A New Online Education Personalized Recommendation Algorithm

DOI : 10.36909/jer.ICCSCT.19485

Zhaojun Pang[1], Wenbin Wu[2], Xinxin Fan[3], Zhixin Liu[4*]

[1]School of Education, Xi'an Fanyi University, Xi'an 710105, China.
[2]Xi'an University, Xi'an, 710065, China.
[3]University of Electronic Science and Technology of China, Chengdu, 610054, China.
[4]School of Life Science, Shaoxing University, Shaoxing 312000, China.

* Corresponding Author: liuzhixin@usx.edu.cn (ZL)

ABSTRACT

For online education platforms, a personalized recommendation system is crucial, and the collaborative filtering algorithm is the primary recommendation algorithm used. This study took the recommendation of crowdfunding platforms as a sample, and enhanced the collaborative filtering algorithm based on the user score and project attribute features of the crowdfunding platform, intending to resolve the cold start issue brought on by the platform's reliance on a single data source. The study concludes with experimental proof of the paper's suggested better method. This approach can alleviate the cold start issue to some degree. The prediction accuracy has been much enhanced in comparison with the conventionally advised method. The method can also adapt to user tastes over time, learning what they like and what they don't. It also has an excellent real-time suggestion impact. The performance verification of the algorithm in this research is also conducted using data from a live crowdfunding site, lending credence to the study's claim of greater practicality.

Keywords: Online Education; Recommendation algorithm; User score; Item attribute.

INTRODUCTION

With the development of online education, personalized selection of educational content has become the common pursuit of educators and educatees, and personalized recommendation for users has become the goal of research. The personalized recommendation system examines the user's routine, personality, and preferences; the user's changing requirements are monitored in real-time; the system then automatically adjusts the structure and subject matter of the user's information services; and finally, the system pushes the relevant products and data to the user. The first recommendation system was developed by Goldberg et al. in 1992, which was called “tapestry” (Goldberg et al., 1992), e-mail filtering was finished by the system by comparing the user's preferences to those of other users, and it was the first time the concept of "collaborative filtering" has been presented.

Even though several new recommendation technologies and algorithms have been presented one after the other, it also has been a hot issue for scholars in both the United States and other countries. Studies on the crowdfunding platform's individualized recommendation system, which is powered by machine learning (Sarma et al., 2021), have substantial repercussions on the growth and expansion of the platform as a whole. However, the algorithms that are currently used to provide recommendations were derived from more conventional machine learning approaches. That is, a prediction model is developed after problems are modeled, original data are trained using the most efficient method in light of the relevant strategy analysis, and then the model is used to make predictions.

Because the recommendation algorithm employed defines the quality of the push, the usefulness of a customized recommendation system is strongly dependent on the suitability of the recommendation algorithm that is utilized. Collaborative filtering (CF) methods generate recommendations based on usage patterns without requiring exogenous information about items or users (Koren et al., 2022). Although proposed earlier, it has shown advanced accuracy so far, therefore, it is the approach that has the most users and the most experience. At present, the recommendation system still has problems such as cold start, sparse assessment data, the difficulty of cross-category recommendation, recommendation accuracy, etc. (Su et al., 2009). Common strategies for addressing problems in customized recommendation systems include Bayesian estimation, the nearest neighbor method, neural networks, support vector machines, decision trees, and others (Khanal et al., 2020).

Improving the system's real-time efficiency while also giving users the chance to broaden their interests is one of the obstacles that must be overcome in collaborative filtering recommendations. Clustering (Merialdo. A.K.B., 1999), Bayesian networks (Breese et al., 1998), dimensionality reduction, etc., and other similar approaches are some of the most widely used techniques in practice today. The search field may be greatly narrowed by using clustering-based recommendation, which first categorizes persons so that targeted users can search for neighbors. This can significantly enhance the real-time performance of suggestions and improve their overall quality. The accuracy of the suggestions may suffer as a result of using this method, which is a clear negative of the strategy. Users are given access to a prediction model as a result of an implementation technique that is based on Bayesian networks. This model has outstanding recommendation influence and is delivered to users. However, the process of putting up each model is laborious and inefficient, which makes it challenging to ensure the system's real-time performance. In addition, each update will need a substantial amount of storage space to store the data it generates.

Currently, the use of probabilistic models in collaborative filtering recommendation algorithms has been met with some degree of success by researchers. Hofmann and colleagues (Hofmann. T., 2004) proposed a method known as probabilistic latent semantic analysis. This software employs statistical modeling methods to incorporate the characteristics that may be used for classification in a hybrid model into a model of user associations and prototype interests. These factors may be included in the model thanks to the program's usage of statistical modeling techniques. This technique offers more accuracy than the conventional approach to collaborative filtering. Shani et al. apply a Markov decision process model to the problem of suggestion generation as a sequential optimization problem (Shani et al., 2012). This makes the task a sequential optimization problem. Because it takes into account both the short-term and the long-term implications of each proposal, it performs far better than earlier models. In addition to these features, the technology behind probability graph models also includes a user rating profile, semantic generation model, and document topic generation model.

In addition, recommendation systems that are Content-based, knowledge-based, utility-based, based on association rules, and based on user statistics are also employed extensively. Content-based recommendation algorithms extract features according to the user's historical preference and then select the top item to recommend to the user according to the similarity ranking among all the items to be recommended. However, because some special feature information is difficult to extract, the recommendation algorithm will be constrained. Knowledge-based recommendation algorithms take into account a user's profile as well as their previous interactions to provide recommendations that are tailored to the user's specific preferences and areas of interest. But it depends a lot on the initial data. The utility-based recommendation ties up any loose ends left over from the procedure by customizing recommendations according to the user's determined utility function for each item. It is possible to broaden the scope of the utility function so that it takes into consideration aspects other than product attributes. However, pinpointing the precise nature of the utility function might be difficult. It is possible to create recommendations based on association rules by first conceptualizing each user as a transaction and their viewed or bought items as an item set. Apriori and FP growth types are both common examples of popular algorithms (Margahny et al., 1994). It is necessary for the operation of the recommendation algorithm (Huang. M., 2019) that it be able to classify users according to the demographic data that they provide, such as their age, employment, location, hobbies, and so on. It requires more detailed user information than a record of the user's previous behaviors.

According to the findings of the aforementioned research, it is abundantly clear that traditional recommendation systems place an excessive amount of weight on a single quality of input. As a result, it is challenging for the produced suggestions to meet the requirements of high accuracy in certain settings. Some researchers mine the hidden qualities of the data to develop a more accurate recommendation. Ziegler et al. developed the strategy of topic diversification as a method to guarantee that their individualized suggestions appeal to the vast variety of interests held by their consumers (Ziegler et al., 2005). Liu and Weng (Weng et al., 2004) analyze consumers' preferences for product-specific functionalities based on transaction data and product features and then construct customer preference criteria to promote potentially appealing items to customers. This allows Liu and Weng (Weng et al., 2004) to promote products that customers may find appealing. A technique for collaborative filtering that is based on weighted clustering was proposed by George et al. to simultaneously group users and items (George et al., 2005). Sobecki presented a mixed recommendation system by using project content attributes, user ratings, and user data (Sobecki. J., 2006).

This work attempts to solve the problem of the cold start that occurs in the collaborative filtering process. The collaborative filtering approach is made more effective in this research by including the user score and the item attribute in the formulation of the final prediction score.

METHODS AND ALGORITHM FLOW

2.1 Algorithm improvement idea

The user will not immediately begin any kind of assessment or support activities at the first stage of the launch, upon the registration of a new user, or the generation of a new project. This is what is meant by the term "cold start," and it refers to the fact that the user will not immediately begin using the platform. There is currently no score recorded for his project on the scorecard. Because score data is the bedrock upon which collaborative filtering suggestions are built, the system is unable to perform the service that has been requested.
The algorithm's cold start problem may easily be remedied for new users with the use of the hot push technique. This entails disseminating the most popular projects that have seen significant growth in their support quotas in a short amount of time. Additionally, it is possible to randomly offer some of the items and then fill in the remainder depending on the user's evaluations of the already proposed things.

The text of the project is studied, and the recommendation is completed based on its proximity to the text information of the user preference project. This is how the "cold start" problem of a new project is addressed and resolved. To make things even worse, it takes a significant amount of time, effort, and energy to install, and it consumes a significant amount of the resources that are available on your computer. A reward system, in which users get something of value in return for evaluating new projects, is an additional alternative that may be implemented. However, it may be difficult to persuade consumers to switch.

In conclusion, the problem of new projects is more difficult to tackle than the problem of new users having a difficult beginning in their experience. It has been determined that one particular piece of input data is the root cause of the problem. Just the score data is taken into consideration when making suggestions. When there is insufficient data on the scores, it is difficult to complete the advice. The crowdfunding website does a good job of grouping projects that are related together. As a result, the recommendation system might benefit from the addition of yet another attribute feature, namely information on item categorization.

Therefore, the new algorithm for this work is based on the following principles: in the first step, the user ratings from the platform and the classifications of the items are gathered. Based on the user's preference degree for a categorized item and the item characteristic, this data is then utilized to calculate the user's preference similarity, which compares the user's preferences to those of other users who have similar preferences. The score and the user's similarity are both used in the calculation of the similarity between the two most recent users. After the user's prediction score for a particular item has been created using the collaborative filtering method, the work of providing product recommendations that the improved algorithm was designed to do has been completed.

2.2 Recommendation algorithm based on score and item attribute

2.2.1 User similarity based on item attribute preference

As the industry has evolved over the last several years, most crowdfunding platforms have settled into a predictable pattern when it comes to categorizing projects. There is also greater consistency and clarity in how crowdfunding sites are categorized when compared to other kinds of platforms. Film, television, advertising exhibitions, digital communication, the home, smart wear, audio-visual entertainment, travel positioning, the arts, cuisine, and other cultures are only some of the primary areas where crowdfunding initiatives are being developed.

If the project classification information of the crowdfunding platform is expressed as attribute characteristics, each project i can be represented by these attributes, denoted as X_i, as shown in formula (1).

$$X_i = \{arr_1, arr_2, \cdots, arr_k\} \quad (1)$$

Each term arr_c in the formula has two values, i.e. 1 and 0. If $arr_c = 1$, item X_i is considered to have item attribute c; if $arr_c = 0$, item X_i is considered not to have item attribute c. Each arr_c clarifies the relationship between item X_i and attribute c.
The item attribute information may be represented by Table 1 for n items.

Table 1. Example table of item attribute information.

	Arr 1	Arr 2	Arr 3	⋯	Arr k
Item 1	0	1	0	⋯	0
Item 2	1	0	0	⋯	0
Item 3	0	0	0	⋯	1
⋯	⋯	⋯	⋯	⋯	⋯
Item n	0	0	1	⋯	0

With the preceding table written as a matrix, a binary matrix detailing the association between items and their defining qualities may be computed with relative ease. Attribute qualities are listed in the rows of the matrix. Using the above formula, we can get the list item number (2).

$$downrightdownright\begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_{n-1} \\ X_n \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 1 & \cdots & 0 & 1 \\ 1 & 0 & \cdots & 0 & 0 \end{bmatrix} = [A_1 \;\; A_2 \;\; \cdots \;\; A_{k-1} \;\; A_k] \quad (2)$$

Where A_c represents the distribution of items in the classification feature attribute c.
The user item scoring matrix R composed of the scores of all users in the platform is shown in formula (3).

$$downrightR = \begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_m \end{bmatrix} = \begin{bmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,n} \\ r_{2,1} & r_{2,2} & \cdots & r_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{m,1} & r_{m,2} & \cdots & r_{m,n} \end{bmatrix} \quad (3)$$

Where R_u represents the row vector composed of all the scores of the user u, and represents the preference characteristics of the user based on the scores.

Therefore, the sum $S_{u,c}$ of the score values of u on all items containing attribute c can be obtained by formula (4).

$$S_{u,c} = R_u A_c \quad (4)$$

The total score S_u of the user u on all items is expressed as shown in formula (5).

$$S_u = \sum_{c=1}^{k} S_{u,c} \quad (5)$$

The preference degree $L_{u,c}$ of the user u for a certain item attribute c can be expressed by formula (6).

$$L_{u,c} = S_{u,c}/S_u \tag{6}$$

If the project in the platform contains k attribute features, that is, the total classification number is k, and the interest degree of u in all attributes is obtained by formula (6), then the similarity degree $sim_S(u,v)$ of the two users u and v based on the project attribute preference can be calculated by formula (7).

$$sim_S(u,v) = \frac{\sum_{c=1}^{k} L_{u,c}L_{v,c}}{\sqrt{\sum_{c=1}^{k} L_{u,c}^2}\sqrt{\sum_{c=1}^{k} L_{v,c}^2}} \tag{7}$$

To illustrate the above solution, the scoring data of User 1 , User 2 and User 3 on crowdfunding projects are shown in Table 2 (only some data are selected here as an example). The project is divided into two types, namely, communication digital (Digital 1-Digital 4) and food culture (Food 1-Food 4). Each project has one of the attributes. According to the data in the table, User 1 and User 2 do not score the same item. If only the score is used as the data source, the similarity between the two users is 0, that is, they are not related. The recommended service for user User 1 cannot be completed according to this method.

Table 2. Example of users' scoring on different types of items.

	Digital 1	Digital 2	Digital 3	Digital 4	Food 1	Food 2	Food 3	Food 4
User 1	5			5	2		3	
User 2		5	5			2		2
User 3	3		3		5			5

However, it can be clearly seen from the data distribution in the table that both User 1 and User 2 give a relatively high score (5 points) for the communication digital classification attribute, and both give a relatively low score (2 or 3 points) for the food culture classification attribute. From this perspective, the interest preferences of User 1 and User 2 are basically the same. Based on the classification attribute information, the calculated correlation degree of user User 1 and User 2 is $sim_S(user1, user2) = 0.9965$, which is close to the previous analysis. Therefore, it is feasible to calculate the similarity of users through the item attribute information, which can solve some problems caused by the single data feature.

2.2.2 Algorithm flow

To sum up, the algorithm design process based on score and item attribute is as follows:

1. Obtain platform data.

Users' ratings and items' categorizations are included in the data, from which a formula is derived to provide a "user item attribute preference matrix" (4). The user's item score matrix and attribute preference matrix serve as inputs for the enhanced algorithm, and their respective expressions are shown in formula (8).

$$R = \begin{bmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,n} \\ r_{2,1} & r_{2,2} & \cdots & r_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{m,1} & r_{m,2} & \cdots & r_{m,n} \end{bmatrix} \quad S = \begin{bmatrix} s_{1,1} & s_{1,2} & \cdots & s_{1,k} \\ s_{2,1} & s_{2,2} & \cdots & s_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ s_{m,1} & s_{m,2} & \cdots & s_{m,k} \end{bmatrix} \tag{8}$$

Where the user's rating of an item corresponds to a cell's value in the scoring matrix, and if that item is not rated, let. The matrix, the total number of users, and the total number of goods are all represented by. Number of users' ratings for products with attribute in the matrix. is the matrix, with attributes representing the number of items.

2. Calculate the score based user similarity matrix and the item attribute based user similarity matrix.

The scoring similarity $sim_R(u,v)$ of users u and v is obtained by formula (9) using the scoring matrix R, and the similarity matrix SIM_R is obtained, as shown in formula (10).

$$sim(u,v) = \frac{\sum_{i\in I_{uv}}(r_{u,i}-\bar{r}_u)(r_{v,i}-\bar{r}_v)}{\sqrt{\sum_{i\in I_{uv}}(r_{u,i}-\bar{r}_u)^2}\sqrt{\sum_{i\in I_{uv}}(r_{v,i}-\bar{r}_v)^2}} \tag{9}$$

$$SIM_R = \begin{bmatrix} sim_R(1,1) & sim_R(1,2) & \cdots & sim_R(1,m) \\ sim_R(2,1) & sim_R(2,2) & \cdots & sim_R(2,m) \\ \vdots & \vdots & \ddots & \vdots \\ sim_R(m,1) & sim_R(m,2) & \cdots & sim_R(m,m) \end{bmatrix} \tag{10}$$

Then, using formula (6), the user's preference degree $L_{u,c}$ for the item feature attribute is obtained based on the attribute preference matrix S. For the users u and v, the similarity matrix SIM_S can be obtained by calculating the similarity $sim_S(u,v)$ of the two based on the item attributes using formula (7), as shown in formula (11).

$$SIM_S = \begin{bmatrix} sim_S(1,1) & sim_S(1,2) & \cdots & sim_S(1,m) \\ sim_S(2,1) & sim_S(2,2) & \cdots & sim_S(2,m) \\ \vdots & \vdots & \ddots & \vdots \\ sim_S(m,1) & sim_S(m,2) & \cdots & sim_S(m,m) \end{bmatrix} \tag{11}$$

3. Combining the user similarity based on the score and the similarity based on the item attribute feature, the final similarity between users is weighted.

Combined with $sim_R(u,v)$ and $sim_S(u,v)$, the weights ω of the two are comprehensively adjusted to obtain the final similarity between user u and user v, as shown in formula (12).

$$sim(u,v) = \omega \times sim_R(u,v) + (1-\omega) \times sim_S(u,v) \tag{12}$$

The formula (13) can be obtained by bringing $sim_R(u,v)$ and $sim_S(u,v)$ into expression (12).

$$sim(u,v) = \omega \times \frac{\sum_{i\in I_{uv}}(r_{u,i}-\bar{r}_u)(r_{v,i}-\bar{r}_v)}{\sqrt{\sum_{i\in I_{uv}}(r_{u,i}-\bar{r}_u)^2}\sqrt{\sum_{i\in I_{uv}}(r_{v,i}-\bar{r}_v)^2}} + (1-\omega) \times \frac{\sum_{c=1}^{k} L_{u,c}L_{v,c}}{\sqrt{\sum_{c=1}^{k} L_{u,c}^2}\sqrt{\sum_{c=1}^{k} L_{v,c}^2}} \tag{13}$$

In the formula, ω and $1\text{-}\omega$ are the weight values of two types of similarity, and its magnitude directly determines which kind the outcome is more attracted to. Adjust $\omega = 1$, and the recommendation process is comparable to similarity calculation based on item score alone. If is toggled on $\omega = 0$, it indicates that the algorithm relies only on the user's item attribute preference matrix and does not have any kind of direct connection to any particular item. In addition, the formula described above includes the following additional components: the set of things on which users have left comments; I_{uv} the average score of users; and the average score of users. The weight is a changeable parameter. In the experiment in Section 3 of this

chapter, it will be altered to find an appropriate value to obtain a better prediction effect.
Calculate the similarity between all user combinations to obtain the final similarity matrix, as shown in formula (14).

$$SIM = \begin{bmatrix} sim(1,1) & sim(1,2) & \cdots & sim(1,m) \\ sim(2,1) & sim(2,2) & \cdots & sim(2,m) \\ \vdots & \vdots & \ddots & \vdots \\ sim(m,1) & sim(m,2) & \cdots & sim(m,m) \end{bmatrix} \tag{14}$$

4. Generate recommendations.
To be able to provide a recommendation, it is necessary to first get a forecast of the user's score value for the item in question. Users are selected from the matrix based on their proximity to the user being used to construct a set, with the values of the row vector of the user being arranged in decreasing order. The following is the formula that will be used to calculate the prediction score: (15).

$$\hat{r}_{u,i} = \bar{r}_u + \frac{\sum_{n \in KNB_u} sim(u,n) \times (r_{n,i} - \bar{r}_n)}{\sum_{n \in KNB_u} (|sim(u,n)|)} \tag{15}$$

The system sorts the calculated predicted values from high to low, and selects the top n items with high scores to recommend to the user.
According to the sorting of the improved algorithm flow, the design flow of personalized recommendation algorithm of crowdfunding platform based on user scores and project attributes is shown in Figure 1.

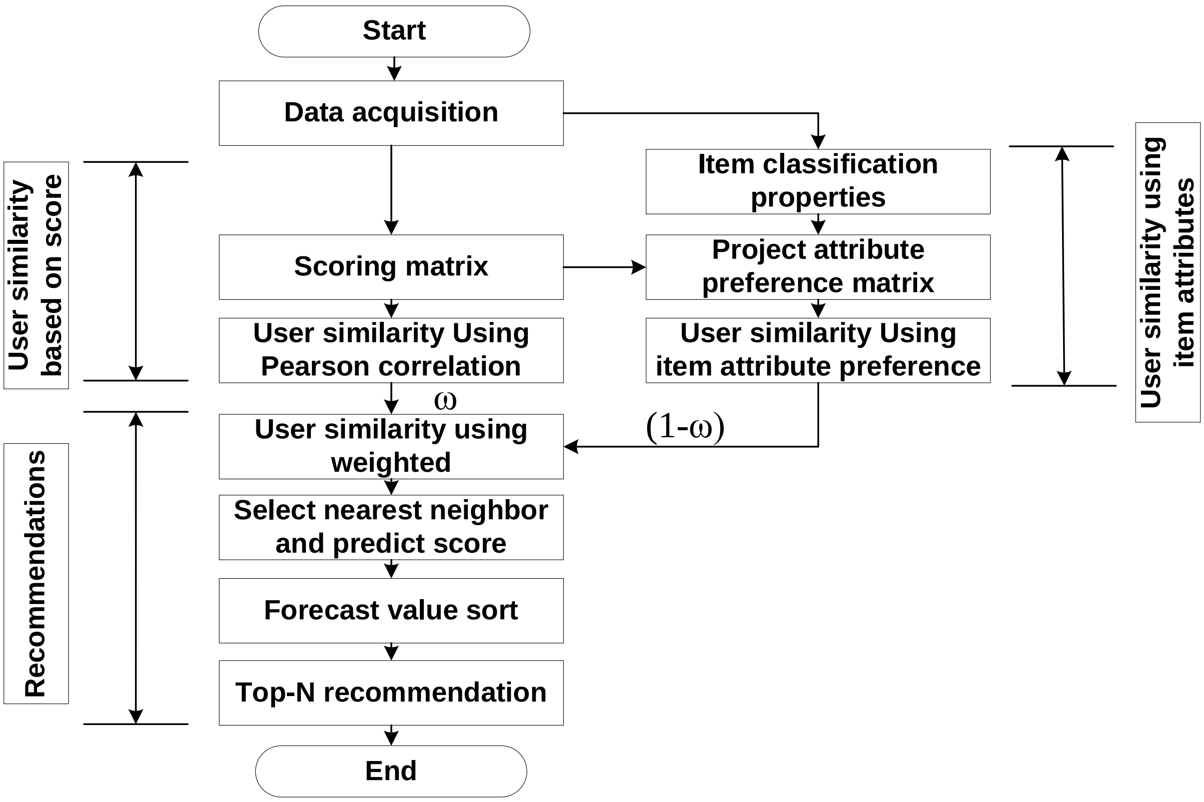

Figure 1. Personalized recommendation algorithm flow of crowdfunding platform based on scores and project attributes.

EXPERIMENTS AND RESULTS

3.1 Dataset

On the crowdfunding website, the main sorts of lists that may be mined for data right now are project information lists, categorization lists, user information lists, score lists, and support lists. As input, the recommendation system that was constructed for this research makes use of the score data from the real crowdfunding site. When selecting an approach, the recommendation system takes into consideration the preferences of the users. This method is known as collaborative filtering. The data is delivered using a format known as Top-N.
In this work, the authors advocate using the mean absolute error (MEA) and the F index (F-measure) to evaluate performance.
Using the following equation, one can get the average absolute error (Gauri et al., 2021) MEA (16).

$$MAE = \frac{1}{|\tau|}\sum_{(u,i)\in\tau}\left|\hat{r}_{u,i} - r_{u,i}\right| \tag{16}$$

Where is the number of items in the test set, stands for the size of the test set, and the projected score and the actual score of the user on the item in the test set, respectively.

$$F - measure = \frac{2\times Precision\times Recall}{Precision+Recall} \tag{17}$$

The system's total performance is more appropriately represented by the F-measure. The higher the F-measure, the more effective the advice and the more closely it will fit the user's requirements. Accordingly, the greater the F-measure and the lower the MEA value, the more precise the forecast.

3.2 Experiments and results

The experiments included in this publication are broken down as follows.

3.2.1 The 1st Experiment and results

A score-and-item-attribute based collaborative filtering method is implemented. The ratio between user similarity based on score and user similarity based on item attribute is affected by the setting. Experiment with a range of neighbor counts, starting at zero and going up by increments of 0.1. Five separate tests were conducted to arrive at each figure. The outcomes of the experiments are shown in Table 3.

Table 3. MAE value of recommended algorithm when ω value changes.

ω	0.1	0.2	0.3	0.4	0.5
MAE	1.23469	1.17967	1.15420	1.11931	1.08894
ω	0.6	0.7	0.8	0.9	1.0
MAE	1.04812	1.03950	1.07784	1.10575	1.13845

Taking ω as the abscissa and MAE as the ordinate, the change curve is shown in Figure 2.

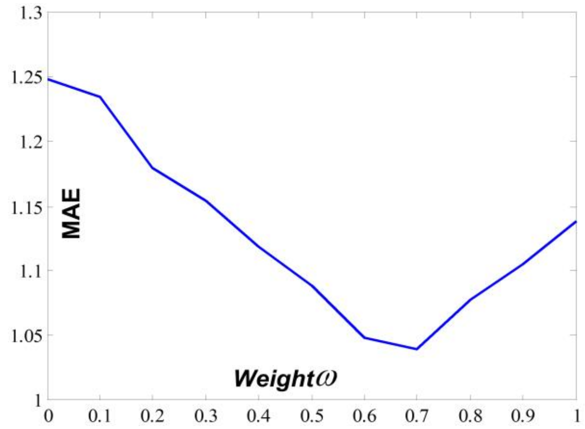

Figure 2. Adjustments to the algorithm based on the newly included weighted score and item properties.

The shifting curve and the table both show that as the weight is raised, the average absolute inaccuracy of the system first decreases and then increases again. This can be demonstrated to be the case when the weight is increased. The suggestion effect is maximized with a weight of 0.7, which also results in the lowest average absolute error. It has been shown that the recommendation algorithm that considers similarities between users and between objects performs better at producing suggestions than those that just evaluate the ratings or characteristics of items In order to ensure that the advice provided have the greatest possible impact, the weight parameter in the trials described in this article is often adjusted to a value of 0.7.

3.2.2 The 2nd Experiment and results

Set the weight $\omega = 0.7$, and the size K of the nearest neighbor set is 5 to 95 respectively, and the increment step is 5. Calculate the average absolute error (MAE) of the collaborative filtering recommendation algorithm before and after adding the item attribute preference improvement, and the results are shown in Table 4.

Table 4. MAE values of algorithms before and after adding item attribute preference when the number of neighbors K is different.

K	5	15	25	35	45
MAE (score)	1.16089	1.16207	1.15467	1.13560	1.14282
MAE (score + Project properties)	1.06871	1.05545	1.04481	1.04222	1.03947
K	55	65	75	85	95
MAE (score)	1.14463	1.14137	1.13867	1.13824	1.14088
MAE (score + Project properties)	1.03160	1.03553	1.03375	1.03447	1.03675

Figure 3 depicts a rectangular coordinate system in which the value change curve based on the user score and the value change curve based on the user score and the item attribute are both depicted.

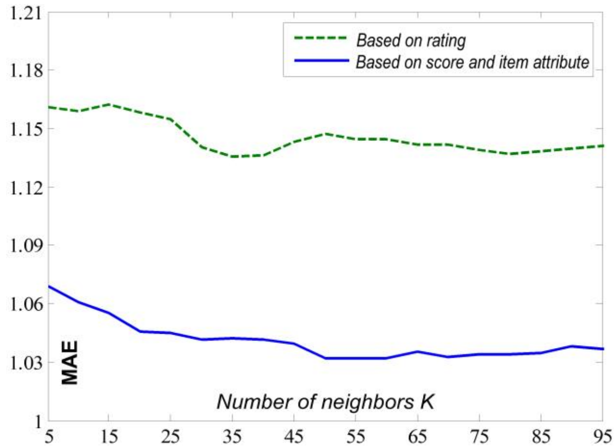

Figure 3. Algorithmic change in respect to neighbor count before to and after the addition of preference attributes to items.

First, we calculate the and of the and improved approaches. Next, we find the corresponding to the number of unique neighbors based on these two indicators. Finally, we get the corresponding to the distance between each neighbor. The results of the trials are shown in Table 5.

Table 5. Evaluation index of recommendation algorithm before and after improvement based on project attribute.

K	5	25	45	65	85
Precision (score)	0.64912	0.64444	0.64889	0.65333	0.65179
Precision (score + attribute)	0.67050	0.68354	0.69130	0.69396	0.69298
Recall (score)	1.00000	0.97973	0.98649	0.99324	0.98649
Recall (score + attribute)	1.00000	0.98780	0.96951	0.96341	0.96341
F-measure (score)	0.78723	0.77748	0.78284	0.78820	0.78495
F-measure (score + attribute)	0.80275	0.80798	0.80711	0.80678	0.80612

The F-measure is calculated by adding the accuracy and recall scores together. Figure 4 demonstrates the improvement in the algorithm's indexes, and the change in the curve can be computed by using the abscissa (the number of surrounding users) and the ordinate (the change in the algorithm's indexes).

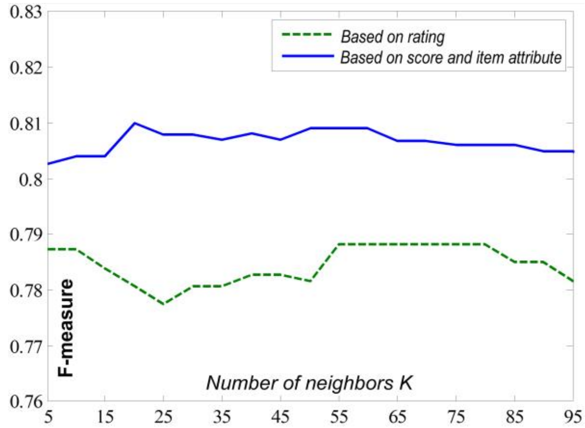

Figure 4. Change curve of F-measure with the number of neighbors K before and after improvement.

DISCUSSION

Following the addition of the information regarding the item attributes, Figure 3 demonstrates that the average absolute error of the improved recommendation algorithm is lower than that of the initial collaborative filtering method by an average of approximately 0.1. This is a significant improvement. The problem that was brought up by the one piece of data is, to some extent, solved.

Table 5 demonstrates that the recommendation algorithm that considers both the score and the item attribute performs noticeably better than the recommendation algorithm that merely considers the score when the number of neighbors changes. This is the case when compared to the recommendation algorithm that merely considers the score. Figure 3 illustrates the impact of the outdated recommendation algorithm that was based solely on the scores, whereas Figure 4 illustrates the effect of the recommendation algorithm that was based on scores as well as items attribute. These figures show that the improved algorithm has resulted in positive outcomes for the crowdfunding platform.

CONCLUSION

The major objective of this effort is to find a solution to the problem of cold start. Investigating the factors that led to the cold start problem led us to discover that the singularity of the data that was fed into the collaborative filtering algorithm was the root of the problem. This study improves the collaborative filtering algorithm by taking into account the features of unambiguous project categorization of crowdfunding platforms and combining them with the user similarity based on score and the user similarity based on project attribute preference. Additionally, this study takes into account the features of crowdfunding platforms' unambiguous project categorization. After the algorithm has been implemented, readjusting the weights of the two similarities to reflect their degree of similarity enables additional analysis of the modified algorithm's performance. According to the results, using an improved form of collaborative filtering helps to mitigate the detrimental consequences of getting off to a chilly start. The results of the experiments support the conclusion that the improved method proposed in the study is applicable. This recommendation method is expected to be used in online education platforms to personalize recommend courses according to user's preferences, to provide a reference for more efficient utilization of educational resources.

REFERENCES

Goldberg, D., Niehols, D., Oki, B.M. & Terry, D., 1992. Using collaborative filtering to weave an information tapestry. Communications of the ACM, 35 (12): 61-70.

Sarma, D., Mittra, T. & Hossain, M. S., 2021. Personalized book recommendation system sing machine learning algorithm. International Journal of Advanced Computer Science and Applications.12 (1).

Koren, Y., Rendle, S. & Bell, R., 2022. Advances in collaborative filtering. Recommender systems handbook.91-142.

Su, X. & Khoshgoftaar, T.M., 2009. A survey of collaborative filtering techniques. Advances in Artificial Intelligence.2009 (12): 4.

Khanal, S.S., Prasad, P.W.C., Alsadoon, A. & Maag, A., 2020. A systematic review: machine learning based recommendation systems for e-learning. Education and Information Technologies: The Official Journal of the IFIP Technical Committee on Education.25 (4).

Merialdo, A.K.B., 1999. Clustering for collaborative filtering applications. Intelligent Image Processing, Data Analysis & Information Retrieval 3 (1999): 199.

Breese, J.S., Heckerman, D. & Kadie, C., 1998. Empirical analysis of predictive algorithms for collaborative filtering. In Proceeding of the 14th Conference on Uncertainty in Artificial Intelligence, Redmond, 43-52.

Hofmann, T., 2004. Latent semantic models for collaborative filtering. Acm Transactions on Information Systems.22 (1): 89-115.

Shani, G., Brafman, R. & Heckerman, D., 2012. An MDP-based recommender system. Journal of Machine Learning Research.6 (1): 1265-1295.

Margahny, M.H. & Mitwaly, A.A., 1994. Fast algorithm for mining association rules. In Proceedings of the 20th International Conference on Very Large Databases, Santiago.487-499.

Huang, M., 2019. Research on recommendation technology based on user portrait. Proceedings of 2019 3rd Scientific Conference on Mechatronics Engineering and Computer Science.472-480.

Ziegler, C.N., Mcnee, S.M., Konstan, J.A. & Lausen, G., 2005. Improving recommendation lists through topic diversification. In Proceedings of the 14th International World Wide Web Conference, Chiba, Japan, 22-32.

Weng, S.S. & Liu. M.J., 2004. Feature-based recommendations for one-to-one marketing. Expert Systems with Applications.26 (4): 493-508.

George, T. & Merugu, S., 2005. A scalable collaborative filtering framework based on Co-Clustering. IEEE International Conference on Data Mining. Washington.625-628.

Sobecki, J., 2006. Implementations of web-based recommender systems using hybrid methods. International Journal of Computer Science & Applications.3 (3): 52-64.

Gauri, S. & Neeraj, R., 2021. Performance evaluation of health recommendation system based on deep neural network. IOP Conference Series: Materials Science and Engineering.1131 (1).