A Mixed Recommended Algorithm Combining User Properties and Item Features

Ning Xia\textsuperscript{1,2}, Shaofei Wu\textsuperscript{1,2,*}, Shenkuo Wang\textsuperscript{1,2} and Huajie Zhang\textsuperscript{1,2}

\textsuperscript{1}Laboratory of Intelligent Robot, Wuhan Institute of Technology, Wuhan, China
\textsuperscript{2}School of Computer Science & Engineering, Wuhan Institute of Technology, 420205, Wuhan, China

*Corresponding author e-mail: 14590041@qq.com

Abstract. In response to the problems of traditional recommended algorithm, such as data sparsity, changing of users’ interest, etc., this thesis proposes a mixed recommended algorithm blending User Properties and item features. First, propose a recommended algorithm based on User Properties after leading in users’ interaction factor and property weight in traditional similarity computation. Second, decompose the user-item matrix, and build the preference matrix and rating frequency matrix of user-item features. And then it proposes a recommended algorithm based on item features through leading in interest’s stable phase and improve the time attenuation function. Lastly, it generates a new mixed recommended algorithm after linearly combining these two algorithms. The Movielens dataset is used in this experiment, which turns out that the mean absolute error of the recommended algorithm proposed in this thesis significantly decreases comparing to traditional recommended algorithm.

1. Introduction

With the rapid development of internet, the data increase by exponential growth. In the face of a bulk of data and overloaded information, it is crucial to recommend a proper product according to users’ preferences. The collaborative filtering recommended algorithm helps a lot when recommending products for users, however it does have the problems of data scarcity [1], changing of users’ interest [2], cold-start[3] etc.

The common recommended algorithms at present include collaborative filtering recommended algorithm [4], Model-based collaborative filtering [5], content-based recommended algorithm [6], Grid-based recommendation algorithm [7] etc. Recently, some scholars start with clustering [8], socialnetwork [9], physical methods [10] and Deep learning [11]. such as Wei [12] proposed collaborative filtering recommended algorithm based on multiple trust, which alleviates the problems of cold boot and data scarcity by combining dominant and implicit trust factors. However, these algorithms neglect the problem that the users’ interest is changing over time. Therefore, taking time factor into account contributes to increase the recommendation systems’ accuracy.

As time goes on, users’ interest is decreasing, but the decreasing speed differs in different periods. According to the Ebbinghaus Forgetting Curve and stable phase of users’ interest, the improved time attenuation function is that users’ interest remains as the time goes on in the initial phase, and then it accords with the Ebbinghaus Forgetting Curve. Moreover, users’ involvement in marking the item is
different. Users’ involvement should be considered when we recommend the product on the basis of users’ marking on item. Therefore, the idea of fusing users’ interaction factor to distinguish users’ involvement when we compute similarity is posed in this thesis. On the other hand, preference matrix and rating frequency matrix of user-item features are obtained by decomposing user-item rating matrix. Users’ rating frequency reflects users’ true preferences, thus improving the similarities by leading in users’ rating frequency on item features.

2. Relevant Theories

2.1. Users’ Engagement
Every user has their own habit. Some are willing to give a rating, while some are reluctant. Therefore, users’ rating track should be considered when computing the similarity of the users who rate the same item so as to neutralize the effect caused by users’ habits. As for User m and User n, their interaction factor $u_{iif}$ can be defined as:

$$u_{iif} = \frac{m_i \cap n_i}{m_i}$$ (1)

In this formula, $m_i$ and $n_i$ are the rating collections of User m and User n accordingly. $m_i \cap n_i$ is the item set which is rated by both User m and User n.

2.2. Weight Factor of User Properties
Take Movielens dataset as an example to build the matrix of User Properties.

| Users  | Age | Gender | Job     |
|--------|-----|--------|---------|
| $u_1$  | 23  | M      | student |
| $u_2$  | 42  | F      | teacher |
| ...    | ... | ...    | ...     |
| ...    | ... | ...    | ...     |
| $u_n$  | 14  | F      | writer  |

When computing the weight factor of User Properties, the age property can be divided into 7 stages depending on their life experience, which are 0-12 years old, 13-18 years old, 19-24 years old, 25-35 years old, 35-44 years old, 45-59 years old, and over 60 years old. For the users who are in the same age stage, their properties are deemed to be same. The gender property is divided into 2 kinds; 1 means male, 0 female. The job property can be divided on the basis of their occupations, such as teacher, students, doctor, and so on. For the users who are of the same occupation, their properties are deemed to be same. We mark 1 for the same properties, 0 for different properties.

$u_p$, the weight factor of User Properties, is formulated as:

$$u_p = \frac{s_p}{c_p}$$ (2)

$c_p$ is the total records of User Properties, and $s_p$ is the number of how many same properties the users have.
2.3. Rating Frequency Weight Factor of User-Item Features

Traditional similarity is computed on the basis of users’ rating matrix, while the rating matrix has the problem of data sparsity. To compute similarity on the basis of building the matrix of user-item features based on users’ rating matrix can help reduce the data sparsity.

| Users | Item Features |
|-------|---------------|
|       | Attr_1 | Attr_2 | Attr_3 | ... | Attr_n |
| u_1   | 0      | 1      | 0      | ... | 1      |
| u_2   | 1      | 1      | 0      | ... | 1      |
| ...   | ...    | ...    | ...    | ... | ...    |
| u_n   | 0      | 1      | 1      | ... | 1      |

In Table 2, 1 means the item set of users’ rating has this feature, while 0 means it doesn’t have.

In view of different features existing in the same item, users’ rating frequency on item features can reflect their true preference when the item is decomposed into corresponding features. Users’ high rating frequency shows that they like the item which has the features, vice versa. Therefore, building the rating frequency matrix of user-item features helps us know users’ true preference.

| Users | Item Features |
|-------|---------------|
|       | Attr_1 | Attr_2 | Attr_3 | ... | Attr_n |
| u_1   | 0      | 23     | 34     | ... | 1      |
| u_2   | 12     | 9      | 0      | ... | 13     |
| ...   | ...    | ...    | ...    | ... | ...    |
| u_n   | 20     | 2      | 21     | ... | 16     |

On the basis of rating frequency matrix of user-item features, the rating frequency factor of User m on Feature i can be defined as:

\[ u_{rf} = 1 + \frac{p_i}{p_c} \]  

(3)

In this formula, \( u_{rf} \) is user’s rating frequency factor on item feature i; \( p_i \) is user’s rating frequency on feature i; \( p_c \) is user’s rating collection on item features.

2.4. Improved Time Attenuation Function

One of traditional recommended algorithm’s disadvantages is that it doesn’t take the changing of users’ interest over time into account, or users’ interest is periodical. Therefore, the stable phase of users’ interest based on traditional Ebbinghaus Forgetting Curve is added in this thesis. In the stable phase of users’ interest, the interest is deemed as constant, and it decreases over time beyond the stable phase. Experimental results show that the recommending effect is the best if the stable phase of users’ interest is 7 days, which means their interest remains constant within 7 days when they rate the item.
In this formula, $s_{u,1}$ is the first score the user gave; $t_1$ is the time the user first rated; $s_{u,l}$ is the final score the user gave; and $t_l$ is the time the user rated lastly; the duration of interest’s stale phase ($t_l - t_5$) is 7 days.

After combining Ebbinghaus Forgetting Curve and the time shaft above, the improved time attenuation function $f(t_j)$ can be defined as:

$$f(t_j) = \begin{cases} \frac{1}{t_{\text{max}} - t_j} & t_j \leq t_w \\ \frac{1}{t_{\text{max}} - t_j} & t_j > t_w \end{cases}$$  \hspace{1cm} (4)

If the time the user rate an item is within their interest’s stable phase, we regard its rating weight is 1; if the time the user rate an item is beyond their stable phase, its rating weight is $f(t_j)$. On the basis of time attenuation function, we can come up with users’ time weight formula on this item, which is:

$$a_{m,j} = \begin{cases} \frac{1}{t_{\text{max}} - t_j} & t_j \leq t_w \\ \frac{1}{t_{\text{max}} - t_j} \Sigma_{i=1}^{n} \frac{1}{t_{\text{max}} - t_i} & t_j > t_w \end{cases}$$  \hspace{1cm} (5)

In this thesis, timestamp is uniformly converted as the number of days, and the denominator is normalized, among which $a_{m,j}$ is User m’s time weight on Item j under time attenuation function; $t_{\text{max}} = t_l - t_w$, $t_l$ is the last time the user rated the item; $t_w$ is the stable phase days of users’ interest; $t_i, t_j$ is the time the user rate the item beyond the stable phase; $n$ is how many times the user rated the item beyond the stable phase.

$$p_{f(m, i)} = \frac{1}{2} \sum_{j=1}^{k} \frac{1}{t_{\text{max}} - t_j} * rate_{m,j} + \frac{1}{2n} \Sigma_{j=1}^{n} rate_{m,j}$$  \hspace{1cm} (6)

$K$ is how many times the user rated the item with Feature i beyond interest’s stable phase; $n$ is how many times the user rated item with Feature i within interest’s stable phase; $rate_{m,j}$ is the score the user rated Item j; $a_{m,j}$ is the time weight on item; $rate_{m,j}$ is the score the user rated the item.

3. Improved Similarity Formula

3.1. Similarity Formula of Recommended Algorithm Based on User Properties

Traditional adjusted cosine similarity formula is based on users’ rating on item, which neglects users’ different involvement in rating the item, and ignores the fact that User Properties can show their preference. Therefore, users’ property factor and interaction factor are brought into the traditional adjusted cosine similarity formula, thus putting forward a new recommended formula based on User Properties, which can be called as RABUP Algorithm.

The traditional adjusted cosine similarity formula is:
\( sim(m, n) = \frac{\sum_{i \in I_{mn}} (rate_{m,i} - \bar{rate}_m) \cdot (rate_{n,i} - \bar{rate}_n)}{\sqrt{\sum_{i \in I_{mn}} (rate_{m,i} - \bar{rate}_m)^2 \cdot \sum_{i \in I_{mn}} (rate_{n,i} - \bar{rate}_n)^2}} \)  

(7)

\( rate_{m,i} \) is the score the User m on Item i; \( rate_{n,i} \) is the score the User n on Item i; \( \bar{rate}_m \) and \( \bar{rate}_n \) are respectively the average scores of User m and User n rated on items; \( I_m \) and \( I_n \) are respectively the rating collections of User m and User n; \( I_{mn} \) is the common rating collection of User m and User n.

The improved similarity formula is:

\[ sim(m, n) = u_p + u_{if} \cdot \frac{\sum_{i \in I_{mn}}(rate_{m,i} - \bar{rate}_m) \cdot (rate_{n,i} - \bar{rate}_n)}{\sqrt{\sum_{i \in I_{mn}}(rate_{m,i} - \bar{rate}_m)^2 \cdot \sum_{i \in I_{mn}}(rate_{n,i} - \bar{rate}_n)^2}} \]  

(8)

\( u_p \) is the property weight factor between User m and User n; \( u_{if} \) is the interaction factor between User m and User n; \( rate_{m,i} \) is the score the User m rated on Item i; \( rate_{n,i} \) is the score the User n rated on Item i; \( \bar{rate}_m \) and \( \bar{rate}_n \) are respectively the average scores of User m and User n rated on items; \( I_m \) and \( I_n \) are respectively the rating collection of User m and User n; \( I_{mn} \) is the common rating collection of User m and User n.

### 3.2. Recommended Algorithm Similarity Formula Based on Item Features

Preference matrix of user-item features and users’ rating frequency matrix on item are obtained by decomposing user-item rating matrix, and preference matrix of user-item features which combines the time is obtained by fusing the time attenuation factor into user-item feature’s preference. Since the users’ rating frequency on item can show their true preference, we lead in rating frequency factor of item features when we compute the adjusted cosine similarity based on preference matrix of user-item features, and put forward the recommended algorithm based on item features, which is called RABIF Algorithm.

The improved similarity formula is:

\[ sim(m, n) = u_{rf} \cdot \frac{\sum_{i \in I_{mn}}(rate_{m,i} - \bar{rate}_m) \cdot (rate_{n,i} - \bar{rate}_n)}{\sqrt{\sum_{i \in I_{mn}}(rate_{m,i} - \bar{rate}_m)^2 \cdot \sum_{i \in I_{mn}}(rate_{n,i} - \bar{rate}_n)^2}} \]  

(9)

\( u_{rf} \) is the rating frequency weight factor between User m and User n; \( rate_{m,i} \) is the score the User m rated on Item i after combining time factor; \( rate_{n,i} \) is the score the User n rated on Item i after combining the time factor; \( \bar{rate}_m \) and \( \bar{rate}_n \) are respectively the average score the User m and User n on all item features after combining time factor; \( I_m \) and \( I_n \) are respectively rating collections of User m and User n on item features; \( I_{mn} \) is the rating collection of User m and User n on the common item features.

### 4. Mixed Recommended Algorithm Combining User Properties and Item Features

The two recommended algorithms RABUP and RABIF proposed in this thesis respectively take factors of user properties and item features into account. By linearly combining these two recommended algorithms to complement each other, a new mixed recommended algorithm arises, which a mixed is recommended algorithm combining user properties and item features, namely, UPIF Algorithm. And we use this algorithm and adjust mixed weight factor \( \alpha \) to measure users’ predicted rating.

Predicted rating formula of RABUP Algorithm:

\[ UP_{m,i} = \frac{\bar{rate}_m + \sum_{n \in NS_m} sim(m, n) \cdot (rate_{m,i} - \bar{rate}_m)}{\sum_{n \in NS_m} |sim(m, n)|} \]  

(10)
In this formula, $NS_m$ is the adjacent user set of User m; $\text{sim}(m,n)$ is the similarity of User m and User n; $\text{rate}_{m,l}$ is the rate of User m on Item i; $\text{rate}_m$ is the average score of User m on item set.

Predicted rating formula of RABIF Algorithm:

$$IF_{m,i} = \frac{\text{rate}_m + \sum_{n\in NT_m} \text{sim}(m,n) \ast (\text{rate}_{m,l} - \text{rate}_m)}{\sum_{n\in NT_m} |\text{sim}(m,n)|} \quad (11)$$

In this formula, $NT_m$ is the adjacent user set of User m; $\text{sim}(m,n)$ is the similarity of User m and User n; $\text{rate}_{m,l}$ is the rate of User m on Item i; $\text{rate}_m$ is the average score of User m on item set.

$$UP_{m,i} = \alpha \ast UA_{m,i} + (1 - \alpha) \ast PC_{m,i}$$

$$= \alpha \ast \left( \frac{\text{rate}_m + \sum_{n\in NT_m} \text{sim}(m,n) \ast (\text{rate}_{m,l} - \text{rate}_m)}{\sum_{n\in NT_m} |\text{sim}(m,n)|} \right) + (1 - \alpha) \ast \left( \text{rate}_m + \frac{\sum_{n\in NT_m} \text{sim}(m,n) \ast (\text{rate}_{m,l} - \text{rate}_m)}{\sum_{n\in NT_m} |\text{sim}(m,n)|} \right) \quad (12)$$

In this formula, $UP_{m,i}$ is the predicted rate of User m on Item i under mixed recommended algorithm; $NS_m$ is the adjacent user set of User m; $\text{sim}(m,n)$ is the similarity of User m and User n; $\alpha$ is the mixed weight factor whose value is between (0, 1); different predicted rating result is come out from adjusting different values of $\alpha$; it is the best to recommend when $\alpha = 0.75$ after conducting 100 experiments in $\alpha \in (0.01, 0.99)$.

Frame diagram of mixed recommended algorithm as shown below:

5. Experimental Results and Analysis

This experimental adopts MovieLens 100k dataset, which has collected 100,000 rating records of 1682 films from 943 users within 7 months from September 19, 1997 to April 22, 1998.

Operating system: win10
Programming language: JAVA
Evaluation indicator of Experiments: mean absolute error (MAE for short)
MAE formula is:

\[ MAE = \frac{\sum_{i \in T} |r_{ai} - \hat{r}_{ai}|}{|T|} \]  \hspace{1cm} (13)

In this formula, T is the total number of items in test set; \( r_{ai} \) is the actual score the User a rated the Item i; \( \hat{r}_{ai} \) is the predicted score the User a on Item i.

This experiment sorts the date set in descending order by time, and chooses top 20% as the test set, and the rest 80% as the training set. It respectively conducts algorithm performance tests on the recommended algorithm combining user properties and user interaction factor, on the recommended algorithm combining user rating frequency factor and improved time attenuation function, and on the mixed recommended algorithm combining user properties and item features.

5.1. The Results of Recommended Algorithm Based on User Properties
In order to verify the validity of RABUP Algorithm, this thesis realizes the accuracy of traditional adjusted cosine similarity under RABUP Algorithm. For convenience, adjusted cosine is abbreviated as A-Cos, and the improved similarity as RABUA. X-axis is the number of adjacent users K, which means the number of how many users have the similarity with the target user. Y-axis is the MAE; the smaller the value is, the effect of the algorithm is better.

![Figure 3. Comparison of MAE in RABUP Algorithm](image)

As shown in the chart, the values of MAE are both decrease as the numbers of adjacent users increase in RABUP Algorithm and A-Cos Algorithm. It is because as the number of adjacent users increases, the number of users who have the similarity with the target user increases, and then the accuracy of recommendation increases accordingly, but the MAE value decreases, among which the MAE value of RABUP Algorithm is much lower than the A-Cos Algorithm. When the number of adjacent users reaches over 80, the two recommended values of MAE in the chart remain the same basically. It is because as the number of adjacent users increases, the users who have the similarity with target user are saturated basically. When k=80, MAE in RABUP Algorithm in this thesis is 1.1% lower than A-Cos Algorithm.

5.2. The Result of Recommended Algorithm Based on Item Features
Based on improved time attenuation function, the thesis improves the traditional similarity, and puts forward a recommended algorithm combining user rating frequency factor and improved time attenuation function. In order to verify the validity of this algorithm, the similarity which only includes
the improved time attenuation function, the similarity which only includes user rating frequency factor, and the traditional adjusted cosine similarity are compared in the thesis, thus realizing the accuracy of recommendation in RABIF Algorithm proposed in the thesis. Among which, the similarity which includes improved time attenuation function is denoted as IT, the similarity which includes user rating frequency factor as IPFS, and the traditional adjusted cosine similarity as A-Cos.

![Figure 4. Comparison of MAE in RABIF Algorithm](image)

As shown in the chart, as the number of adjacent users $k$ increases, the values of MAE in four algorithms are in downturn, among which the MAE value of RABIF Algorithm is obviously lower than that of A-Cos and IT, and a bit lower than IPFS Algorithm. Besides, we can see from the chart that when $k>80$, the MAE values of four algorithms decreases slowly. It is because as the number of adjacent users increases, the users who have the similarity with target user is saturated basically. When $k=80$, MAE value of RABIF Algorithm in the thesis is 2% lower than the traditional A-Cos Algorithm.

5.3. The Result of Mixed Recommended Algorithm Combining User Properties and Item Features

In order to effectively verify the mixed recommended algorithm combining user properties and item features, the thesis compares the subalgorithms of RABIF and RABUP which combines mixed recommended algorithm.

![Figure 5. Comparison of MAE in Three Recommended Algorithms](image)

As shown in the chart, as the number of adjacent user’s increases, the MAE value of three algorithms is obviously in downturn, and the MAE value of UPIF Algorithm which is the mixed recommended algorithm proposed in the thesis is obviously lower than two subalgorithms. When $k>80$, as the user set
which is similar with target user becomes saturated, the MAE values of three algorithms decreases slowly. When k=80, the MAE value of UPIF Algorithm is 6.3% lower than that of RABIF, and 1.4% lower than that of RABUP.

6. Conclusion
The traditional collaborative filtering recommended algorithm only takes users’ rating on time into account, and neglects the potential indicators which can reflect user’s interest, such as user properties, item features, etc. The thesis leads in user properties and user interaction factor when computing traditional adjusted cosine similarity so as to improve the traditional similarity formula. Secondly, it improves the time attenuation function through time stable phase, and decomposes user-item rating matrix to get preference matrix of user-item features and users’ rating frequency on item features, and improves traditional similarity formula by combining users’ rating frequency on item features and time attenuation function. Lastly, it linearly mixes two improved similarity formulas to get the mixed recommended algorithm which combines user properties and item features. The experimental results show that this mixed recommended algorithm can obviously reduce the mean absolute error and increase the accuracy of recommendation.

Acknowledgments
This work was supported by Natural Science Foundation of Hubei Province of China (Grant No.2018CFB681).

References
[1] Guo G, Zhang J, Thalmann D. Merging trust in collaborative filtering to alleviate data sparsity and cold start [J]. Knowledge-Based Systems, 57 (2014), 57-68.
[2] Chunxiao X, Fengrong G, Sinan Z, et al. A Collaborative Filtering Recommendation Algorithm Incorporated with User Interest Change [J]. Journal of Computer Research and Development, 44 (2) (2007), 296-301.
[3] Wei, Jian, et al, Collaborative filtering and deep learning based recommendation system for cold start items, Expert Systems with Applications 69 (2017), 29-39.
[4] Herlocker, Jonathan L, et al, Evaluating collaborative filtering recommender systems, ACM Transactions on Information Systems 22 (1) (2004), 5-53.
[5] Yang, Haiqin, et al, Boosting Response Aware Model-Based Collaborative Filtering, IEEE Transactions on Knowledge and Data Engineering 27 (8) (2015), 2064-2077.
[6] Shu, Jiangbo, et al, A content-based recommendation algorithm for learning resources, Multimedia Systems 1 (2017), 1-11.
[7] Yang, Shengxiang, et al, A Grid-Based Evolutionary Algorithm for Many-Objective Optimization, IEEE Transactions on Evolutionary Computation 17 (5) (2013), 721-736.
[8] Khalid, Astra, et al, Scalable and practical One-Pass clustering algorithm for recommender system, Intelligent Data Analysis 21 (2) (2017), 279-310.
[9] Ying-Nan, Liu, et al, Recommendation Algorithm Based on Trust in Social Network, Journal of Chinese Computer Systems 556-562 (2015), 5790-5793.
[10] Yuyu, Yin, et al, Network Location-Aware Service Recommendation with Random Walk in Cyber-Physical Systems, Sensors 17 (9) (2017), 2059.
[11] Zhang, Libo, et al, A Recommendation Model Based on Deep Neural Network, IEEE Access PP.99 (2018), 1-1.
[12] Wei, Chen, R. Khoury, and S. Fong, Web 2.0 Recommendation service by multi-collaborative filtering trust network algorithm, Information Systems Frontiers 15 (4) (2013), 533-551.