IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Mathematical Modelling of Engineering Problems
Vol. 6, No. 3, September, 2019, pp. 363-368
Journal homepage: http://iieta.org/journals/mmep

A Personalized Collaborative Filtering Recommendation Algorithm Based on Linear Regression

Chuping Xiong, Hua Sun*, Ding Pan, Yao Li

School of Software, Xinjiang University, Urumqi 830000, China

Corresponding Author Email: xj_sh@163.com

https://doi.org/10.18280/mmep.060307

Received: 4 May 2019
Accepted: 15 July 2019

Keywords:
Tag; linear regression; collaborative filtering; recommender system (RS)

ABSTRACT

This paper attempts to solve the problems with linear regression-based collaborative filtering recommendation algorithm, namely, the difficulty in extracting eigenvalues, the low accuracy and the poor interpretability. For this purpose, the tag weights were introduced as eigenvalues and the prediction accuracy was improved by the principle of collaborative filtering recommendation algorithm, creating a personalized collaborative filtering recommendation algorithm based on linear regression (PCFLR). Firstly, the tag weights for users were computed by term frequency–inverse document frequency (TF-IDF), and taken as the eigenvalues of the linear regression model. Then, the linear regression model was constructed based on the users' historical scores. After that, the cost function was set up by the least squares method, and regularized to prevent over-fitting. Next, the optimal value of the cost function was computed by gradient descent method, yielding the tag weights for items. On this basis, the predicted scores of all unrated items were obtained considering the linear relationship between the tag weights for users and those for items. The mean absolute error (MAE) between the predicted and actual scores was computed, and used to adjust the predicted scores into the final results. In addition, the set of recommendable items for the target user was produced based on the scores rated by all neighboring users, and coupled with the linearly regressed scores to make recommendations to the target user. The experimental results show that the PCFLR outperformed the traditional recommendation algorithms in accuracy and interpretability.

1. INTRODUCTION

In the Internet era, the user demands are increasingly diverse with the information boom. The massive amount of data makes it difficult for users to acquire the specific information that satisfies their needs. The recommender system (RS) [1, 2] is one of the most promising methods to filter out redundant information. The RS searches for the items that might interest the users based on the historical behaviors of users or the similarity between items [3]. As the core of the RS, the recommendation algorithm calculates the recommendation level of each item, revealing the most interesting items for each user.

The traditional recommendation algorithms are based on content, collaborative filtering or both. Among them, collaborative filtering recommendation algorithm has been widely applied, because it is simple, effective and easy to implement [4~6]. There are two types of collaborative filtering recommendation algorithms: those based on user or item, and those based on model. The former recommends items in the light of the neighbor set of the user or item, and the latter makes recommendations by setting up algorithm models. Despite its popularity, collaborative filtering recommendation algorithm also faces some problems in actual application [7], such as data sparsity and cold start. Data sparsity refers to the small ratio (<1%) of user evaluations to the huge number of items [8]. The extremely sparse data severely suppresses the recommendation effect. The cold start means it is difficult to make recommendations for new users or items, due to the lack of scores. Even if some items are recommended, no reason is given for making that recommendation, resulting in a poor user experience.

The proliferation of the social media has brought plentiful of user generated contents, which provides the RS with more diverse resources. The main features of these contents are summed up by users as tags, based on their understanding of such information resources [9]. Therefore, many scholars have introduced tags into the RS to solve the problems with the recommendation algorithm. For example, some scholars integrate tags into the collaborative filtering recommendation algorithm, and improves the algorithm performance based on the similarity between users, items and tags [10~13]. Wen et al. and Li et al. [14, 15] proposes a recommendation algorithm based on tag and topic, and explores the potential semantic topics in the RS to realize personalized recommendation. If the users' eigenvalues and scores are known, the collaborative filtering recommendation algorithm based on linear regression can set up a linear relationship to compute the eigenvalue of each item and predict the score of unrated items. But the linear regression-based algorithm has difficulty in extracting user features.

In the light of the above, this paper puts forward a personalized collaborative filtering recommendation algorithm based on linear regression (PCFLR). The tag weight was introduced as a user feature in the linear regression model, aiming to solve the difficulty in feature extraction. The scores of all unrated items were predicted by the linear regression model. To further reduce the prediction error, the mean absolute error (MAE) between the predicted score and actual score were computed, and used to adjust the predicted score to the final value. Next, the set of recommendable items was generated based on the scores rated by neighboring users, all the items in the set were ranked by the predicted score, and the top items were recommended to the target user. Compared with the traditional methods, the PCFLR can reduce the dimensionality of the matrices involved in the operations and overcome the matrix sparsity problem. Besides, the recommendations are interpretable, because they are generated based on tag weights to users. The experimental results show that the PCFLR clearly improved the accuracy of predicted scores from the level of traditional algorithms.

2. ALGORITHM DESIGN

With the linear regression model as the core, the PCFLR mainly identifies the linear relationship between the features of user preference and those of item attributes, and then predicts the item scores rated by users. Considering the difficulty to extract user and item features, the PCFLR treats the tag types of the items as the shared features of users and items [16], and derives the weights of user features by term frequency–inverse document frequency (TF-IDF), according to the users' rating of the items and the tag distribution of the items. Meanwhile, the weights of item features were obtained in two steps: setting up the cost function by least squares method, and finding the optimal solution of the cost function using gradient descent algorithm. On this basis, the item scores were predicted based on the weights of user features and item features. After that, the MAE between the predicted scores and actual scores was calculated, and used to further adjust the predicted scores to the final value. Next, the set of recommendable items was generated based on the scores rated by neighboring users, further reducing the prediction error and enhancing the recommendation accuracy to the target user.

2.1 TF-IDF weight calculation

The PCFLR assigns weights to tags by the TF-IDF method [17]. According to the TF-IDF principle, the ability of a tag to distinguish the document content is positively correlated with its frequency in the document and negatively with the number of documents containing the tag in the document set [18]. Here, the users' historical scores are combined with item tags into the score documents. Suppose user u1 has rated three items i1, i2 and i3, which contain the tags listed in Table 1. Then, the score documents can be expressed as du1= {a, c, a, b, b, c, d}.

Meanwhile, each combination of the tags in all items was considered as an item document di (Table 1). Then, three item documents can be established: di1= {a, c}, di2= {a, b} and di3= {b, c, d}. Both documents du and di describe the item tags, and were thus called tag documents d. The collection of all score documents and item documents is denoted as the set of documents D.

Table 1. The list of item tags

Name of item	The tag types in the items			
i_1	a		c	
i_2	a	b		
i_3		b	c	d

The tag frequency TF (*t*, *d*) was defined as the frequency of tag *t* appearing in score document *d*. The higher the frequency, the stronger the correlation between tag *t* and user *u*, and the greater its weight to this user. It is assumed that the score documents of user u_1 contain a total of *n* tags, in which tag *t* appears *m* times. Then, the frequency TF (*t*, *d*) of tag *t* in score document du_1 can be defined as:

$$TF(t,d) = \frac{m}{n} \tag{1}$$

In general, a tag frequently appearing in the score documents of a user is of great interest to that user. However, this might not be the case if the tag appears frequently in all documents. Suppose there are ten score documents, in which eight contain tag A and three contain tag B. If both tags appear three times in the score documents of user C, then it is wrong to conclude that user C has the same interest in tag A and tag B. After all, there are more tag as than tag Bs in all documents. To solve the problem, the inverse document frequency IDF (*t*, *d*) was introduced to describe the frequency of tag *t* in all documents *d*. The higher the IDF, the smaller the weight of the tag. Let $|D|$ be the total number of documents and $|d \in D: t \in d|$ be the total number of documents *d* containing tag *t* within the set of documents $|D|$. Then, the IDF (*t*, *d*) can be expressed as:

$$IDF(t,d) = \lg \frac{|D|}{|d \in D : t \in d|} \tag{2}$$

The following TF-IDF formula was adopted for the PCFLR:

$$TF - IDF(t,d) = TF(t,d) * IDF(t,d) \tag{3}$$

The TF-IDF value of each tag to each user was computed, forming the tag weights ***Tu*** to all users.

2.2 Construction of linear regression model

Linear regression [19] analyzes the combinational relationships between given features and actual values, and then fits the actual values through linear combinations. Let ***U***= {u_1, u_2, u_3, ..., u_n} be the set of users, ***I***={i_1, i_2, i_3, ..., i_m} be the set of items, and $r_{m,n}$ be the score of i_m rated by user u_n. Then, the matrix ***R*** of item scores rated by the users can be described as:

$$\mathbf{R} = \begin{bmatrix} r_{1,1} & \cdots & r_{1,n} \\ \vdots & \ddots & \vdots \\ r_{m,1} & \cdots & r_{m,n} \end{bmatrix}$$

The tag weights ***Tu***= (tu_1, tu_2, ..., tu_k) (k is the number of tags) of user *u* can be obtained by formula (3). Let ***Ti*** = (ti_1, ti_2, ..., ti_k) be the tag weights of item *i*. Then, the score of i_m rated by user u_n can be approximated by:

$$(\mathbf{Tu}^{(n)})^T * \mathbf{Ti}^{(m)} = r^{(m,n)}{}' \tag{4}$$

For example, $r^{(1,1)}{}' = (\mathbf{Tu}^{(1)})^T * \mathbf{Ti}^{(1)}$ and $r^{(1,2)}{}' = (\mathbf{Tu}^{(2)})^T * \mathbf{Ti}^{(1)}$ are the approximate scores of item i_1 rated by u_1 and u_2, respectively. The equations of the two scores have solutions, if the number of times h the item is rated is greater than the number of tags k of the item.

2.3 Construction of cost function

Based on the all known tag weights ***Tu*** and the actual score matrix ***R***, the cost function J can be established by the least squares method according to formula (4) as:

$$\mathrm{J} = \min\left(\frac{1}{2}\sum_{\mathrm{r}^{(m,n)}\neq 0}((\mathbf{Tu}^{(n)})^T * \mathbf{Ti}^{(m)} - r^{(m,n)})^2 + \frac{\lambda}{2}\sum_{k=1}^{n}(\mathbf{Ti}_k{}^{(m)})^2\right) \quad (5)$$

To minimize the value of the cost function J, the approximate score $r^{(m,\ n)}$' of formula (4) should approach the actual score $r^{(m,\ n)}$ as much as possible. The latter term on the right side of formula (5) is regularization, i.e. adding a penalty term to the cost function. To minimize the value of the cost function, the penalty term must be very small. The value of parameter λ should be determined properly. If the λ value is relatively large, the final value of parameter ***Ti*** will be very small, and the resulting curve will be smooth. In this case, there will be no over-fitting phenomenon. However, if the λ value is too large, the value of parameter ***Ti*** will approach zero, leading to over-fitting problem. The λ value is subsequently selected according to experimental data.

2.4 Gradient descent algorithm

To minimize the value of the cost function J, the gradient descent algorithm was adopted to find the partial derivative of the function to parameter ***Ti***, and to update the parameter value:

$$\mathbf{Ti}_k = \mathbf{Ti}_k - \alpha\frac{\partial \mathrm{J}}{\partial \mathbf{Ti}_k} \quad (6)$$

where k is the number of item tags. The above formula was executed iteratively to update all parameter values until the value of the cost function J no longer decreased. The value of α determines the rate of gradient descent. The larger the value of α, the faster the rate and the more efficient the computation. But an excessively large α will impede the convergence of the gradient descent algorithm.

2.5 Adjustment of predicted results

The predicted scores were further adjusted based on the MAE between them and the actual scores. Table 2 shows the actual scores rated by user A.

Table 2. The actual scores rated by user A

Items	Actual scores	Predicted scores
i_1	3	3.9573
i_2	5	4.1247
i_3	?	3.5
i_4	?	4

As shown in Table 2, user A has rated items i_1 and i_2, but not i_3 or i_4. The predicted scores of i_3 and i_4 were computed as 3.5 and 4, respectively, by the linear model. Based on user A's historical scores, the predicted score of i_3 was adjusted as $3.5+\frac{(3\text{-}3.9573)+(5\text{-}4.1247)}{2}=3.459$. That of i_4 was adjusted in a similar manner. The adjusted results are the final predicted scores.

2.6 Selection of neighboring users

The PCFLR uses cosine similarity to describe how similar two users are. Cosine similarity measures the likeliness between two vectors by the cosine of the angle between them, and distinguishes between the two vectors mainly by direction. The tag weights obtained by TF-IDF method provide the preference ratios of all tags. For example, the weights of tags A, B and C to users 1, 2 and 3 are shown in Table 3 below.

Table 3. The tag weights of three users

Tags	A	B	C
User 1	1	2	3
User 2	3	4	5
User 3	3	2	3

As shown in Table 3, users 1 and 3 have the closest tag weights, but users 1 and 2 have the greatest cosine similarity. In the PCFLR, users 1 and 2 are considered as having the most similar interests. The similarity between two users was measured by the cosine of the angle between their vectors. The similarity between users u and v can be illustrated as:

$$\cos(u,v) = \frac{\sum_{h=1}^{w}\mathbf{Tu}_h\mathbf{Tv}_h}{\sqrt{\sum_{h=1}^{w}\mathbf{Tu}_h}\sqrt{\sum_{h=1}^{w}\mathbf{Tv}_h}} \quad (7)$$

where Tu and Tv are the tag weights of users u and v, respectively; w is the dimensionality of each vector.

3. ALGORITHM DESCRIPTION

To construct the PCFLR, the set of tag documents was established based on the users' historical scores and item tags. Then, the tag weights to all users were computed by the TF-IDF method. Next, the linear regression model and cost function were set up, and the tag weights of items were computed by gradient descent algorithm. After that, the scores of the items not yet rated by the target user were predicted according to the tag weights to the user and the items. Finally, the set of recommended items was selected for the target user, in the light of the set of scores rated by the neighboring users. The workflow of the PCFLR is explained in Figure 1 below.

The specific steps of the PCFLR are as follows:

Algorithm: The PCFLR

Inputs: User-item datasets, the number of top-N items

Outputs: The set of top-N recommended items for the target user

Step 1. Setting up the set of tag documents

The users' historical scores and item information were sorted to obtain the user-item-tag datasets. The item tags were sorted based on user ID, creating the tag documents for users. The tags in each item were considered as the tag documents for that item. The tag documents for users and those for items were combined into the set of documents. Then, the following information were counted: the types of tags in the document set, the total number of different types of tags, the total number of tags in each user document, and the number of each type of tags in each user document.

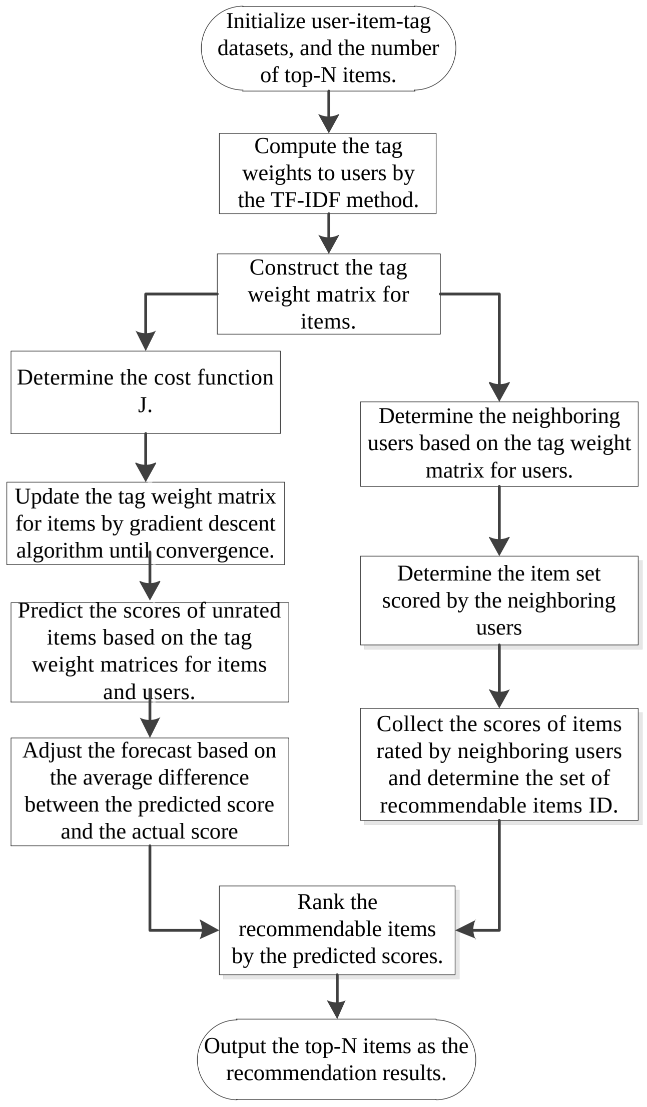
Initialize user-item-tag datasets, and the number of top-N items.
Compute the tag weights to users by the TF-IDF method.
Construct the tag weight matrix for items.
Determine the cost function J.
Update the tag weight matrix for items by gradient descent algorithm until convergence.
Predict the scores of unrated items based on the tag weight matrices for items and users.
Adjust the forecast based on the average difference between the predicted score and the actual score
Determine the neighboring users based on the tag weight matrix for users.
Determine the item set scored by the neighboring users
Collect the scores of items rated by neighboring users and determine the set of recommendable items ID.
Rank the recommendable items by the predicted scores.
Output the top-N items as the recommendation results.

Figure 1. The workflow of the PCFLR

Step 2. Computing the tag weights for users

Based on the information acquired in Step 1, the TF and IDF values were computed by formulas (1) and (2), respectively. Then, the tag weights for each user were determined, forming a tag weight matrix for users.

Step 3. Constructing the tag weight matrix for items

The tag weight matrix for items was initialized randomly according to the number and types of known tags.

Step 4. Determining the cost function for linear regression

The predicted scores of terms were derived from the linear relationship between the tag weights for users and those for items. Then, the cost function was established by formula (5), considering the MAE between the predicted and actual scores.

Step 5. Gradient descent

The value of the cost function was minimized by gradient descent method. The parameter ***Ti*** in the cost function was updated iteratively by formula (6) until the function value no longer decreased. The parameter value at the termination was recorded as the tag weight for the corresponding item.

Step 6. Generating and adjusting predicted scores

The MAE between predicted and actual scores of the training set was computed, and then the predicted scores of the test set were adjusted to obtain the final results.

Step 7. Determining the item set of neighboring users

The neighboring users of the target user were identified based on cosine similarity, which was computed based on the tag weight matrix for users. The k users with the highest cosine similarities were selected as neighboring users. Then, the item IDs of all k users were obtained.

Step 8. Outputting the recommended items

Based on the item IDs of the neighboring users, the predicted scores of these items were obtained from the linear model and ranked in descending order. Next, the top-N items in the ranking were recommended to the target user.

4. EXPERIMENTS AND RESULTS ANALYSIS

4.1 Experimental datasets

To verify its performance, the PCFLR was tested on MovieLens, which was created by GroupLens Research at the University of Minnesota, in order to gather research data on personalized recommendations. The MovieLens contains millions of ratings for movies and tags on movie genres. In this paper, the MovieLens 100K dataset is selected for the experimental verification. This dataset involves the scores of 1,682 movies rated by 943 users. Each user has rated 20 movies at the least. The scores were rated against a five-point scale {1, 2, 3, 4, 5}. The higher the score, the more interested the user is in the movie. There are 19 tags in the dataset, including unknown, action, adventure, animals, children, comedy, crime, science fiction, etc. The sparsity of the user-item scoring matrix is 1-(100000/(943*1682))=93.69%.

4.2 Evaluation index

The MAE was taken as the evaluation index for the PCFLR. This index measures the prediction accuracy by the error between the predicted and actual scores. The smaller the MAE, the better the recommendation. Let $\{p_1, p_2, ..., p_n\}$ be the set of predicted scores and $\{r_1, r_2, ..., r_n\}$ be the set of actual scores. Then, the MAE can be defined as:

$$MAE = \frac{\sum_{i=1}^{n} |p_i - r_i|}{n} \tag{8}$$

4.3 Determination of λ value

The experimental data were designed as in Table 3 to determine the λ value in the cost function.

Table 3. The division of the experimental dataset

The Ratio of the Testing Set	The Ratio of the Validation Set	The Number of Neighboring Users
10%	80%	10

The experimental results are shown in Figure 2. It can be seen that, with the increase of the λ value, the MAE firstly exhibited a gradual decline, reached the ideal value at λ=8, and then gradually increased. Hence, the λ value was selected as eight in the subsequent experiment.

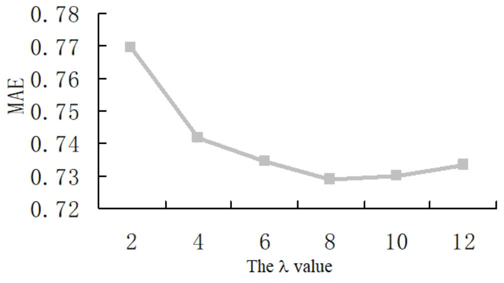

Figure 2. The relationship between the MAE and the λ value

4.4 The impacts of the number of neighboring users

The experimental data were designed as in Table 4 to determine how the number of neighboring users influences the recommendation results.

Table 4. The setting of the experimental data

The Ratio of the Testing Set	The Ratio of the Validation Set	The Number of Neighboring Users	The λ Value
10%	80%	10-60	8

The experimental results in Figure 3 show that the MAE increased with the number of neighboring users. This is because the neighboring users were selected based on the similarity. If there are many neighboring users, it is very likely for a user with low similarity to the target user to be identified as a neighbor.

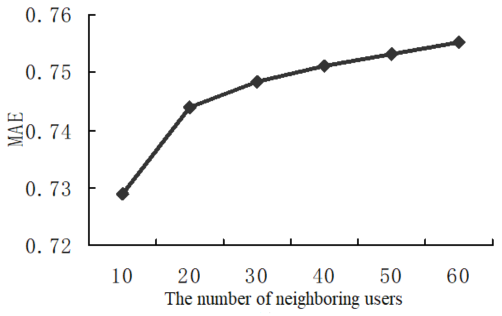

Figure 3. The relationship between the MAE and the number of neighboring users

4.5 Comparative experiments

The PCFLR, the collaborative filtering recommendation algorithm based on linear regression (CFLR), the user-based collaborative filtering (UserCF) algorithm, and the collaborative filtering based on item clustering and tag topics (ICFT) were tested with the same experimental dataset, parameter setting, and experimental environment. The recommendation results of the four algorithms are shown in Figure 5. It is clear that the PCFLR achieved the smallest error and the best prediction accuracy among the four algorithms. The results verify the effectiveness of our algorithm.

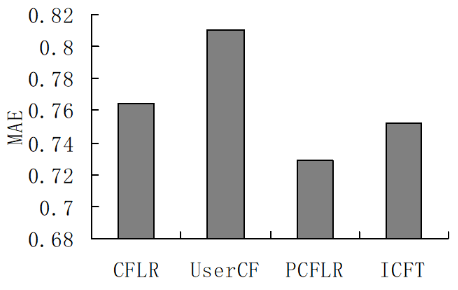

Figure 4. The comparison of experimental results

5.CONCLUSIONS

This paper improves the linear regression-based collaborative filtering recommendation algorithm, aiming to solve the following problems: the difficulty in extracting eigenvalues, the low accuracy and the poor interpretability. The proposed algorithm, denoted as the PCFLR, was proved through experiments as effective in making accurate recommendations. In addition, the recommendation results can be interpreted based on the tag weights to the users. The future research will optimize the tag values based on the scores rated by users and further enhance the recommendation accuracy.

ACKNOWLEDGEMENT

This paper is supported by the Natural Science Foundation of Xinjiang Uygur Autonomous Region of China (Grant No.: 2015211C263).

REFERENCES

[1] Li, Y. (2018). Design and implementation of intelligent travel recommendation system based on internet of things. Ingénierie des Systèmes d'Information, 23(5): 159-173. https://doi.org/10.3166/ISI.23.5.159-173
[2] Adomavicius, G., Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering, 17(6): 734-749. https://doi.org/10.1109/TKDE.2005.99
[3] Huang, L.W., Jiang, B.T., Lv, S.Y., Liu, Y.B., Li, D.Y. (2018). Survey on deep learning based recommender systems. Chinese Journal of Computers, 41(7): 1619-1647. https://doi.org/10.11897/SP.J.1016.2018.01619
[4] Lu, J., Wu, D., Mao, M., Wang, W., Zhang, G. (2015). Recommender system application developments. Decision Support Systems, 74(C): 12-32. https://doi.org/10.1016/j.dss.2015.03.008
[5] Bobadilla, J., Ortega, F., Hernando, A. (2013). Recommender systems survey. Knowledge-based Systems, 46(1): 109-132. https://doi.org/10.1016/j.knosys.2013.03.012
[6] Schafer, J.B., Frankowski, D., Herlocker, J., Sen, S. (2004). Collaborative filtering recommender systems. Acm Transactions on Information Systems, 22(1): 5-53.
[7] Su, X.Y., Khoshgoftaar, T.M. (2009). A survey of

collaborative filtering techniques. Advances in Artificial Intelligence, 1-19. https://doi.org/10.1155/2009/421425

[8] Yang, X.Y., Yu, J., Ibrahim, T., Liao, B., Ying, C.T. (2015). Collaborative filtering recommendation model based on trust model filling. Computer Engineering, (5): 6-13.

[9] Ding, Y. Jacob, E.K., Zhang, Z.X., Foo, S., Yan, E., George, N.L., Guo, L.J. (2014). Perspectives on social tagging. Journal of the Association for Information Science & Technology, 60(12): 2388-2401. https://doi.org/10.1002/asi.21190

[10] Deng, X.Y., Wang, C. (2018). A hybrid collaborative filtering model with context and folksonomy for social recommendation. Ingénierie des Systèmes d'Information, 23(5): 139-157. https://doi.org/10.3166/ISI.23.5.139-157

[11] Zhang, J.L., Huang, M.X., Zhang, Y., Wu, Q.Z. (2018). Collaborative filtering recommendation algorithm based on tag optimization. Application Research of Computers, 35(10): 2916-2919. https://doi.org/10.1109/ISCID.2015.91

[12] He, M., Yao, K.S., Yang, P., Zhang, J.L. (2018). Collaborative filtering personalized recommendation based on similarity of tag information feature. Computer Science, 45(S1): 415-422.

[13] Kong, X.X., Su, B.C., Wang, H.Z., Gao, H., Li, J.Z. (2017). Research on the modeling and related algorithms of label-weight rating based recommendation system. Chinese Journal of Computers, 40(6): 1440-1452. https://doi.org/10.11897/SP.J.1016.2017.01440

[14] Wen, J.H., Yuan, P.L., Zeng, J., Wang, X.B., Zhou, W. (2017). Research on collaborative filtering recommendation algorithm based on topic of tags. Computer Engineering, 43(1): 247-252, 258. https://doi.org/10.3969/j.issn.1000-3428.2017.01.043

[15] Li, H.Y., Fu, Y.Q. (2018). Collaborative filtering recommendation algorithm based on tag clustering and item topic. Computer Science, 45(4): 247-251. https://doi.org/10.11896/j.issn.1002-137X.2018.04.041

[16] Chen, J., Liu, X.J., Li, B., Zhang, W. (2017). Personalized microblog recommendation based on dynamic interests and social networking of users. Acta Electronica Sinica, 45(4): 898-905. https://doi.org/10.3969/j.issn.0372-2112

[17] Zhou, Y., Dai, M.H. (2013). News recommendation technology combining semantic analysis with TF-IDF method. Computer Science, 40(s2): 267-269. https://doi.org/10.3969/j.issn.1002-137X.2013.z2.066

[18] Shi, C.Y., Xu, C.J., Yang, X.J. (2009). Study of TFIDF algorithm. Journal of Computer Applications, 29(b06): 167-170, 180.

[19] Pang, H.L., Zhao, H., Li, W., Ma, Y., Cui, Y. (2019). Linear regression recommendation algorithm with collaborative filtering. Application Research of Computers, 36(5): 1302-1304, 1310. https://doi.org/10.19734/j.issn.1001-3695.2017.11.0732