Higher Education Online Courses Personalized Recommendation Algorithm Based on Score and Attributes

Wenbin Wu¹, Bixi Wang¹, Wenfeng Zheng², Yan Liu² and Lirong Yin³,*

¹ Xi'an University No .1 Science and Technology Road, Xi'an, 710065 P.R.C
² University of Electronic Science and Technology of China, Chengdu, 611731, R.P.C
³ Louisiana State University Baton Rouge, LA 70803, United States.
* Corresponding Author Email: lirongyu5@lsu.edu

Abstract. Education through the online platform has gained popularity. While the online course keeps increasing, its benefit does not increase in a proportional way. Personalized recommendation systems can effectively solve this problem by mining users' interests and preferences. Aiming at the cold start problem, this paper improves the collaborative filtering algorithm by combining the user score and project attribute characteristics of the online course platform. The network communication technology is used to obtain the user ratings and project attribute data to verify the feasibility of the recommendation algorithm, adjust the parameters in the model, and compare the accuracy of the algorithm. The designed algorithm can provide accurate and rapid personalized recommendation services, which is convenient for users and conducive to the development of the platform. The improved algorithm solves the cold start problem compared with the traditional algorithm with a significantly improved prediction accuracy. The scheme can also be modified according to the changes in user preferences and can achieve good real-time recommendation effect.

1 Instructions

With the rapid development of network technology, education through the online platform has gained popularity and is growing at a fast rate. In order to solve the difficulty of information overload in the platform, people designed and proposed Personalized Recommendation System (Resnick and Varian 1997). Popular recommendations include content-based recommendations (Aggarwal, Gates, and Yu 2002), Association Rules recommendations (Lin, Alvarez, and Ruiz 2000), and Collaborative Filtering recommendations (Haiyan, Feng, and Lihong 2005). The most common and mature algorithm is collaborative filtering algorithm.

Zhao and Shang (Zhao and Shang 2010) implemented user-based CF algorithm on cloud computing platform Hadoop (Ni et al. 2019; Ding et al. 2019), so that all processors can complete tasks at the same time and realize linear acceleration. Wei et al. (Wei et al. 2012) proposed a collaborative filtering recommendation algorithm based on item category similarity and interest measure. Cai et al. (Cai et al. 2013) proposed TyCo, a collaborative filtering recommendation method based on typicality. Yin et al. (Yin et al. 2016) proposed a tag-based system to generate recommendations for TV users.

Collaborative filtering technology has achieved great success in theoretical exploration and in practice. However, the problems of cold start still exist in the recommendation system. Cold start refers to that the user does not immediately initiate evaluation or support behaviors during the initial launch of the system (Su and Khoshgoftaar 2009). Ziegler et al. (Ziegler et al. 2005) proposed a scheme of subject diversification, which is a new method aimed at balancing and diversifying personalized recommendation lists. Liu and Weng analyzed consumers' preferences for product proprietary functions according to transaction records and product characteristics, and then generated customer
preference rules (Weng and Liu 2004; Chen et al. 2020). George T et al. proposed a collaborative filtering algorithm based on weighted clustering for analysis of users and items (George and Merugu 2005; Zheng et al. 2016). Sobecki proposed a hybrid recommendation using project content attributes, user ratings and user statistics (Sobecki 2006; Zheng et al. 2017). Ning Yuan et al. introduced the time-varying values of user attribute data and project score into the traditional recommendation algorithm in 2012 (Yuan, Limin, and Zhiwei 2011; Yin et al. 2019).

To solving the cold start problem, we improved the user-based collaborative filtering system by adding classification information. We obtain the user’s rating information and course classification information from the platform. We find out the user’s preference for categories and then the user’s similarity using category preference. The similarity between two users is calculated by weighting, and the user’s prediction score on a certain item is calculated based on the collaborative filtering algorithm.

2 Dataset
The algorithm adopted in this paper is based on the user-based collaborative filtering algorithm, which needs to obtain user list, item list and user rating data of items. Through browsing a number of online course platforms. After completion, we have a total of 1095 online course projects, which are divided into different categories. There were 277 user-rated items, with a total of 4,645 user ratings and 3,445 users participating in the ratings. There are 1084 course projects supported by users, with 172,841 rating, 88,153 users participating, and 89,427 people available.

3 Methods and Experiments
For each project i, these attributes can be represented and denoted as X_i:

\[
X_i = \{arr_i, arr_{i,L}, \ldots, arr_{i_k}\}
\]  \hspace{1cm} (1)

Each of these terms has two values, 1 and 0. Where 1 represents X attribute c, while 0 represents X does not have attribute c. Each term illustrates the relationship between item X, and attribute c. The scores of all users on the platform constitute the user-item score matrix R. Therefore, the sum of the score values S_{ui} for items containing attribute c and the total score S_u of user u on all items is shown by formula (2).

\[
S_{u, c} = R_u g A_c \quad S_u = \sum_{c=1}^{k} S_{u, c}
\]  \hspace{1cm} (2)

R_u represents the composed of all scores of user u, and S represents the user's preferences based on scores. User u's preference for an item attribute c (I_{u,c}) can be expressed by the division of the sum of the composed score over the selected composed score. The similarity degree of two users based on item attribute preferences \text{sim}_u(u, v) can be calculated by formula (3).

\[
\text{sim}_u(u, v) = \frac{\sum_{c=1}^{k} I_{u,c} I_{v,c}}{\sqrt{\sum_{c=1}^{k} I_{u,c}^2} \sqrt{\sum_{c=1}^{k} I_{v,c}^2}}
\]  \hspace{1cm} (3)

The input data of the improved algorithm are user-item score matrix R and user-item attribute preference matrix S, and the representation is shown in formula (4).

\[
R = \begin{bmatrix}
\ell_{1,1} & \ell_{1,2} & \ell_{1,n} \\
\ell_{2,1} & \ell_{2,2} & \ell_{2,n} \\
M & M & O
\end{bmatrix}
S = \begin{bmatrix}
s_{1,1} & s_{1,2} & L & s_{1,k} \\
s_{2,1} & s_{2,2} & L & s_{2,k} \\
M & M & O & M
\end{bmatrix}
\]  \hspace{1cm} (4)

Where, the value \( \ell_{i,j} \) in the scoring matrix R is user i’s rating of item j, and if i does not rate j, let \( \ell_{i,j} = 0 \). The matrix R is the \( m \times n \) matrix, m is the total number of users, and n is
the total number of items. For $S$ matrix, $s_{i,c}$ is the total score of user $i$ for items with attribute $c$. S is the m x k matrix, where $k$ is the number of feature attributes of the item. The score matrix R is used to obtain the score similarity SIM$_{g}(u, v)$ of user u and v, and the similarity matrix is obtained:

$$sim(u, v) = \frac{\sum_{i \in I(u)} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I(u)} (r_{ui} - \bar{r}_u)^2} \sqrt{\sum_{i \in I(v)} (r_{vi} - \bar{r}_v)^2}}$$  

(5)

The similarity matrix can be obtained in the same form. Combined the two similarity matrices, the use weight $\omega$ of the two is adjusted comprehensively to obtain the final similarity between user u and user v, as shown in formula (6).

$$sim(u, v) = \omega \times \frac{\sum_{i \in I(u)} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I(u)} (r_{ui} - \bar{r}_u)^2} \sqrt{\sum_{i \in I(v)} (r_{vi} - \bar{r}_v)^2} } + (1 - \omega) \times \frac{\sum_{i \in I(u,v)} L_{ui} L_{vi}}{\sqrt{\sum_{i \in I(u)} L_{ui}^2} \sqrt{\sum_{i \in I(v)} L_{vi}^2} }$$

(6)

In the formula, $\omega$ and $1-\omega$ are the weight values of two similarity degrees. Adjusting $\omega=1$, the recommendation algorithm is equivalent to similarity calculation based only on project score. If $\omega=0$ is set, the algorithm only adopts the user-item attribute preference matrix. In addition, $L_{uv}$ is the set of items that both user $u$ and $v$ have commented, $\bar{r}_u$ is the average score of user $u$, $\bar{r}_v$ is the average score of user $v$. The weight $\omega$ is a variable parameter.

To generate recommendations, first need to predict user u's score value $\hat{r}_{ui}$ for project $i$. The values of row vectors of user $u$ in the E matrix are arranged in descending order, K users are taken as the nearest neighbors of $u$ to form set $KNB_u$. Prediction score $\hat{r}_{ui}$ can be obtained, as shown in Equation (7).

$$\hat{r}_{ui} = \bar{r}_u + \frac{\sum_{m \in KNB_u} sim(u, n)(r_{m,i} - \bar{r}_n)}{\sum_{m \in KNB_u} [sim(u, n)]}$$

(7)

Where $sim(u, n)$ represents the similarity of user $u$ and $n$, $\bar{r}_u$ and $\bar{r}_n$ represent the average score of user $u$ and $n$ respectively, and $r_{m,i}$ represents the score value of user $n$ for item $i$. The system sorts the calculated predicted values and selects the top N items with high scores to recommend to users.

For similarity calculation, the setting of $\omega$ can affect the proportion of user similarity. Set the default number of K=30, which is between 0 and 1 and increases by 0.1 each time and conduct experimental tests. Each value is the average of 5 experiments. The results are shown in Table 1.

| $\omega$ | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 |
|----------|-----|-----|-----|-----|-----|
| MAE      | 1.23469 | 1.17967 | 1.15420 | 1.11931 | 1.08894 |
| $\omega$ | 0.6 | 0.7 | 0.8 | 0.9 | 1.0 |
| MAE      | 1.04812 | 1.03950 | 1.07784 | 1.10575 | 1.13845 |

Then set the weight $\omega=0.7$, the size $K$ of the nearest neighbor set is progressive from 5 to 95, and the step size is 5. Calculate the average absolute error (MAE) of the algorithm before and after adding the improvement. The results are shown in Table 2.

The improved algorithm is compared with the traditional algorithm. With the change of K, the MAE values of the algorithm before and after the improvement are shown in Table 3.
Table 3. The MAE values of the recommended algorithm before and after the overall improvement

| K   | MAE_score | MAE_Sco+attr | K   | MAE_score | MAE_Sco+attr |
|-----|-----------|--------------|-----|-----------|--------------|
| 5   | 1.16089   | 1.06871      | 55  | 1.14463   | 1.03160      |
| 15  | 1.16207   | 1.05545      | 65  | 1.14137   | 0.82723      |
| 25  | 1.15467   | 1.04481      | 75  | 1.13867   | 0.82559      |
| 35  | 1.13560   | 1.04222      | 85  | 1.13824   | 0.82600      |
| 45  | 1.14282   | 1.03947      | 95  | 1.14088   | 0.83214      |

| K   | MAE_score | MAE_Sco+attr | K   | MAE_score | MAE_Sco+attr |
|-----|-----------|--------------|-----|-----------|--------------|
| 5   | 1.15100   | 1.03160      | 55  | 0.88844   | 0.83367      |
| 15  | 1.13680   | 0.86126      | 65  | 1.04481   | 0.82723      |
| 25  | 1.13508   | 0.85114      | 75  | 1.04222   | 0.82559      |
| 35  | 1.13382   | 0.84245      | 85  | 1.03947   | 0.82600      |
| 45  | 1.14936   | 0.83777      | 95  | 1.14088   | 0.83214      |

Table 4. Classification of items to be predicted

| Users love | advice | not recommended |
|------------|--------|-----------------|
|            | $N_p$ ($Ture – Positive$) | $N_{fp}$ ($False – Negative$) |
| Users don’t like | $N_{fp}$ ($False – Positive$) | $N_{fn}$ ($Ture – Negative$) |

The raw data is divided into training set and test set. The training set is used to predict the scores, then are compared with the actual scores of the users for evaluation. The main evaluation methods are precision and recall rate. The recommended precision and recall rate are represented in formula (8):

$$Precision = \frac{N_p}{N_p + N_{fp}} \quad Recall = \frac{N_p}{N_p + N_{fn}}$$

The commonly used index is F (F-measure), which is calculated as shown in formula (9).

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

F-measure more fully represents the performance of the system. Taking $\omega$ as the x-coordinate and MAE as the y-coordinate, the change curve is obtained as shown in Figure 1. It can be found that the average absolute error of the system first decreases and then increases. When the weight $\omega$ is 0.7, the average absolute error of the system is the smallest and the recommendation effect is the best. It can be explained that the algorithm combining the two kinds of similarity has more accurate performance than the recommendation algorithm based only on score ($\omega=1$) or item attribute ($\omega=0$).
Figure 1. The change of MAE with weight based on score and item attributes

Figure 2. The MAE with changes K before and after adding item attribute preference

According to the change of the K, the change curve of MAE value based on user score in Experiment 2 and the change curve of MAE value based on user score and item attributes are represented. Compared with the average absolute error curves, the improved algorithm has a smaller average absolute error than the original algorithm. The accuracy rate (Precision) and recall rate (Recall) of the algorithm before and after the improvement were calculated, and the F-measure corresponding to different K was calculated. The experimental results are shown:

Table 5. Evaluation indexes of recommendation algorithm before and after attribute improvement

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

When the K changes, the accuracy rate of the algorithm based on score and attributes is significantly better than that based only on score. Algorithm indexes before and after improvement are compared. The K is taken as the abscissa and F-measure as the ordinate, and curve changes is obtained as shown in Figure 3.

Figure 3. F-measure with changing K before and after improvement

Figure 4. MAE with changing K before and after the overall improvement

According to Figure 1 and 2, the recommendation algorithm based on score and project attributes is significantly better than the traditional algorithm, and the algorithm improvement has achieved. The accuracy rate (Precision), the recall rate (Recall), and F-measure of the recommended algorithm before and after improvement were tested, and the results were shown in Table 6. The change of F-measure before and after improvement is expressed with the K as a variable, as shown in Figure 5.
Table 6. Evaluation indexes with different K

| K   | Precision Before | Precision After | Recall Before | Recall After | F-measure Before | F-measure After |
|-----|------------------|-----------------|--------------|-------------|-----------------|----------------|
| 5   | 0.63637          | 0.78473         | 1.00000      | 0.87578     | 0.77778         | 0.82775        |
| 25  | 0.63954          | 0.79022         | 0.98137      | 0.88199     | 0.77441         | 0.83359        |
| 45  | 0.64468          | 0.79022         | 0.98137      | 0.88199     | 0.77816         | 0.83359        |
| 65  | 0.64613          | 0.79022         | 0.98758      | 0.88199     | 0.78117         | 0.83359        |
| 85  | 0.64990          | 0.78596         | 0.98137      | 0.88199     | 0.78196         | 0.83121        |

Figure 5. F index with different K

We found from the curve variation trend that the MAE of the improved recommendation algorithm is significantly lower than that of the improved recommendation algorithm. The improved algorithm has higher accuracy rate. To sum up, the improved algorithm has achieved obvious effect and reduced the influence of the algorithm caused by data sparsity and cold start.

5 Conclusion
Aiming at solving the cold start, we design an improvement scheme of personalized recommendation algorithm based on score and project attributes. The improved algorithm was tested by using the user rating data and project classification attributes obtained from the website. The following conclusions can be drawn from the experiment. When the weight is 0.7, the average absolute error of the system is the smallest and the recommendation effect is the best. It can be explained that the recommendation algorithm combining the two kinds of similarity has more accurate recommendation performance than the recommendation algorithm based only on score ($\omega=1$) or item attribute ($\omega=0$). Compared with the average absolute error curve before and after the improvement, the improved recommendation algorithm with the addition of project attribute information has a smaller average absolute error than the original collaborative filtering algorithm with an average reduction of about 0.1. Finally, this paper compares the overall improved algorithm with the pre-improved algorithm. The experimental results show that the improved algorithm improves the accuracy of prediction by about 0.04. The accuracy rate (Precision) improved by about 0.05, the recall rate (Recall) decreased by about 0.1, and the F-measure improved by 0.05.

The effect of the recommendation algorithm based on score and project attributes is obviously better than the traditional recommendation algorithm based on score only, and the algorithm improvement has achieved effects. The results show that the improved algorithm has application value and solves the cold start problem of the original algorithm.
6 Reference

[1] Aggarwal, Charu Chandra, Stephen C Gates, and Philip Shi-lung Yu. 2002. "System and method for generating taxonomies with applications to content-based recommendations." In: Google Patents.

[2] Cai, Yi, Ho-fung Leung, Qing Li, Huaqing Min, Jie Tang, and Juanzi Li. 2013. 'Typicality-based collaborative filtering recommendation', IEEE Transactions on Knowledge and Data Engineering, 26: 766-79.

[3] Chen, Xiaobing, Lirong Yin, Yulin Fan, Lihong Song, Tingting Ji, Yan Liu, Jiawei Tian, and Wenfeng Zheng. 2020. 'Temporal evolution characteristics of PM2. 5 concentration based on continuous wavelet transform', Science of The Total Environment, 699: 134244.

[4] Ding, Yueming, Xia Tian, Lirong Yin, Xiaobing Chen, Shan Liu, Bo Yang, and Wenfeng Zheng. 2019. "Multi-scale Relation Network for Few-Shot Learning Based on Meta-learning." In International Conference on Computer Vision Systems, 343-52. Springer.

[5] George, Thomas, and Srujana Merugu. 2005. "A scalable collaborative filtering framework based on co-clustering." In Fifth IEEE International Conference on Data Mining (ICDM'05), 4 pp.: IEEE.

[6] Haiyan, Zhang, Ding Feng, and Jiang Lihong. 2005. 'A collaborative filtering recommendation method based on fuzzy clustering', Computer Simulation, 22: 144-47.

[7] Lin, Wei, Sergio A Alvarez, and Carolina Ruiz. 2000. 'Collaborative recommendation via adaptive association rule mining', Data Mining and Knowledge Discovery, 6: 83-105.

[8] Ni, Xubin, Lirong Yin, Xiaobing Chen, Shan Liu, Bo Yang, and Wenfeng Zheng. 2019. "Semantic representation for visual reasoning." In MATEC Web of Conferences, 02006. EDP Sciences.

[9] Resnick, Paul, and Hal R Varian. 1997. 'Recommender systems', Communications of the ACM, 40: 56-58.

[10] Sobecki, Janusz. 2006. 'Implementations of web-based recommender systems using hybrid methods', IJCSA, 3: 52-64.

[11] Su, Xiaoyuan, and Taghi M Khoshgoftaar. 2009. 'A survey of collaborative filtering techniques', Advances in artificial intelligence, 2009.

[12] Wei, Suyun, Ning Ye, Shuo Zhang, Xia Huang, and Jian Zhu. 2012. "Item-based collaborative filtering recommendation algorithm combining item category with interestingness measure." In 2012 International Conference on Computer Science and Service System, 2038-41. IEEE.

[13] Weng, Sung-Shun, and Mei-Ju Liu. 2004. 'Feature-based recommendations for one-to-one marketing', Expert Systems with Applications, 26: 493-508.

[14] Yin, Fuhuan, Xiaowei Liu, Wanying Ding, and Ruizhe Zhang. 2016. "Tag-based collaborative filtering recommendation algorithm for TV program." In 2016 13th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), 47-52. IEEE.

[15] Yin, Lirong, Xiaolu Li, Wenfeng Zheng, Zhengtong Yin, Lihong Song, Lijun Ge, and Qingchuan Zeng. 2019. 'Fractal dimension analysis for seismicity spatial and temporal distribution in the circum-Pacific seismic belt', Journal of Earth System Science, 128: 22.

[16] Yuan, Ning, Liu Limin, and Xu Zhiwei. 2011. "Research on personalized recommendation algorithm based on user model and user-project matrix." In Proceedings of 2011 International Conference on Computer Science and Network Technology, 2400-02. IEEE.

[17] Zhao, Zhi-Dan, and Ming-Sheng Shang. 2010. "User-based collaborative-filtering recommendation algorithms on hadoop." In 2010 Third International Conference on Knowledge Discovery and Data Mining, 478-81. IEEE.

[18] Zheng, Wenfeng, Xiaolu Li, Lirong Yin, and Yali Wang. 2016. 'The retrieved urban LST in Beijing based on TM, HJ-1B and MODIS', Arabian Journal for Science and Engineering, 41: 2325-32.

[19] Zheng, Wenfeng, Xiaolu Li, Lirong Yin, Zhengtong Yin, Bo Yang, Shan Liu, Lihong Song, Yu Zhou, and Yanhong Li. 2017. 'Wavelet analysis of the temporal-spatial distribution in the Eura-
sia seismic belt', International Journal of Wavelets, Multiresolution and Information Processing, 15: 1750018.

[20] Ziegler, Cai-Nicolas, Sean M McNee, Joseph A Konstan, and Georg Lausen. 2005. "Improving recommendation lists through topic diversification." In Proceedings of the 14th international conference on World Wide Web, 22-32.