next geographical location that user u_3 is likely to visit. After calculating the similarity between users using Formula 5, we obtained the following similarity values: $w_{31} = 0.67$, $w_{32} = 0.29$. In this example, assuming $K = 1$, the candidate user is u_1 . By comparing the historical data of users u_2 and u_3 , we found that user u_2 has previously visited location l_3 , while user u_3 has not visited l_3 yet. Next, using Formula 6, we calculated the $Score(u_3, l_3) = 0.67$, and therefore, we recommend geographical location l_3 to user u_3 .

5 Experiments

In the experiment, we conducted a comparative study of collaborative filtering recommendation, Markov Chain recommendation, and our HMGR recommendation model using evaluation metrics such as accuracy, recall rate, and F-value. These solutions were implemented in Java language on a personal computer equipped with an AMD Ryzen 7 5800H CPU and 16GB RAM. We utilized the Foursquare dataset, which contains offline data for location recommendation. Specifically, we selected user geographical location interaction sequences from different areas of New York, USA. 80% of the data was used as the training set, and 20% of the data was used as the test set, for constructing the user similarity matrix and conducting recommendation evaluations.

Let R_u represent the list of recommended geographical locations calculated by the model based on user behavior in the training set, and T_u represent the list of geographical locations that the user will actually visit in the future based on the test set. The primary evaluation method is as follows:

(1) Precision measures the proportion of accurately recommended geographical locations among the samples predicted as other locations. This metric calculates the ratio of correctly classified positive samples to the total number of samples classified as positive by the classifier. It indicates how many locations in the predicted recommendation list are actually visited by the user in the future. The precision is calculated using Formula 7.

$$Precision = \frac{\sum_{u \in U} R_u \cap T_u}{\sum_{u \in U} R_u} \quad (7)$$

(2) Recall measures the proportion of correctly recommended locations among the recommended results. This metric reflects how many of the locations that the user will actually visit in the future are accurately predicted by the recommendation algorithm. It is calculated by dividing the number of correctly classified positive samples by the total number of actual positive samples. The recall can be computed using Formula 8.

$$Recall = \frac{\sum_{u \in U} R_u \cap T_u}{\sum_{u \in U} T_u} \quad (8)$$

(3) The F1 score, also known as the balanced F-score, is a metric that balances both precision and recall, providing an overall measure of the model's performance. It can be seen as the harmonic mean of precision and recall. The F1 score is calculated using Formula 9.

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (9)$$

5.1 Determining the Recommendation Length L

Based on Fig. 2, Fig. 3 and Fig. 4, the HMGR model demonstrates improved accuracy compared to the individual collaborative filtering and Markov chain models. The recall rate of the HMGR model initially falls between the collaborative filtering and Markov chain recommendations, but as the number of recommended locations increases, the HMGR model surpasses both collaborative filtering and Markov chain recommendations. The F1 score of the HMGR model consistently falls between collaborative filtering and Markov chain recommendations. As the number of recommended locations increases, the accuracy also improves. Therefore, we set the recommendation length to 5. When the recommendation length is 5, the F1 score of our HMGR model is slightly lower than that of the collaborative filtering, but this difference can be considered negligible.

As the recommended length increases, the decline in precision, recall, and F1 score can be attributed to the presence of varying quality of recommended locations within longer recommendation lists. Lower-quality recommendations may be mixed with high-quality ones, which can adversely affect the precision and recall of the recommendations. The F1 score serves as a comprehensive metric that takes into account both precision and recall, providing an assessment of the overall model performance. When precision and recall decrease, the F1 score naturally decreases as well.

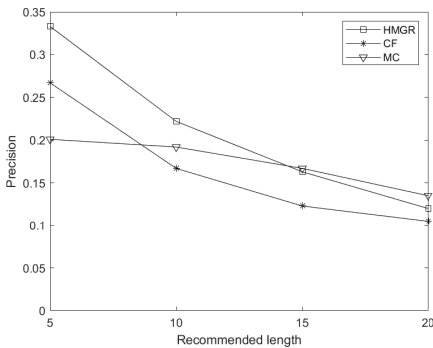


Fig. 2. Precision

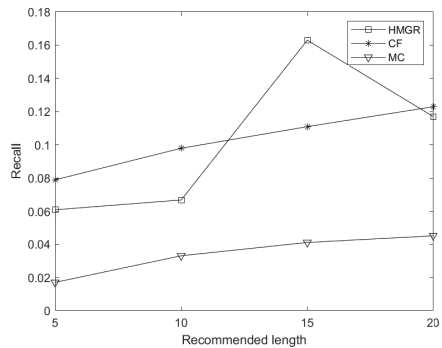


Fig. 3. Recall

5.2 Solution to the Cold-Start Problem

As shown in Fig. 2, our proposed HMGR model achieves significantly higher precision compared to the Markov Chain and Collaborative Filtering recommendations when

the recommendation length is set to 5. This indicates that our HMGR model effectively addresses the cold-start problem, where limited user data is available for accurate recommendations. By combining the strengths of the Markov Chain and Collaborative Filtering approaches, the HMGR model provides more accurate and personalized recommendations even in scenarios with sparse user history.

In the HMGR model, we combine the Markov chain and collaborative filtering recommendation methods. When the user's historical data is limited, the Markov chain can provide relatively higher recommendation accuracy. The Markov chain primarily relies on the transition probabilities between the user's current and future locations, rather than relying on extensive historical data. In such cases, the Markov chain can effectively predict the user's next likely location and offer accurate recommendations. However, the Markov chain model has limitations in capturing user preferences and personalized needs since it only considers transition probabilities and overlooks user interests and contextual information. On the other hand, collaborative filtering algorithms can leverage the similarity between users and their extensive historical behavior data to predict locations that a user is likely to prefer. Thus, when users have abundant historical data, collaborative filtering algorithms generally provide higher recommendation accuracy. Our HMGR model combines both approaches, leading to improved accuracy in recommendations. By leveraging the strengths of both the Markov chain and collaborative filtering, we can provide more accurate and personalized recommendations to users, overcoming the limitations of each individual method.

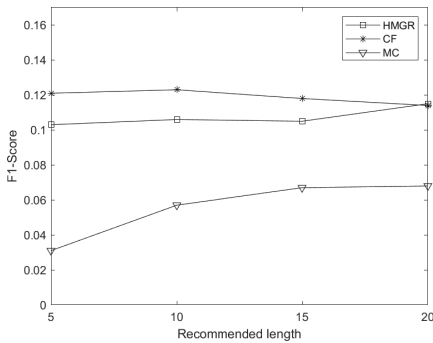


Fig. 4. F1-score

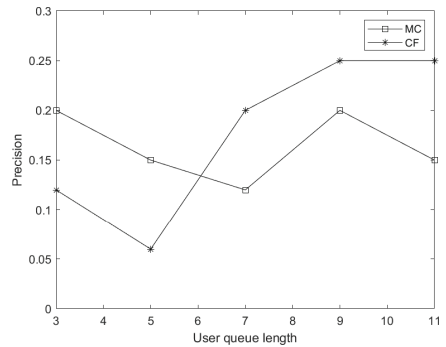


Fig. 5. User queue length

5.3 The Determination of New Users

As shown in Fig. 5. When the length of the user queue is less than 6, the accuracy of the Markov chain is higher than that of collaborative filtering. However, when the length exceeds 6, the accuracy of the Markov chain is lower than that of collaborative filtering. This suggests that for users with a relatively short history of interactions (queue length < 6), the Markov chain approach performs better in terms of recommendation accuracy. On the other hand, for users with a longer history of interactions (queue length > 6), collaborative filtering yields higher accuracy in recommendations. Therefore, the HMGR model takes into account the length of the user queue to determine whether a user

is considered new or existing and accordingly selects the appropriate recommendation method to provide accurate and personalized recommendations.

This is because the Markov chain recommends geographic locations to users based on probabilistic transitions, without directly considering user interests. As the user queue length increases, the accuracy of the Markov chain does not necessarily improve. On the other hand, collaborative filtering recommendation utilizes users' historical visit data to observe their preferences and the likelihood of revisiting specific geographic locations. As the user queue gradually grows, the user similarity matrix becomes more refined, resulting in increased accuracy. When determining whether a user is new, we can compare the user queue length at which the accuracy of collaborative filtering recommendation surpasses that of the Markov chain through testing.

6 Conclusion

In this paper, we present a novel and efficient hybrid recommendation algorithm, the HMGR model. By combining the Markov chain algorithm with collaborative filtering, the HMGR model successfully overcomes the limitations of traditional recommendation methods, especially in terms of handling cold-start issues and providing personalized recommendations. The experimental results demonstrate that the HMGR model achieves significant improvements in various scenarios and exhibits outstanding performance in terms of accuracy and efficiency. Of particular significance is the HMGR model's innovative approach to address the cold-start problem in collaborative filtering recommendations. By leveraging the Markov chain algorithm to predict users' latent interests, the HMGR model can deliver effective recommendation services even for new users in the system. For emerging recommendation systems, solving the cold-start problem is critical, and the remarkable performance of the HMGR model brings renewed hope in this aspect. However, we also acknowledge that there is still room for further refinement in the HMGR model. Future research could explore more intricate Markov chain models to enhance the accuracy of predicting changes in users' interests. Additionally, integrating other recommendation algorithms into the HMGR model presents an enticing avenue for further enhancing the recommendation system's overall performance.

In conclusion, this study firmly establishes the effectiveness and potential of the HMGR model in the user geolocation recommendation system. By blending the power of the Markov chain and collaborative filtering methods, we successfully address the cold-start problem and substantially improve the recommendation system's accuracy. We believe that this research will offer valuable insights for the ongoing development and enhancement of recommendation systems, providing users with more personalized and precise geolocation recommendation services.

Acknowledgement. This research is supported by the National Natural Science Foundation of China (No. 61962029, No. 62062045, No. 62262033), the Jiangxi Provincial Natural Science Foundation of China (No. 20202BAB212006) and the Science and Technology Research Project of Jiangxi Education Department (No. GJJ201832).

References

1. Tran, D.T., Huh, J.H.: Building a model to exploit association rules and analyze purchasing behavior based on rough set theory. *J. Supercomput.* **78**(8), 11051–11091 (2022)
2. Gupta, M., Thakkar, A., Gupta, V., Rathore, D.P.S.: Movie recommender system using collaborative filtering. In *2020 international conference on electronics and sustainable communication systems (ICESC)*, pp. 415–420. IEEE (2020)
3. Zhang, C., Mei, Z., Wu, B., Zhao, Z., Yu, J., Wang, Q.: Query with assumptions for probabilistic relational databases. *Tehnički vjesnik* **27**(3), 923–932 (2020)
4. Zhang, C., Mei, M., Mei, Z., Zhang, J., Deng, A., Lu, C.: PLDANet: reasonable combination of PCA and LDA convolutional networks. *Int. J. Comput. Commun. Control* **17**(2), 1–13 (2022)
5. Wu, B., Chen, X., Wu, Z., Zhao, Z., Mei, Z., Zhang, C.: Privacy-guarding optimal route finding with support for semantic search on encrypted graph in cloud computing scenario. *Wirel. Commun. Mob. Comput.* **2021**, 1–12 (2021)
6. Guo, X., Yin, S.C., Zhang, Y.W., Li, W., He, Q.: Cold start recommendation based on attribute-fused singular value decomposition. *IEEE Access* **7**, 11349–11359 (2019)
7. Papadakis, H., Papagrigoriou, A., Kosmas, E., Panagiotakis, C., Markaki, S., Fragopoulou, P.: Content-based recommender systems taxonomy. *Found. Comput. Dec. Sci.* **48**(2), 211–241 (2023)
8. Afoudi, Y., et al.: Impact of feature selection on content-based recommendation system. In: *Proceedings of the 2019 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS)*, pp. 1–6 (2019)
9. Yadalam, T.V., Gowda, V.M., Kumar, V.S., Girish, D., Namratha, M.: Career recommendation systems using content based filtering. In *2020 5th International Conference on Communication and Electronics Systems (ICCES)*, pp. 660–665. IEEE (2020)
10. Javed, U., Shaikat, K., Hameed, I.A., Iqbal, F., Alam, T.M., Luo, S.: A review of content-based and context-based recommendation systems. *Int. J. Emerging Technol. Learn. (iJET)* **16**(3), 274–306 (2021)
11. Fu, M., Qu, H., Yi, Z., Lu, L., Liu, Y.: A novel deep learning-based collaborative filtering model for recommendation system. *IEEE Trans. Cybernet.* **49**(3), 1084–1096 (2018)
12. Afsar, M.M., Crump, T., Far, B.: Reinforcement learning based recommender systems: a survey. *ACM Comput. Surv.* **55**(7), 1–38 (2022)
13. Shokeen, J., Rana, C., Rani, P.: A trust-based approach to extract social relationships for recommendation. In: Khanna, A., Gupta, D., Pólkowski, Z., Bhattacharyya, S., Castillo, O. (eds.) *Data Analytics and Management. LNDECT*, vol. 54, pp. 51–58. Springer, Singapore (2021). https://doi.org/10.1007/978-981-15-8335-3_6
14. Shokeen, J., Rana, C.: A trust and semantic based approach for social recommendation. *J. Ambient. Intell. Humaniz. Comput.* **12**(11), 10289–10303 (2021)
15. Shokeen, J., Rana, C.: Social recommender systems: techniques, domains, metrics, datasets and future scope. *J. Intell. Inform. Syst.* **54**(3), 633–667 (2020)
16. Walek, B., Fajmon, P.: A hybrid recommender system for an online store using a fuzzy expert system. *Expert Syst. Appl.* **212**, 118565 (2023)
17. Chen, Z., Zhu, S., Niu, Q., Zuo, T.: Knowledge discovery and recommendation with linear mixed model. *IEEE Access* **8**, 38304–38317 (2020)
18. Walek, B., Fojtik, V.: A hybrid recommender system for recommending relevant movies using an expert system. *Expert Syst. Appl.* **158**, 113452 (2020)
19. Berkani, L., Hanifi, R., Dahmani, H.: Hybrid recommendation of articles in scientific social networks using optimization and multiview clustering. In: Hamlich, M., Bellatreche, L., Mondal, A., Ordonez, C. (eds.) *SADASC 2020. CCIS*, vol. 1207, pp. 117–132. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-45183-7_9

20. Khanal, S.S., Prasad, P.W.C., Alsadoon, A., Maag, A.: A systematic review: machine learning based recommendation systems for e-learning. *Educ. Inf. Technol.* **25**, 2635–2664 (2020)
21. Meng, D.: Collaborative filtering algorithm based on trusted similarity. In: Proceedings of the 2018 IEEE 3rd International Conference on Signal and Image Processing (ICSIP), pp. 572–576 (2018)
22. Jebamalar, B., Sudha, K., Maheswari, B., Nalini, M., Dinesh, M.G., Subramanian, R.S.: Collaborative filtering for bargaining and customer feedback in e-commerce. In: 2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS), pp. 1804–1808. IEEE (2023)
23. Chen, Z., et al.: Towards explainable conversational recommendation. In: Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, pp. 2994–3000 (2021)
24. Nair, A.M., Benny, O., George, J.: Content based scientific article recommendation system using deep learning technique. In: Suma, V., Chen, J.I.-Z., Baig, Z., Wang, H. (eds.) *Inventive Systems and Control. LNNS*, vol. 204, pp. 965–977. Springer, Singapore (2021). https://doi.org/10.1007/978-981-16-1395-1_70
25. Muhammad, M., Rosadi, D.: Comparison of user-based and item-based collaborative filtering methods in recommender system. In: *AIP Conference Proceedings*, vol. 2720, no. 1. AIP Publishing (2023)
26. Zhang, J., Lin, Z., Xiao, B., Zhang, C.: An optimized item-based collaborative filtering recommendation algorithm. In: 2009 IEEE International Conference on Network Infrastructure and Digital Content, pp. 414–418. IEEE (2009)
27. Zhang, W., Wang, Y., Chen, H., Wei, X.: An efficient personalized video recommendation algorithm based on mixed mode. In: 2019 IEEE International Conferences on Ubiquitous Computing & Communications (IUCC) and Data Science and Computational Intelligence (DSCI) and Smart Computing, Networking and Services (SmartCNS), pp. 367–373. IEEE (2019)
28. Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. *arXiv preprint arXiv:1301.7363* (2013)