Collaborative filtering algorithm based on multi-factors

Chonghuan Xu*

Business Administration College,
Zhejiang Gongshang University,
Hangzhou City, China
Email: talentxch@163.com
*Corresponding author

Jie Wang

Business Administration College,
Zhejiang Gongshang University,
Hangzhou City, China
and
School of Shangmao,
Zhejiang Institute of Economics and Trade College,
Hangzhou City, 310018, China
Email: andersonwang111@foxmail.com

Jiangjun Yuan

School of Shangmaolvyou,
Hangzhou Vocational and Technical College,
Hangzhou, 310018, China
Email: yjjwell@gmail.com

Abstract: Recommender systems are widely used to provide e-commerce users appropriate items and have emerged in response to the problem of information overload. Collaborative filtering (CF) is one of the most successful recommender methods which recommend items to a given user based on the opinions of the similar users. However, the existing CF methods lack the consideration of factors such as time and geo-location. In this paper, we take into account many influencing factors including time and geo-location in the process of similarity computation. The simulation results on two real-world data sets show that our algorithm achieves superior performance to existing methods.

Keywords: recommender systems; collaborative filtering; CF; multi-factors.

Reference to this paper should be made as follows: Xu, C., Wang, J. and Yuan, J. (2020) 'Collaborative filtering algorithm based on multi-factors', *Int. J. Computing Science and Mathematics*, Vol. 11, No. 1, pp.29–39.

Biographical notes: Chonghuan Xu received his BS and PhD in the Zhejiang Gongshang University, Hangzhou in 2011 and 2016, respectively. He is currently an Associate Professor in the College of Business Administration, Zhejiang Gongshang University. His research interests include data mining, electronic commerce. He has published more than 30 papers which are indexed by SCI and EI.

Jie Wang is a doctoral candidate of Management, Lecturer. He achieved his Master's degree from the Zhejiang Gongshang University in 2015 and now he is studying PhD program of Management at the Zhejiang Gongshang University. He also works in Zhejiang Technical Institute of Economics. His research interest is Cross-border e-commerce.

Jiangjun Yuan is a Professor at the Hangzhou Vocational and Technical College, China. He achieved his BS in e-business at the Zhejiang University, China in 2002 and Master's of Enterprise Management at the Zhejiang Gongshang University, China in 2009. He also has held a post as the Assistant General Manager at the Zhejiang Wuchang e-business company since 2014. His current research focuses on electronic commerce

1 Introduction

Recommendation algorithms have been widely applied to deal with information overload problems in e-commerce sites. They utilise recorded user activities and feasible profiles to reveal user preferences and tastes. In recent years, recommender systems become extremely common in a variety of applications. The most popular ones are among movies, music, news, books, research articles, search queries, social tags. Generally, the performance of a recommender system is determined by its recommendation algorithms. So designing an excellent algorithm is crucial to enhance the performance of a recommender system.

Collaborative filtering (CF) is a class of information filtering technique which can predict what users will like according to their similarity to other users based on collecting and analysing a large amount of information on users' behaviours, activities or preferences. As we know, CF suffers from two major problems: data sparsity and cold-start. The data sparsity problem arises from the small percentage of available ratings from users to items. The cold-start problem refers to items (or users) without sufficient previous rating history.

In order to alleviate sparsity and cold-start problems of CF method and increase accuracy and diversity of recommendation results, we propose an effective CF recommendation approach. In this approach, we consider more factors such as user location information obtained by personal computers or mobile devices, user preference, user shopping time and so on. The simulation results on two real-world data sets show that our algorithm achieves superior performance to existing approaches.

The main contribution is that we consider many influencing factors in constructing similarity model. It can alleviate the sparsity and cold-start problems and also enhance the accuracy and diversity.

The rest of this paper is organised as follows. In Section 2, a review of related work is given. In Sections 3, we describe our recommendation method. Section 4 provides experiment results and analysis of the proposed method on two real-world data sets. Finally, we draw conclusions in Section 5.

2 Related work

In this section, we reviewed some good CF methods. Liu and Meng (2014) have demonstrated that mobile users who live nearby tender to rate similar score for a certain object. The conclusion can be drawn that mobile users nearby share similar preference to a given topic. Kumar and Fan (2015) proposed a hybrid method based on item based CF trying to achieve a more personalised product recommendation for a user while addressing some of these challenges. Zhou and Wu (2016) proposed a rating LDA (RLDA) model for CF by adding rating information to the latent Dirichlet allocation (LDA). User behaviour was not independent; it followed the trend of others. Kim et al. (2016) proposed an improvement of an existing preference prediction algorithm to increase the accuracy of recommendation systems. In a recommendation system, prediction of items preferred by users was based on their ratings. Bellogín and Sánchez (2017) proposed a technique to compare users – also extendable to items –, working with them as sequences instead of vectors, hence enabling a new perspective to analyse the user behaviour by finding other users who had similar sequential patterns instead of focusing only on similar ratings in the items. Wang et al. (2017) proposed a new user similarity scheme by a hybrid method, which considered the influence of all possible rated items, the non-linear relationship between variables, the asymmetry between users and the rating preference of users. Aghdam et al. (2017) presented a non-negative matrix factorisation method to alleviate sparseness and scalability problems via decomposing rating matrix into user matrix and item matrix. Kumar et al. (2017) proposed an alternative and new MMMF scheme for discrete-valued rating matrix. Their work drew motivation of recent advent of proximal support vector machines (SVMs). Shams and Haratizadeh (2017) proposed a new algorithm, called TasteMiner that efficiently learns partial users taste to restrict the neighbourhood space. They framed TasteMiner as a method for neighbourhood CF and showed its effectiveness compared to previous algorithms. Ren and Wang (2018) proposed a SVM-based CF service recommendation approach, namely SVMCF4SR. For a user, SVM could acquire a separating hyper-plane from the historical rating data, which could filter out the services that might not be preferred by the user. Nilashi et al. (2018) developed a new hybrid recommendation method based on CF approaches. In this research they solved two main drawbacks of recommender systems, sparsity and scalability, using dimensionality reduction and ontology techniques.

The target of our work is to design a recommendation method which can give users high accuracy and certain diversity recommendation results. In addition, our method should alleviate the sparsity and cold-start problems as much as possible. At the end, the experimental results show that our method is better than many other recommendation approaches.

3 CF method based on multi-factors

In the proposed CF model, we adopt many influencing factors. At the first, we use modified cosine similarity method considering user location information, user preference and user shopping time to computer similarity among users. Then, we generate the recommendation lists and recommend them to the target users, in which location information obtained by users' PC or smart mobile devices.

The proposed method contains three steps:

1 Data pre-processing

Data pre-processing is an important step in the data mining process. Generally, data-gathering methods are often lacking attribute value, certain attributes of interest or containing only aggregate data, errors or outliers. Furthermore, the primary data may come from different domains and it must be pre-processed by data fusion operation. So, we tackle with the primary data to fulfil the requirement of recommendation method before similarity computing. For instance, we convert and normalise the users' location information in order that our proposed recommendation method can compute these data directly.

2 Similarity computing

We propose a novel user similarity computational method based on resource diffusion to tackle with the deficiencies of traditional method. Firstly, we assume that there are m users and n objects in a recommendation model. Each user has selected some objects, also, each object has been selected by some users. In a bipartite network, let $U = \{u_1, u_2, \dots, u_m\}$ denote users-set and $O = \{o_1, o_2, \dots, o_n\}$ denote objects-set, then the recommendation model can be fully described by an adjacency matrix $A = \{a_{ij}\}$, where $a_{ij} = 1$ when object j is selected by user i; otherwise, $a_{ij} = 0$. The user similarity computational method based on resource diffusion will be described in next section.

3 Generate recommendation

After computing the similarity between users, we compute the comprehensive preference degree of each object, then generate recommendation lists in descending order. Finally, we recommend the top L objects within the presented recommendation lists to the target users. On the basis of the various kinds of users purchasing behaviours, we can dynamically adjust the number of L to suit for them.

3.1 Cosine similarity computation based on resource diffusion

Cosine similarity is a measure of similarity between two vectors of an inner product space that measures the cosine of the angle between them, often used to compare users or products in recommending. In this paper, a novel user similarity computational method based on resource diffusion is proposed which tackles with the deficiencies of traditional method. Traditional CF method usually adopts the standard cosine similarity or Pearson correlation to compute the similarity between two users. According to the definitions mentioned above, the number of common objects shared by them can be expressed as:

$$C_{il} = \sum_{j=1}^{n} a_{ij} a_{lj} \tag{1}$$

Let S_{il} denote the similarity between user i and user l, $k(u_i)$ is the degree of the user i that means how many objects are collected by this user. So we can formulate the traditional cosine similarity as:

$$S_{ij} = \frac{C_{ij}}{\sqrt{k(u_i)k(u_l)}} = \frac{\sum_{j=1}^{n} a_{ij} a_{lj}}{\sqrt{k(u_i)k(u_l)}} \tag{2}$$

Obviously, this formula exists some problems, the biggest one is that it does not consider the influence of each object. It means that different objects in the bipartite network have same contribution to the similarity.

3.2 Improved recommendation method

With the development of e-commerce and more complexity of users' online shopping behaviours, traditional recommendation models do not meet the current requirements. Therefore, we need to improve the standard CF method to fulfil the complex context. We propose an improved CF algorithm based on multi-factors to increase the accuracy and certain diversity of recommender systems.

In our similarity model, we take into account user preference, location information, shopping time record. Through analysis, we know that location information and shopping time are recorded by observed platform directly. The object-node's degree and preference degree relate to their popular degree and corresponding users' ratings, respectively. Especially, for location information, we need to convert and normalise them.

Firstly, we assume the normalised location information of user i to object j as $w_{ij}(0 \le w_{ij} \le 1)$. Let t_{ij} denote the normalised time series that means the shopping time of user i to object j. Finally, we formulate the preference value of user i to object j as:

$$v_{ij} = v_{ij} * t_{ij} \tag{3}$$

in which, v_{ij} represents the ratings that object j obtained from user i.

We can formulate the improved cosine similarity as:

$$S_{il} = \frac{1}{\sqrt{k(u_i)k(u_l)}} \sum_{j=1}^{n} a_{ij} a_{ij} \left[\frac{\left(1 - \frac{|v'_{ij} - v'_{lj}|}{M}\right) * \left(1 + |w_{ij} - w_{lj}|\right)}{k(o_j)} \right]^{\alpha} \tag{4}$$

in which, M is the difference of the maximum and minimum rating scores. $k(u_i)$ expresses the degree of the user i (i.e., the number of objects that user i has selected). $k(o_j)$ expresses the degree of the object j (i.e., the number of users who have selected object j). α is an adjustable parameter.

Let P_{lj} denote the comprehensive preference value of object j which is obtained by target user l. The formulation can be expressed as:

$$p_{lj} = \sum_{i=1,\, l \neq i}^{m} S_{il} a_{ij} \tag{5}$$

In the process of recommendation, we sort the elements of P_{lj} uncollected by target in descending order. Generally, we deem target users prefer the top objects, so we recommend the sequential L objects to these users.

3.3 Recommendation performance metrics

According to many literatures, we apply five traditional metrics [F-measure, ranking score, root mean square error (RMSE), degree of popularity, hamming distance] to test the performance of the proposed method. The detailed descriptions of these metrics are as follows (Ju and Xu, 2014):

1 F-measure. The formulation of F-measure can be expressed as:

$$\text{F-measure} = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}} \tag{6}$$

Function $f(P, R)$ is a F-measure metric which measures accuracy. It is a comprehensive and more important metric which is composed of precision and recall to evaluate the accuracy of recommendations. In pattern recognition and information retrieval with binary classification, precision is the fraction of retrieved instances that are relevant, while recall is the fraction of relevant instances that are retrieved. Both precision and recall are therefore based on an understanding and measure of relevance. The precision can be expressed as a/L, where a represents the number of recommended objects selected by target users appearing in their corresponding ordered recommendation lists and L is the number of all objects in recommendation lists. The recall can be expressed as a/M, where a represents the number of recommended objects selected by users appearing in their corresponding ordered recommendation lists and M is the number of objects actually selected by target users. The higher the $f(P, R)$ is, the better the algorithmic performance will be.

2 Ranking score. For a target user i, a recommendation list is generated by ranking all his/her uncollected objects in descending order. The ranking score measures the relevant rank of each recommended object. If the recommended object j is ranked R_{ij} in list L_i, then its ranking score can be expressed as $r_{ij} = R_{ij} / L_i$. The mean of r_{ij} of the overall user-object pairs in the test set defines the averaging ranking score $\langle r \rangle$, which can be used to evaluate the algorithmic accuracy. The smaller the Ranking Score is, the higher the algorithmic accuracy will be.

3 RMSE is the square root of the ratio of the square of the deviation between observed and real value and the number of observations. The real values have to be substituted by the most reliable values (the best ones), due to the limitation of number of times to undertake observation in real-world measurement. In the process of evaluating personalised recommendations, RMSE calculates all the values of the square root of the mean of the quadratic sum of the deviation of real values and predicted values, the smaller the value, the higher the accuracy of the recommender's prediction, the formula is:

$$RMSE = \sqrt{\frac{1}{T}\sum_{i=1}^{T}\left(R_i - R_i^{'}\right)^2} \tag{7}$$

where R_i stands for real value, while stands for predicted value and R_i stands for times of observation.

4 Degree of popularity is used to measure directly by averaging the degree <*k*> of the overall recommended objects. The smaller the degree of popularity is, the more diversified the results will be. In the same precision, the smaller the mean value of recommended products' degrees is, the better the effect will be.

5 Hamming distance measures the diversity of algorithm by comparing the difference between two recommendation lists. If the overlapped number Q of objects in two recommendation lists generating for user i and user l, respectively. The Hamming distance can be expressed as:

$$H_{il} = 1 - Q / L \tag{8}$$

Generally speaking, a more personalised recommendation list should have larger Hamming distances to other lists. Therefore, we apply the mean value $S = \langle H_{il} \rangle$, averaged over all the user-user pairs, to measure the diversity of recommendations.

4 Experimental results and analysis

The program is written in Python running on Ubuntu 14.04. The tests are performed on three servers which are powered by Xeon(R) X3323 2.5 GHz with 8 GB Memory and 1 TB hard disk. Performance profiling is done by running our method against two benchmark data sets and one real-world data set.

In this section, we adopt two real-world data sets to verify the performance of our method. The first one is obtained from a well-known e-commerce platform. It contains 1,455 users, providing 232,768 ratings about 8,455 products (objects), between June and October 2016. The second one is extracted from another e-commerce. It contains 76,300 users with 146,200 location information, providing 2,860,360 ratings about 340,500 products (objects), between August and December 2016. Generally, each user from the selected website has an account which records some information about him. Each user has rated at least ten products by using a discrete number on the scale of 1 to 5. And only the one no less than three are considered. Besides, we divide each data set into two parts: the training set which contains 90% of the data and the remaining 10% of the data for the test.

For parameter α, we set F-measure as objective function and, that is, we will get the corresponding parameter values when the value of F-measure reaches minimum. According to many literatures on modified CF method, for instance the former research of Ju and Xu (2014), we predict the optimum value range of α change from –1 to 0. To find the optimal value of parameters α rapidly, we execute the iterative computation based on the strategy of binary search. All these computational definition and steps lead to lower computational cost. Table 1 show that the F-measure varies with the value of parameter α.

Table 1 The values of F-measure and α are corresponding to $L = 30$

	$\alpha = -0.77$	$\alpha = -0.78$	$\alpha = -0.79$	$\alpha = -0.80$
F-measure	0.071	0.073	0.080	0.083
	$\alpha = -0.81$	$\alpha = -0.82$	$\alpha = -0.83$	
F-measure	0.094	0.082	0.076	

Note: Present results are obtained by averaging over four independent divisions.

What we can know from Table 1 is that the algorithm reaches the highest value of Precision when the parameter $\alpha = -0.81$.

When generating recommendations, we need to set a certain length of recommendation list L. Generally, the recommendation list is no less than 30. We compare our algorithm (named IMCF) with the following four widely used recommendation algorithms: standard CF, MCF (Liu et al., 2009), CF-M (Xu, 2013), NNCosNgbr (Ju and Xu, 2014). We summarise the algorithmic performance in Table 2 to Table 4 with the small real-world dataset and Tables 5 to 7 with the big real-world dataset.

Table 2 Five metrics for different algorithms with the small real-world dataset

	CF	*MCF*	*NN-CosNgbr*	*CF-M*	*IMCF*
F-measure	0.071	0.081	0.082	0.090	0.103
Ranking score	0.134	0.122	0.121	0.110	0.098
RMSE	0.805	0.721	0.722	0.692	0.612
Degree of popularity	178	132	129	108	101
Hamming distance	0.512	0.544	0.543	0.602	0.664

Notes: For F-measure, ranking score, RMSE, degree of popularity and hamming distance, $L = 30$. Present results are obtained by averaging over four independent divisions.

Table 3 Five metrics for different algorithms with the small real-world dataset

	CF	*MCF*	*NN-CosNgbr*	*CF-M*	*IMCF*
F-measure	0.068	0.079	0.078	0.088	0.101
Ranking score	0.122	0.112	0.113	0.093	0.082
RMSE	0.801	0.712	0.711	0.680	0.599
Degree of popularity	166	121	120	103	92
Hamming distance	0.506	0.537	0.538	0.582	0.621

Notes: For F-measure, ranking score, RMSE, degree of popularity and hamming distance, $L = 40$. Present results are obtained by averaging over four independent divisions.

What we can see from the above tables is that the proposed algorithm has better performance than others. For instance, comparing with the standard CF, on the small real-world dataset, by recommendation list $L = 30$, the ranking score and root mean square can be further reduced by 26.9% and 24%, respectively. The F-measure can be further enhanced by 45%. Similarly, IMCF algorithm has lower ranking score and higher F-measure than NN-CosNgbr and CF-M algorithms. For degree of popularity and Hamming distance, our algorithm is also the best.

Table 4 Five metrics for different algorithms with the small real-world dataset

	CF	*MCF*	*NN-CosNgbr*	*CF-M*	*IMCF*
F-measure	0.062	0.071	0.070	0.081	0.099
Ranking Score	0.113	0.104	0.105	0.088	0.076
RMSE	0.786	0.701	0.701	0.663	0.584
Degree of popularity	162	112	113	98	86
Hamming distance	0.501	0.528	0.526	0.572	0.613

Notes: For F-measure, ranking score, RMSE, degree of popularity and hamming distance, $L = 50$. Present results are obtained by averaging over four independent divisions.

Table 5 Five metrics for different algorithms with the big real-world dataset

	CF	*MCF*	*NN-CosNgbr*	*CF-M*	*IMCF*
F-measure	0.063	0.072	0.073	0.081	0.102
Ranking Score	0.156	0.132	0.133	0.121	0.104
RMSE	0.895	0.821	0.822	0.796	0.663
Degree of popularity	365	278	277	248	195
Hamming distance	0.504	0.538	0.539	0.596	0.632

Notes: For F-measure, ranking score, RMSE, degree of popularity and hamming distance, $L = 30$. Present results are obtained by averaging over four independent divisions.

Table 6 Five metrics for different algorithms with the big real-world dataset

	CF	*MCF*	*NN-CosNgbr*	*CF-M*	*IMCF*
F-measure	0.053	0.066	0.065	0.072	0.092
Ranking Score	0.143	0.121	0.120	0.114	0.098
RMSE	0.845	0.802	0.803	0.762	0.628
Degree of popularity	325	238	237	214	173
Hamming distance	0.482	0.519	0.520	0.575	0.611

Notes: For F-measure, ranking score, RMSE, degree of popularity and hamming distance, $L = 40$. Present results are obtained by averaging over four independent divisions.

Table 7 Five metrics for different algorithms with the big real-world dataset

	CF	*MCF*	*NN-CosNgbr*	*CF-M*	*IMCF*
F-measure	0.046	0.058	0.059	0.066	0.087
Ranking score	0.136	0.114	0.115	0.101	0.081
RMSE	0.811	0.786	0.787	0.751	0.620
Degree of popularity	311	220	220	203	162
Hamming distance	0.466	0.504	0.503	0.561	0.602

Notes: For F-measure, ranking score, RMSE, degree of popularity and hamming distance, $L = 50$. Present results are obtained by averaging over four independent divisions.

The results of the above tables show that our proposed algorithm exceeds other four algorithms in all the five metrics: F-measure, ranking score, RMSE, degree of popularity, hamming distance.

IMCF algorithm adjusts the accuracy and diversity via parameter α. When $\alpha < 0$, our algorithm tends to recommend unpopular products to users. On the other hand, when $\alpha > 0$, our algorithm tends to recommend popular products to users. Certainly, users can adjust this parameter according to actual situation. In addition, for an online recommender system, we need to consider the processing time and memory consumption to display the performance of this recommender system. In the process of similarity computation, the computing range is significantly reduced which leads to lower computational complexity. So, the computational complexity of IMCF is similar to standard CF's. Furthermore, if a new object is added to the collection or a new user is registered to the recommender system, our algorithm can properly generate recommendation results for it.

5 Conclusions

In this paper, we propose a new CF algorithm based on multi-factors to alleviate the sparsity and cold-start problems. This method also increases the accuracy and diversity of recommendation results. In the process of similarity computation, we consider many influencing factors, for example, user location information obtained by PC or mobile devices, user preference degree reflected by rating scores of products, user shopping time recorded by e-commerce platforms and so on. The experimental results show that our algorithm is effective in addressing the sparsity problem and has high accuracy and more diversity.

Concerning future work, we will research in the following aspects:

1 How to keep the robustness of recommendation algorithm when it meets hostile attacks. Hostile attacks mean that someone makes hostile and large number of invalid ratings or evaluations to recommender systems. Through hostile attacks, it is possible to affect the availability of the recommender systems.

2 Research on the recommendation method by the fusion of multi-source and heterogeneous data. Construct a more complex model which considers implicit information of users extracted from different domains. For example, users reputation reflected by number of fans or others. And these fans may come from different SNS.

Acknowledgements

This research was supported by National Natural Science Foundation of China (71702164); Zhejiang Public Welfare Technology Applied Research Project, China (LGN18G010001); Ministry of Education Youth Fund Project (18YJC630223).

References

Aghdam, M.H., Analoui, M. and Kabiri, P. (2017) 'Collaborative filtering using non-negative matrix factorization', *Journal of Information Science*, Vol. 43, No. 4, pp.567–579.

Bellogín, A. and Sánchez, P. (2017) 'Collaborative filtering based on subsequence matching: a new approach', *Information Sciences*, Vols. 418-419, pp.432–446.

Ju, C. and Xu, C. (2014) 'Personal recommendation via heterogeneous diffusion on bipartite network', *International Journal on Artificial Intelligence Tools*, Vol. 23, No. 3, pp.1–22.

Kim, S-C., Sung, K-J., Park, C-S. and Kim, S.K. (2016) 'Improvement of collaborative filtering using rating normalization', *Multimedia Tools and Applications*, Vol. 75, No. 9, pp.4957–4968.

Kumar, N.P. and Fan, Z. (2015) 'Hybrid user-item based collaborative filtering', *Procedia Computer Science*.Vol. 60, No. 1, pp.1453–1461.

Kumar, V., Pujari, A.K., Sahu, S.K., Kagita, V.R. and Padmanabhan, V. (2017) 'Proximal maximum margin matrix factorization for collaborative filtering', *Pattern Recognition Letters*, Vol. 86, pp.62–67.

Liu, R-R., Jia, C-X., Zhou, T., Sun, D. and Wang, B-H. (2009) 'Personal recommendation via modified collaborative filtering', *Physica A*, Vol. 388, No. 4, pp.1–15.

Liu, S. and Meng, X. (2014) 'Approach to network services recommendation based on mobile users' location', *Journal of Software*, in Chinese, Vol. 25, No. 11, pp.2556–2574.

Nilashi, M., Ibrahim, O. and Bagherifard, K. (2018) 'A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques', *Expert Systems with Applications*, Vol. 92, pp.507–520.

Ren, L. and Wang, W. (2018) 'An SVM-based collaborative filtering approach for Top-N web services recommendation', *Future Generation Computer Systems*, Vol. 78, pp.531–543.

Shams, B. and Haratizadeh, S. (2017) 'TasteMiner: mining partial tastes for neighbor-based collaborative filtering', *Journal of Intelligent Information Systems*, Vol. 48, No. 1, pp.165–189.

Wang, Y., Deng, J., Gao, J. and Zhang, P. (2017) 'A hybrid user similarity model for collaborative filtering', *Information Sciences*, Vols. 418–419, pp.102–118.

Xu, C. (2013) 'Personal recommendation using a novel collaborative filtering algorithm in customer relationship management', *Discrete Dynamics in Nature and Society*, Article ID 739460, pp.1–9.

Zhou, X. and Wu, S. (2016) 'Rating LDA model for collaborative filtering', *Knowledge-Based Systems*, Vol. 110, pp.135–143.