Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2024 SCPE. Volume 25, Issues 4, pp. 2903–2915, DOI 10.12694/scpe.v25i4.2860

MULTIPLE CONSTRAINT HYBRID TRAVEL ROUTE RECOMMENDATION MODEL BASED ON COLLABORATIVE FILTERING RECOMMENDATION ALGORITHM

QIONG ZHANG*

Abstract. The study presents a hybrid model for recommending travel routes that takes into account multiple constraints. This model is based on a collaborative filtering recommendation algorithm and addresses the issue of disorganized travel route recommendations. To improve upon the k-mean clustering algorithm, the proposed model introduces the Dynamic Gaussian Kernel Density K-means algorithm. After the data was processed, the initial clustering center was determined and k-means clustering was performed. Subsequently, the travel route recommendation model was created by integrating various constraints. The study's proposed algorithm was compared with alternative algorithms, and the experimental results demonstrated superior performance across a range of datasets, with the minimum sum of squared errors and a running time of approximately 1.4 seconds - a noteworthy improvement. Comparative experiments were conducted on various forgetting coefficients in the model, and the forgetting coefficient with the lowest sum-of-squares of errors was selected to replace the existing one. Upon comparing the proposed research model with other models, it was found that the former had greater accuracy and recall, amounting to 98.1% and 96.8% respectively. This suggests that the proposed research model serves as a more efficient solution for travel route recommendation.

Key words: Collaborative filtering techniques; multi-constraint mixing; travel route recommendation; cold start; sum of squared errors

1. Introduction. Travel has become a common way for people to unwind as living standards have increased, and the tourist sector is growing quickly. Travel has become a common way for people to unwind as living standards have increased, and the tourist sector is growing quickly. However, people are often overwhelmed by a vast amount of information and struggle with decision-making. In response to this, recommender systems have been developed to assist with the process. Recommendation systems research has advanced significantly in recent years as a result of the big data and artificial intelligence fields' quick advances [1, 2, 3]. Alongside the burgeoning demand for travel, the corresponding Travel Recommendation System (TRS) has emerged. Previous TRS encounter issues such as lack of data and cold start, resulting in low recommendation accuracy. Therefore, this research suggests employing Collaborative Filtering (CF) recommendation algorithm to develop a novel Multi-Constraint Hybrid Travel Route Recommendation Model (MCHTRRM). The DGKDK algorithm is utilized in an innovative way to enhance the effectiveness of the conventional clustering algorithm. Following this, the MCHTRRM is developed, which capitalizes on this improved algorithm to analyses user data and take into account multiple constraints including user information. The outcome is a personalized Travel Route Recommendation (TRR) which enables users to make more informed travel arrangements. The proposed model holds significance in increasing the efficacy of TRR and offering tailored recommendations. The research is structured into four main components. Firstly, it provides a summary of CF algorithm and recommendation technology research conducted by scholars both domestically and abroad, with the results analyzed. The second component includes an analysis of the proposed model and improvements made to the algorithms followed by its creation. The third part involves validating the model's actual effect through experimental comparison. Lastly, the research summarizes the experimental results and identifies any research shortcomings while suggesting future research directions.

2. Related Works. Deep learning has led scholars to apply CF algorithms across many fields [4, 5]. Lim et al. proposed a weighted interpolation domain single-class CF algorithm that addresses the challenge of

*College of Business Administration, Zhengzhou University of Science and Technology, Zhengzhou, 450064, China (zhangqiong 2023@126.com)

predicting unobserved TF gene associations due to current limited understanding of genome-wide TF targeting profiles. This algorithm uses a neighbor regularization method to evaluate independent data. Experimental results indicated an accuracy of 37.8% for the initial 495 predicted association realizations, demonstrating the potential to enhance TF gene prediction [7]. Wu and other scholars in the field of e-commerce, propose a CF algorithm that combines restricted Bohrman and trust information. The study inputs user preferences and item ratings in a restricted Bohrmann machine model, calculates user similarity using weighted user similarity and trust information, and prediction is carried out by integrating user's historical ratings. The proposed approach has a greater prediction accuracy than other widely used algorithms, according to experimental results [8]. For the purpose of solving the information overload problem, Dun L. and other academics presented a CF algorithm based on social information and a dynamic temporal window. In order to find the closest neighbors, the study first incorporates social media data and user-submitted shares. It then dynamically modifies the time window and adds a time function to validate the method. The study's method was proven to be highly efficient and useful by the experimental results [9]. Yu and other scholars in the field of education, propose an elective course recommendation algorithm with local CF. The study created a student personalization model based on information for online learning resources. Based on the evaluation model, a similarity matrix is combined to recommend course resources. The study's proposed algorithm can more accurately recommend the knowledge students need, according to experimental data [10].

Based on the massive amount of information emerging from the development of the web, many scholars have studied recommender systems [11, 12, 13]. Liang and other scholars propose a balanced recommendation algorithm for teaching sports network based on trust relationship in response to the problems of low recommendation trust and poor recommendation resources in traditional sports network. The research first uses SVM algorithm to classify the sports recommendation video to achieve the extraction of teaching resources on sports network. Kalman filtering method is used to reduce noise and fuse similar data to achieve data preprocessing. Then the model is established to determine the relationship attributes between the data. The experimental findings demonstrated that the suggested research approach produced a resource balance of 96 with a high level of confidence in the suggested resources [14]. A content-based recommendation system was proposed by Li and other experts in the field of recommender systems. The research uses the class decision layer for image recognition, accurately identifies the commodity image features, and uses the correspondence between different commodities to make recommendations. The findings demonstrated that this method can fully synergies with commodity special discounts, and the recommendation results were highly accurate [15]. Xie N and other scholars propose a recommendation model based on personalized double matrix recommendation algorithm for customers' personalized needs. The study uses adjectives to filter image labels, which simplifies the user process and improves the model recommendation efficiency. Finally, a perceptual demand acquisition model is constructed and validated using an air purifier. The outcomes of the experiment demonstrated how quickly and accurately the research-proposed model can identify client demand, and the forecast findings were more accurate [16]. Wang and other scholars propose a gradient descent matrix decomposition collaborative model for the sparsity sensitivity and long iteration times of the traditional singular value decomposition CF recommendation algorithm. The model is based on the singular value filtering recommendation algorithm, using the mean, column mean and global mean pre-population method, the matrix is pre-populated, and then singular value decomposition is performed to reduce data sparsity. Experimental results showed that the proposed model had high recommendation accuracy and improved prediction scores [17].

In summary, numerous scholars have conducted research on recommender systems and CF algorithms. However, there are fewer studies focusing on travel information for massive user bases. Previous TRS have encountered cold-start problems, which have resulted in low recommendation efficiency. Consequently, this research proposes a MCHTRRM based on a CF recommendation algorithm, which successfully integrates the DGKDK algorithm into the model. Furthermore, this model resolves the problems posed by data scarcity and cold-start, ultimately improving recommendation efficiency.

3. TRR Model Based on CF Algorithm. The proposed TRR model is divided into two parts, the study firstly improves the traditional K-means algorithm and then builds the MCHTRRM based on several constraints such as user and travel route.

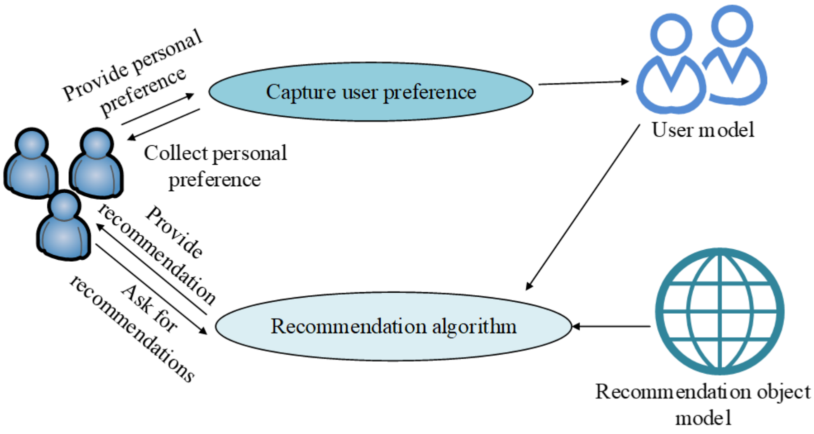


Fig. 3.1: Recommendation system model

3.1. Personalized CF Algorithm Design and Implementation. The arrival of the information age has flooded people with a huge amount of information, and users are unable to obtain truly useful information. Recommender systems emerged and developed to address this issue of information overload. After gathering data on the user's preferences, the recommender system makes educated guesses about the kind of content the user might find interesting and makes recommendations for it. A general recommender system is shown in Figure 3.1.

Figure 3.1 displays the recommender system model. The recommender system is comprised of three primary components, as seen in the figure: the recommender algorithm, the recommender object modeling component, and the user modeling component. The CF algorithm is used in the study's algorithm section. Because the CF technique requires a lot of calculations when dealing with a big number of users and products [18, 19, 20], the clustering approach can partially address this issue. Using distance as a similarity metric—closer distances signifying greater similarity—the K-means algorithm is a common clustering algorithm. The similarity between samples is calculated using the Euclidean equation, which is calculated as in Equation 3.1.

$$\operatorname{dist}(x, y)=\sqrt{(-y_1+x_1)^2+\ldots+(-y_n+x_n)^2} \tag{3.1}$$

In Equation 3.1, x, y denotes the data in n dimensions. k-means algorithm selects K centers of mass for a given K categories, after which the data nodes in the vicinity of each center of mass are clustered to obtain K clusters of categories. Then the centroid of each category is updated and the similarity between the nodes and the centroid is continued to be calculated to make the points within the cluster more concentrated and at the same time to make the distance between the clusters as large as possible. Sum of the Squared Error (SSE) is calculated as in Equation 3.2.

$$SSE=\sum_{i=1}^{k}\sum_{x\in C_i}\operatorname{dist}(\bar{x}_i, x)^2 \tag{3.2}$$

In Equation 3.2, k represents the number of clusters, C_i represents the ith category out of k clusters, and $\bar{x}_i$ represents the center of clustering of C_i. The K-means algorithm is more efficient in handling large data sets while the algorithm complexity is low. Kernel density estimation methods investigate the data characteristics of the samples, using kernel functions, which are processed and mapped in space. It is assumed that there are n data points $x_1, x_2, \ldots, x_n$ with independent distribution F. The distribution density function f obeyed by these data points is defined as in Equation 3.3.

$$\hat{f}(x)=\frac{1}{nh^2}\sum_{i=1}^{n}K\left(\frac{x-x_i}{h}, \frac{y-y_i}{h}\right) \tag{3.3}$$

In Equation 3.3, $K(x)$ denotes the kernel function which is non-negative and integrates to 1, h represents the bandwidth and $h > 0$, n denotes the number of points that can be observed within the bandwidth, $i = 1, \ldots, n$. The kernel smoothing function used in the study is the Gaussian kernel function as in Equation 3.4.

$$K(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}x^2\right) \tag{3.4}$$

The degree of similarity between samples can be calculated using the distance between data using equation 3.4, which can effectively capture the similarity between samples. The value of bandwidth h is calculated as in Equation 3.5.

$$h_{\text{optimal}} = \frac{\int K^2(t)\, dt}{n \left(\int t^2 K(t)\, dt\right)^2 \int [f''(x)]^2\, dt} \tag{3.5}$$

The bandwidth can be calculated using equation 3.5. Through the above calculation, the K-means algorithm can be implemented. However, the K-means algorithm has the disadvantages of requiring manual input of the clustering K-value as well as randomly determining the initial center, the study proposes Dynamic Gaussian Kernel Density K-means (DGKDK) algorithm based on Gaussian kernel density for dynamically determining the initial clustering center. Firstly, the distance within the class is calculated as in Equation 3.6.

$$s_i = \frac{1}{|\bar{x}_i|} \sum_{x \in C_i} \|x - \bar{x}_i\| \tag{3.6}$$

Equation 3.6 can be used to determine the mean value of the distance between each data point and the cluster's center point. The interclass distance is then calculated as in Equation 3.7.

$$d_{i,j} = \|\bar{x}_i - \bar{x}_j\| \tag{3.7}$$

Using Equation 3.7, the distance between the centroids of two clusters can be calculated. Afterwards, the average maximum similarity within class (AMS) is calculated as in Equation 3.8.

$$AMS = \frac{1}{k} \sum_{i=1}^{k} \max\left\{\frac{s_i + s_j}{d_{i,j}}\right\} = \frac{1}{k}\left[\max\left\{\frac{s_1 + s_j}{d_{1,j}}\right\} + \max\left\{\frac{s_2 + s_j}{d_{2,j}}\right\} + \ldots + \max\left\{\frac{s_k + s_j}{d_{k,j}}\right\}\right] \tag{3.8}$$

Using Equation 3.8 the AMS can be obtained by taking the maximum value of similarity between each class and other classes and taking the mean value of the maximum value. When the AMS is smaller, the clustering effect is better. As a result, the clustering effect is at its best and the ideal number of clusters is reached when AMS takes the minimal value. Figure 3.2 depicts the DGKDK algorithm's structure.

Figure 3.2 shows the structure of DGKDK algorithm, as shown in Figure 3.2, the data samples are processed using kernel density estimation to obtain the set of extremely dense points. It is calculated accordingly to get the best clustering centre and finally K-means clustering is performed.

3.2. Design and Implementation of MCHTRRM. TRS can be categorised into content-based filtering techniques, CF-based recommendation techniques and hybrid recommendation techniques depending on the algorithm used. The study uses the CF recommendation technique, which can assist in achieving target user recommendations based on other users' past behavior toward the products and might suggest things that consumers could find interesting. User-based, model-based, and item-based CF recommendations are the three further categories into which CF techniques can be separated. In order to create a "neighbourhood" user group that is similar to the target user, user-based CF recommendation first uses correlation between users. Next, it uses the user group's historical preferences to calculate the target user's predicted score for the project. Lastly, it recommends the target user based on the score level. Fig. 3 depicts the user-based CF procedure.

Figure 3.3 illustrates the user-based CF recommendation principle. According to Fig. 2, User A indicates a preference for Commodity 1 and Commodity 3, User B for Commodity 2, and User C for Commodity 1,

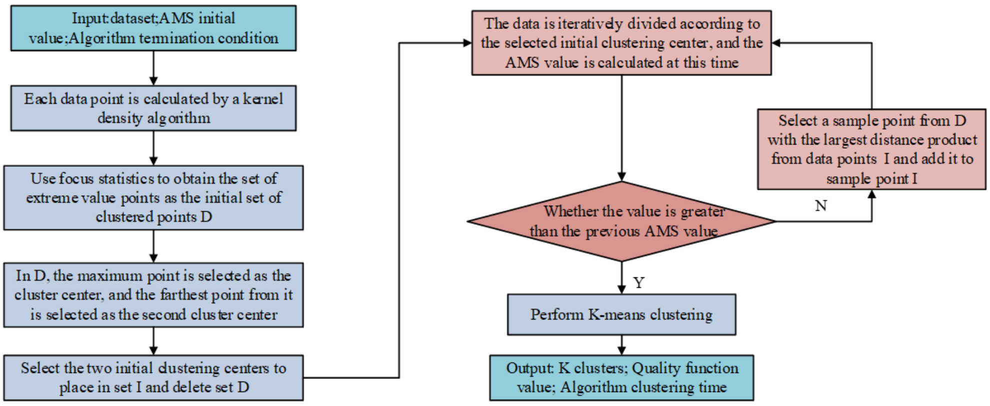


Fig. 3.2: DGKDK algorithm structure

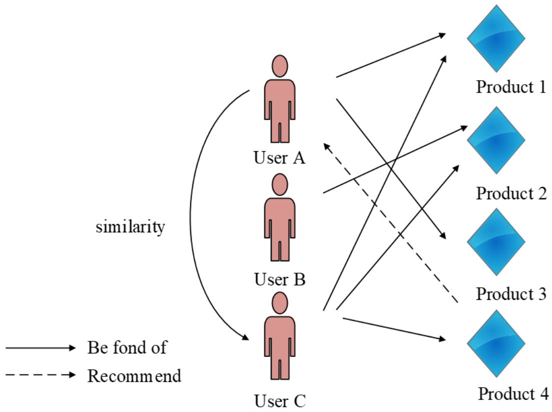


Fig. 3.3: User-based collaborative filtering recommendation schematic

Commodity 3, and Commodity 4. Drawing from the available data, it can be inferred that User A and User B do not share any preferences, however User A and User C have a shared interest in Commodities 1 and 3. Furthermore, User C shows a greater similarity to User A in terms of product preference. However, in addition to User C's liking for Commodity 1 and Commodity 3, they also prefer Commodity 4. Consequently, the system speculates that User A may also be interested in Commodity 4, and thus suggests it to User A. The user based coordinated filtering algorithm basically consists of four steps: preprocessing user data, calculating nearest neighbor set, calculating predictive scores and generating recommendation list. The raw user data is preprocessed as well as modelled using data mining and preprocessing to get the user item rating matrix. Finding the nearest neighbor set is a crucial stage in the recommendation algorithm that has an impact on the algorithm's efficiency. Three approaches are available for calculating the nearest neighbor set: Pearson's correlation coefficient, modified cosine similarity, and cosine similarity. Equation 3.9 is used to determine the cosine similarity, or Cos-S.

$$\mathrm{sim}(\mathbf{u},\mathbf{v}) = \cos(\mathbf{u},\mathbf{v}) = \frac{\mathbf{u}\cdot\mathbf{v}}{\|\mathbf{u}\|\cdot\|\mathbf{v}\|} \tag{3.9}$$

In Equation 3.9, vector u represents the ratings of user u in n-dimensional space, and vector v represents the ratings of user v in n-dimensional space, and the similarity between the two users can be calculated by using

Equation 3.9. The modified Cos-S calculation is shown in Equation 3.10.

$$\operatorname{sim}(\mathbf{u},\mathbf{v})=\frac{\sum_{i\in I_{uv}}(-\bar{R}_u+R_{ui})(-\bar{R}_v+R_{vi})}{\sqrt{\sum_{i\in I_u}(-\bar{R}_u+R_{ui})^2\sum_{i\in I_v}(-\bar{R}_v+R_{vi})^2}} \tag{3.10}$$

In Equation 3.10, I_u and I_v represent the aggregated ratings of users u and v , respectively;R_{ui} and R_{vi} represent the ratings of users u and v , respectively, on item $\bar{R}_u$. And $\bar{R}_v$ represents the aggregated ratings of users u and v on the same item. $\bar{R}_u$ and $\bar{R}_v$ represent the mean values of the ratings of users u and v , respectively, on the aggregated I_{uv} on the same item. The corrected Cos-S corrects for inertia in user ratings. Equation 3.11 can be used to get the Pearson correlation coefficient.

$$\operatorname{sim}(u,v)=\frac{\sum_{i\in I_{uv}}(-\bar{R}_u+R_{ui})(-\bar{R}_v+R_{vi})}{\sqrt{\sum_{i\in I_{uv}}(-\bar{R}_u+R_{ui})^2\sum_{i\in I_{uv}}(-\bar{R}_v+R_{vi})^2}} \tag{3.11}$$

Equation 3.11 can be used to calculate how similar users are to one another on various items. Based on the similarity between the target user and the nearest neighbour set, the unselected items of the target user can be predicted after the nearest neighbour set has been obtained. The list of ratings is predicted, sorted and the top N items are recommended. The study proposes a MCHTRRM based on CF recommendation algorithms and content-based recommendation algorithms. The user preference part of the recommendation model is the underlying content in which the user's preference changes over time. For example, users will decrease their interest in a particular type of attraction as their browsing and searching behaviour for that type of attraction increases. The study uses a non-forgetting curve to represent the change in user's interest as in Equation 3.12.

$$h(t_{i_c})=(1-\theta)+\theta\left(\frac{t_{i_c}-t_{i_e}}{t_{i_l}-t_{i_e}}\right)^2 \tag{3.12}$$

In Equation 3.12, $t_{i_e}<t<t_{i_l}$, $0\leq\theta\leq 1$, $0<h(t_{i_c})<1$. t_{ic} represents the time of user's interaction behaviour for attractions of category i. t_{ie} represents the earliest time of user's interaction behaviour for attractions of category i in the history record. t_{il} represents the latest time of user's interaction behaviour for attractions of category i in the history record. θ represents the coefficient of forgetfulness, the larger θ is, the faster the user forgets, and conversely, the slower the user forgets. D represents the coefficient of forgetting. The study incorporates the DGKDK algorithm into CF recommendation, improves it, and proposes a hybrid attraction constraint algorithm. The hybrid constraint attraction recommendation algorithm is used to derive a set S of tourist attraction recommendations, after which a tourist itinerary is developed based on the recommendation results and other constraints, as shown in Figure 3.4.

Figure 3.4 shows the path network graph as shown in Fig. The graph contains the user's starting point Su, the attraction's ticket price SC, and the estimated playing time ST. The edge set of the paths is defined as the set $R=\{r_1,r_2,\ldots,r_{|R|}\}$, and the weights on each path are denoted as Equation 3.13.

$$TT(s_i,s_j)=\frac{\operatorname{distance}(s_i,s_j)}{V_{\text{average}}} \tag{3.13}$$

In Equation 3.13, s_i,s_j denotes two attractions, $TT(s_i,s_j)$ denotes the time spent travelling between the two attractions, *distance* denotes the distance, and $v_{average}$ denotes an average speed. The time to reach the next attraction can be defined as Equation 3.14.

$$AT(s_{k+1})=AT(s_k)+ST(s_k)+TT(s_k,s_{k+1}) \tag{3.14}$$

In Equation 3.14, $ST(s_k)$ denotes the time spent travelling to attraction s_k and $AT(s_k)$ denotes the time to reach the previous attraction s_k. The travelling time is expressed as Equation 3.15.

$$TPT(s_u,tp)=TT(s_u,s_k)+TT(s_{k+n},s_u)+\sum_{j=1}^{n-1}TT(s_{k+j},s_{k+j+1})+\sum_{i=1}^{n}ST(s_{k+i}) \tag{3.15}$$

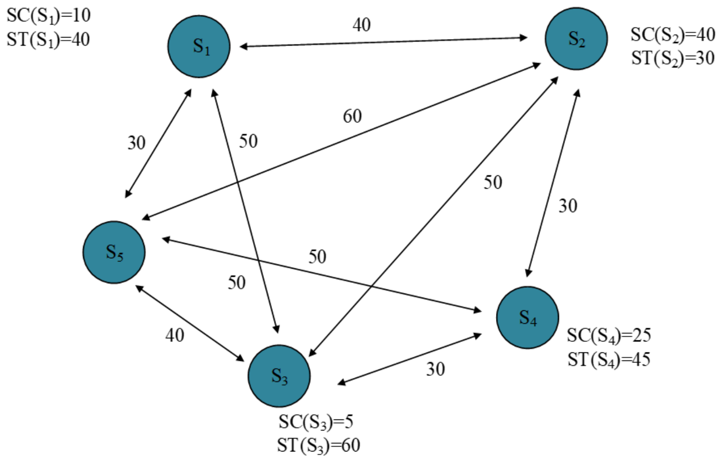


Fig. 3.4: Path network diagram

Using Equation 3.15 the travel time for a segment of attractions with a travel path of $tp = \langle s_k, s_{k+1}, \ldots, s_{k+n} \rangle$ and a start point of S_u can be calculated to represent the time spent by the user on this journey. The cost in travelling is calculated as Equation 3.16.

$$TPC(tp) = \sum_{i=1}^{n} SC(s_i) \tag{3.16}$$

Using Equation (16) the total economic cost spent can be calculated when the travel path is $t_p = \langle s_k, s_{k+1}, \ldots, s_{k+n} \rangle$. The MCHTRRM framework is shown in Figure 3.5.

Figure 3.5 shows the structure of MCHTRRM. As shown in the Figure the model enters the user data, preprocesses the data, and based on the attraction scores derived from the algorithm, recommends the user and performs route planning.

4. MCHTRRM Performance Study. The study tests the model to ensure that the TRR time criteria are satisfied before confirming the validity of the suggested algorithms in the model. This confirms the performance of the suggested model.

4.1. Performance Study of DGKDK Algorithm. By using four datasets from the University of California, Irvine Machine Learning Repository (UCI) dataset for testing, the study demonstrates the effectiveness of the proposed DGKDK method. The four datasets examined with various datasets to better validate the efficacy of the suggested enhanced algorithm are the Iris dataset, Wine dataset, Glass dataset, and Yeast dataset. Table 4.1 displays the setting of the laboratory environment used for the dataset validation.

Table 4.1 shows the laboratory hardware and software environments, and Java was chosen as the development language for the operating system. The study compares the K-means algorithm, Diversity-Aware Crossover Clustering Based on K-means (DACC-KM) algorithm, Intergrated Clustering and Classification System Based on K- means (ICCS-KM) algorithm is compared with the proposed DGKDK algorithm of the study and the results are shown in Figure 4.1.

Figure 4.1 shows the accuracy and SSE comparison of each clustering algorithm. The ICCS-KM algorithm combines similar clustering methods with the K-means algorithm to improve the accuracy and efficiency of data analysis. DACC-KM algorithm uses K-means algorithm for dynamic attribute and attribute clustering to improve the clustering effect. As can be seen in Figure 4.1a, the DGKDK algorithm is mostly more accurate than the other algorithms, and is slightly lower than the DACC-KM algorithm on the Iris dataset, but also

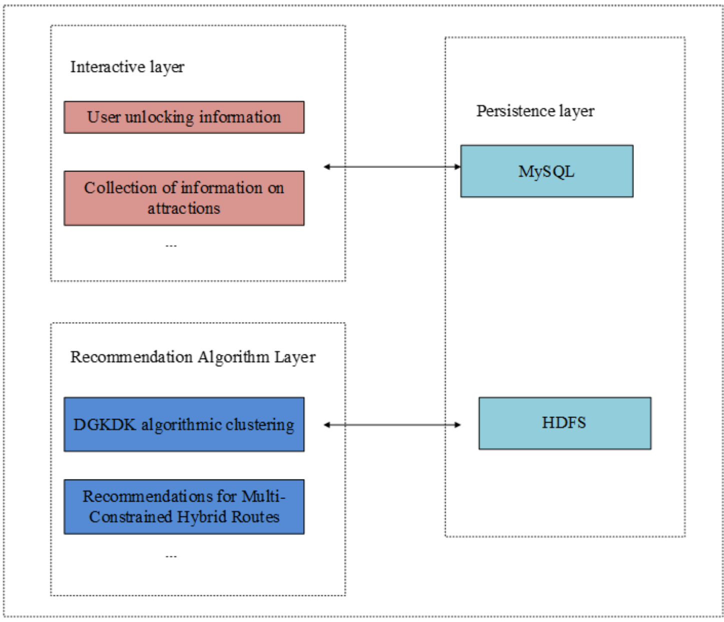


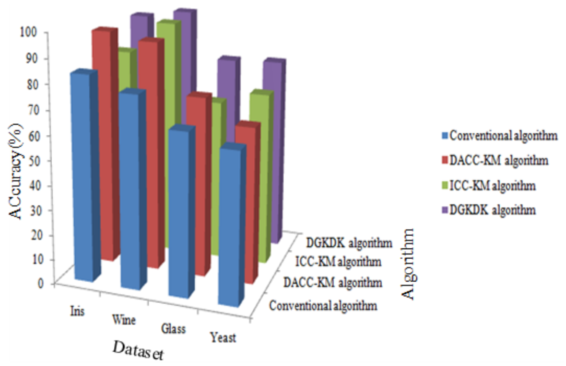
Fig. 3.5: Multi-constraint hybrid travel route recommendation model structure

Table 4.1: Laboratory environment setting

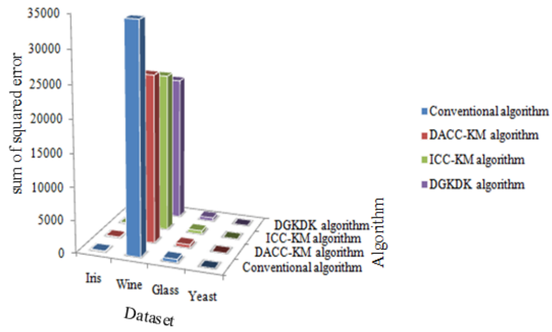
Hardware and software configuration	Version model
CPU	Intel(R)Core i7-7700@3.6GHz
GPU	GTX 1060
Operating system	Microsoft Window10
RAM	32G
Display memory	6G
CUDA	9.1
Deep learning frameworks	Pytorch1.10
Python version	3.7

maintains a high accuracy of 94.67%. At 79.33%, the DGKDK algorithm outperforms the other three clustering algorithms on the Yeast dataset. The DGKDK algorithm performs marginally better than the other algorithms on the Wine and Glass datasets, scoring 97.73% and 78.54%, respectively. In Figure 4.1b, the DGKDK algorithm has the smallest SSE and the smallest SSE on the four datasets, indicating that the DGKDK algorithm fits the data better. The running time of the experiments on the datasets using the four algorithms is shown in Figure 4.2.

On the Iris dataset, which has a small amount of data, the traditional K-means algorithm and the DACC-KM algorithm have an advantage with a shorter running time. However, on the Yeast dataset, which has a larger total amount of data, the DGKDK dataset proposed in the study has a significant advantage, with a running time of around 1.4s. The other algorithms, on the Yeast dataset, have a slower running time of around 2s. The study applies the proposed DGKDK algorithm to CF recommender system and evaluates the performance of the algorithm using actual user ratings compared to predicted ratings. The study compares the proposed algorithm with the traditional User CF algorithm, M-User CF algorithm, and the clustered CF algorithm with improved similarity. Additionally, Figure 4.3 displays the algorithm's running time as well as



(a) Comparison of Accuracy of Each Algorithms



(b) Comparison of Error of Sum of Square Across Clustering Algorithms

Fig. 4.1: Comparison of accuracy and error sum of squares of each clustering algorithm

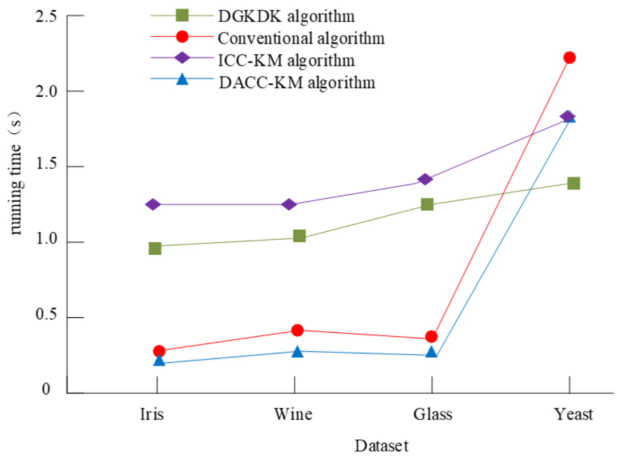
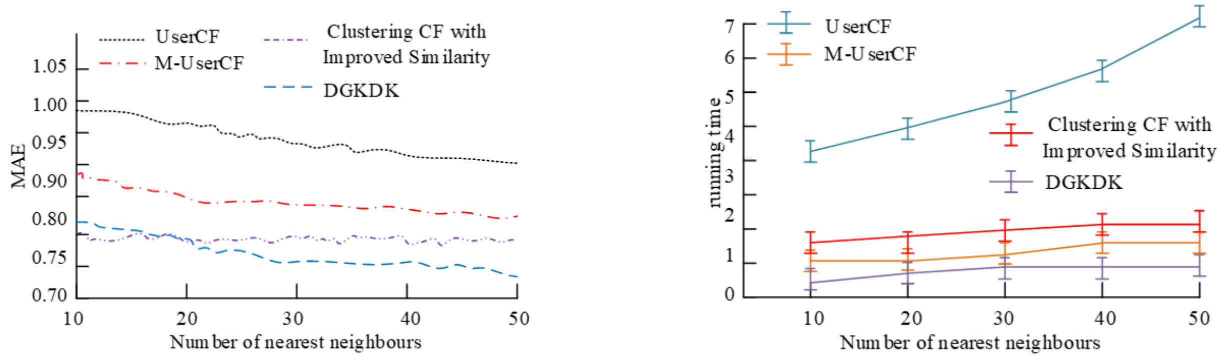


Fig. 4.2: Clustering time line plots of four algorithms on different data sets

the Mean Absolute Error (MAE) results.

Figure 4.3 shows the MAE and algorithm runtime for the four algorithms with different number of neighbors. The traditional User CF algorithm is a user-based CF algorithm that analyses the similarity of users to predict user preferences. The M-User CF algorithm solves the problem of high sparsity of the algorithmic matrix and comprehensively exploits the similarity data of users. The clustering CF algorithm with improved similarity clusters users or items and combines the clustering information to achieve prediction and recommendation. The MAE values of the various algorithms decrease as the number of nearest neighbors rises from 10 to 50 in Fig 4.3a suggesting that the recommendation algorithms' accuracy is increasing steadily and that the predicted ratings are becoming closer to the user's actual ratings. The MAE value of the study's recommended approach, which is 0.75 when the number of nearest neighbors is 50, is significantly less than the values of the other techniques. This implies that compared to other recommendation systems, the algorithm is more accurate. The study's suggested technique in Fig 4.3b runs substantially faster than the other algorithms for varying numbers of nearest neighbors, which are maintained inside a 1 s window. The M-User CF algorithm and the clustered CF algorithm with improved similarity increase from about 1s to about 1.5s with the increase in the number of



(a) Comparison of MAE values of Four Algorithms (b) Comparison of Running Time of Four Algorithms

Fig. 4.3: The MAE and the running time of the algorithm under different neighbor numbers of the four algorithms

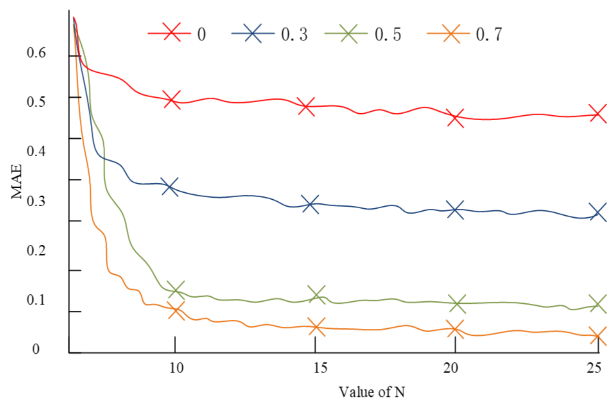


Fig. 4.4: Comparison of MAE under different forgetting coefficients

nearest neighbors, while the traditional User CF algorithm has a significant increase in running time with the number of nearest neighbors.

4.2. TRR Model Performance Study. The study analyses the performance of MCHTRRM based on CF recommendation algorithm, which randomly grabs 1480 users' basic information as well as 2389 attraction ratings data as a dataset on a travel website. There exists a forgetting coefficient θ in the user preference part of the model, and different values of N in the TopN of the recommendation list are selected, and the experimental results are shown in Figure 4.4.

Figure 4.4 shows the comparison of MAE under different forgetting coefficients. The models under different forgetting coefficients have different mean absolute errors, i.e., there is a significant difference in the accuracy of the models under different forgetting coefficients. When the forgetting coefficient θ takes the value of 0, the overall MAE value of the model is the highest, i.e., the model accuracy is the worst. The model with the highest accuracy and the lowest MAE value at various values of N is the one with the value of forgetting coefficient θ set to 0.7. The accuracy of the model grows and its MAE value lowers as N increases. Using the model with a forgetting coefficient of 0.7 as an example, the model MAE value changes more steadily when N increases from

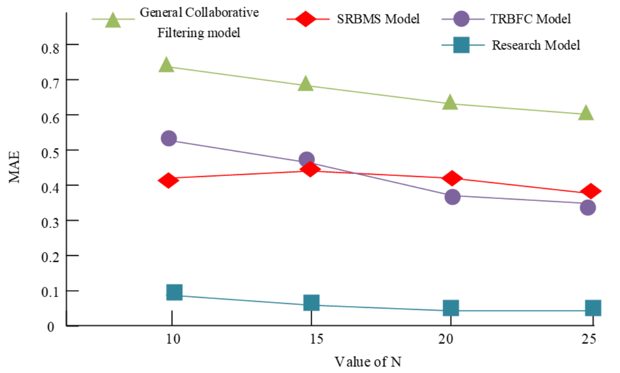


Fig. 4.5: MAE values of each model with different N values

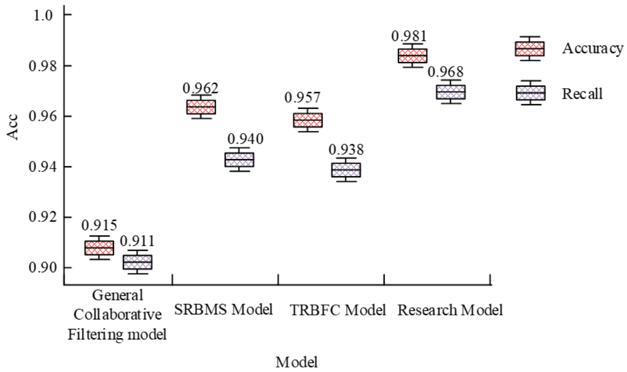


Fig. 4.6: Comparison of accuracy and recall rate of four models

20 to 25 and declines from 0.9 to 0.5 when N increases from 10 to 20. Therefore, the forgetting coefficient in the model is chosen to be 0.7. To validate the model performance, the proposed model of the study is compared with the general CF model, Sequential Rule-Based Model Selection (SRBMS) model and Takagi-Sugeno Type Radial Basis Function Fuzzy Classifier (TRBFC) model for comparison. In Fig. 10, the experimental outcomes are also displayed.

The MAE values of each model for various N values are displayed in Figure 4.5 As can be observed from the Fig., the research proposed model has the lowest overall MAE value, which suggests that the model fit is good and the prediction results have fewer error with the user's genuine value. Whereas the general CF model has the highest MAE value, indicating that the model predicts results with larger error from the user's true value and poorer model fit. The SRBMS model is optimized iteratively to improve the classification performance. The very flexible radial basis function and fuzzy clustering form the foundation of the TRBFC model. The MAE values of the general CF model, SRBMS model and the model proposed by the study are significantly affected by the N value, while the MAE values of the TRBFC model are not changed much by the N value. In the case of the model proposed by the study, for example, the model MAE value decreases from about 0.9 to about 0.5 when the N value increases from 10 to 15, while the model MAE value stabilizes when the N value increases from 15 to 25. According to the dataset, the accuracy and recall of the four models are calculated in Fig 4.6.

The accuracy and recall of the four models are compared in Fig. 11. The research-proposed model has good performance, as evidenced by the Fig., which shows that it has high accuracy and recall, reaching 0.981

and 0.968, respectively. Whereas, the semi-CF model, SRBMS model, and TRBFC model have lower accuracy and recall metrics than the research-proposed model and have average performance. Therefore, the research-proposed MCHTRRM based on CF recommendation algorithm can achieve the recommendation purpose better and meet the experimental requirements.

5. Conclusion. The study develops MCHTRRM, utilizing a CF recommendation algorithm to offer travel guidance relevant to users' information. The study improves personalized TRS from two angles to address data sparsity and cold-start issues. The kernel density function is used to find alternate centers of clustering in the DGKDK algorithm, which is presented. The K-means clustering computation is then carried out after the first centers of clustering are identified. The study introduces the MCHTRRM CF recommendation algorithm, which is informed by user information, attraction information, and interactive use information. The study conducts comparative experiments between the DGKDK algorithm proposed in the study and the traditional K-means algorithm, DACC-KM algorithm, and ICCS-KM algorithm on four datasets, specifically Iris, Wine, Glass, and Yeast. The study's experimental results demonstrate that the proposed DGKDK algorithm exhibits high accuracy in comparison to other algorithms, achieving 97.73% and 94.67% on Wine and Iris datasets, respectively. The algorithm proposed in the research also records the smallest SSE on all four datasets. Additionally, the DGKDK algorithm demonstrates prompt processing, with a running time of approximately 1.4 seconds. This indicates that the DGKDK algorithm is a good fit for the data and can process it rapidly. Based on the comparison of the effect of different forgetting coefficients on the model, the model forgetting coefficient was chosen to be 0.7, which had the lowest MAE value in the range of 0.1. Comparing the research proposed model with the general CF model, SRBMS model, and TRBFC model, the research proposed model had the lowest MAE value at different values of N. The model had the highest accuracy and recall, 98.1% and 96.8% respectively, indicating the superior performance of the research proposed model. The study suggests that the model has some limitations and that time is calculated in a more consistent manner. Further studies could enhance the model by accounting for floating tour times.

Funding. The research is supported by: Henan Social Science Planning Special Project, Research on the Implementation of the Integration Strategy of Culture, Tourism, Culture and Creativity in Henan Province, (No: 2022ZT29); Soft Science Project of Henan Provincial Department of Science and Technology, Research on the High Quality Development Path of Rural Health Tourism in Henan Province under the Background of Population Aging (NO.242400410298).

REFERENCES

[1] Liu, Z., Han, J., Meng, F. & And, L. and web-based group decision support system in multilingual environment with hesitant fuzzy linguistic preference relations. *International Journal Of Intelligent Systems.* **8**, 5186-5216 (2021)

[2] IoT, W. for smart English education: AI-based personalized learning resource recommendation algorithm. *Int.* **3**, 200-207 (2023)

[3] Chen, C., Zhang, S., Yu, Q., Ye, Z. & Hu, F. Personalized travel route recommendation algorithm based on improved genetic algorithm. *Journal Of Intelligent & Fuzzy Systems: Applications In Engineering And Technology.* **3**, 4407-4423 (2021)

[4] Pang, S., Yu, S., Li, G., Qiao, S. & Wang, M. Time-Sensitive Collaborative Filtering Algorithm with Feature Stability. *Computing And Informatics.* **1**, 141-155 (2020)

[5] Na, L., Ying, L., Jun, T., Xia, L. & Wang, C. Improved user-based collaborative filtering algorithm with topic model and time tag. *International Journal Of Computational Science And Engineering.* **2** pp. 22 (2020)

[6] Xu, C., Wang, J. & Yuan, J. Collaborative filtering algorithm based on multi-factors. *International Journal Of Computing Science And Mathematics.* **1**, 29-39 (2020)

[7] Lim, H. & Xie, L. New Weighted Imputed Neighborhood-Regularized Tri-Factorization One-Class Collaborative Filtering Algorithm: Application to Target Gene Prediction of Transcription Factors. *IEEE/ACM Transactions On Computational Biology And Bioinformatics.* **18**, 126-137 (2021)

[8] Wu, X., Yuan, X., Duan, C. & Wu, J. novel collaborative filtering algorithm of machine learning by integrating restricted Boltzmann machine and trust information. *Neural Computing And Applications.* **9**, 4685-4692 (2019)

[9] Dun, L., Cui, W., Lun, L. & Zhiyan, Z. Collaborative filtering algorithm with social information and dynamic time windows. *Applied Intelligence: The International Journal Of Artificial Intelligence, Neural Networks, And Complex Problem-Solving Technologies.* **52**, 5261-5272 (2022)

[10] Yu, J., Xiong, Z., Bao, Q. & Ning, X. Design of an algorithm for recommending elective courses based on collaborative filtering. *Journal Of Computational Methods In Sciences And Engineering.* **6**, 2173-2184 (2022)

[11] Yang, Z., Xia, D., Liu, J., Zheng, C., Qu, Y., Chen, Y. & Zhang, C. Fusion of Internal Similarity to Improve the Accuracy of Recommendation Algorithm. *Journal On Internet Of Things.* **3**, 65-76 (2021)
[12] Gao, Y., Liang, H. & Sun, B. Dynamic network intelligent hybrid recommendation algorithm and its application in online shopping platform. *Journal Of Intelligent & Fuzzy Systems: Applications In Engineering And Technology.* **5**, 9173-9185 (2021)
[13] Zhang, H., Jian, Y. & Zhou, P. Collaborative Filtering Recommendation Algorithm Based on Class Correlation Distance. *Recent Advances In Computer Science And Communications.* **3**, 887-894 (2021)
[14] Liang, X. & Yin, J. Recommendation Algorithm for Equilibrium of Teaching Resources in Physical Education Network Based on Trust Relationship. *Journal Of Internet Technology.* **1**, 133-141 (2022)
[15] Li, B., Li, J. & Ou, X. Hybrid recommendation algorithm of cross-border e-commerce items based on artificial intelligence and Multiview collaborative fusion. *Neural Computing & Applications.* **9**, 6753-6762 (2022)
[16] Xie, N., Chen, D., Fan, Y. & Zhu, M. The acquisition method of the user's Kansei needs based on double matrix recommendation algorithm. *Journal Of Intelligent & Fuzzy Systems: Applications In Engineering And Technology.* **2**, 3809-3820 (2021)
[17] Wang, X., Wang, C., Chen, J., Liao, Y., Descent, H. & Pre-For Svd Recommendation ALGORITHM. Journal of nonlinear and convex analysis. (2021)
[18] Guo, Y., Mustafaoglu, Z. & Koundal, D. Spam Detection Using Bidirectional Transformers and Machine Learning Classifier Algorithms. *Journal Of Computational And Cognitive Engineering.* **2**, 5-9 (2023)
[19] Chen, Y., Zheng, G., Zhang, Y., Wang, C., Su, F. & Wen, S. User-based collaborative filtering algorithm fusing the local and global nearest neighbor. *International Journal Of Internet Manufacturing And Services.* **5**, 260-278 (2019)
[20] Xiaohui, C., Li, F. & Qiong, G. Collaborative Filtering Algorithm based on Data Mixing and Filtering. *International Journal Of Performability Engineering.* **8**, 2267-2276 (2019)

Edited by: Zhengyi Chai
Special issue on: Data-Driven Optimization Algorithms for Sustainable and Smart City
Received: Nov 16, 2023
Accepted: Dec 28, 2023