2018 42nd IEEE International Conference on Computer Software & Applications

Web Items Recommendation Based on Multi-View Clustering

1st Hong Yu
School of Software
Dalian University of Technology
Dalian, China
hongyu@dlut.edu.cn

2nd Tiantian Zhang
School of Software
Dalian University of Technology
Dalian, China
ztt_dlut@163.com

3rd Jiaxin Chen
School of Software
Dalian University of Technology
Dalian, China
xpluto@qq.com

4rd Chen Guo
School of Software
Dalian University of Technology
Dalian, China
guochen95@163.com

5th Yahong Lian
School of Software
Dalian University of Technology
Dalian, China
lianyahong1@163.com

***Abstract*—Nowadays, using recommendation system to provide users with personalized recommendation service is significantly meaningful. However, traditional collaborative filtering methods may suffer from the cold start problem, while another common recommendation model called content-based recommendation may not have the ability to dig out the potential semantic features of web items sufficiently.**

In this paper, we propose a novel multi-content clustering collaborative filtering model (MCCCF) for recommendation system. The proposed model can apply multi-view clustering to the mining of the similarity and relevance of web items so that they can be used to improve the classic collaborative filtering. Consequently, the data sparsity problem can be solved. We propose to use multi-view clustering to analyse web items or users from different views such as user ratings and user comments so that it discovers deeper similarity and relevance. At the same time, features from multiple views can be used to complement the user views or item views where the features are deficient, which declines the problem of data sparsity drastically. In this way, we can analyse users' preference by their historical interaction features and supplementary behaviour features to give corresponding recommendation. Above all, the weak spots of the traditional model can be filled in and its performance can be improved. Extensive experiments on real world datasets show that our method outperforms the baselines remarkably.

***Index Terms*—Recommendation System, Multi-view Clustering, Content-based Recommendation, Collaborative Filtering**

I. Introduction

Recommendation System, which plays a more and more important role in nowadays online services, is significantly meaningful to people's lives. Recently, many online apps take advantage of personalized recommendation to help their users get the information that are relevant or attracted to them, so that users can be more satisfied when using their apps and can be released from searching for content they prefer.

Research reported in this publication was supported by the National Natural Science Foundation of China(61602081) and Natural Science Foundation of Liaoning Province(201602180).

A common kind of approach to the above task is called Collaborative Filtering (CF) [1]–[7], which uses the users' historical interactions in the product system to predict what they like. Another common approach of recommendation is called content-based recommendation [1], [8], [9], which can dig out the latent relevance between web items or/and users and use that to make the prediction. In many real-world apps and web sites, the performances of the two approaches are both pretty good. However, as the requirements of personalization become more demanding, they still suffer from some limitations and problems.

Specifically, only when the data of users' previous interaction with systems are plenty and rich can the Collaborative Filtering method works well. In other words, this method may suffer from data sparsity problem which may lead to a low quality recommendation. For example, in a new published app, because most of users may just register or have a little interaction with the new system, the limited engagement will be the obstacle against any high quality recommendations. On the other hand, the content-based recommendation approach has its limitations too. Traditional content-based method usually use single-view clustering to discover the latent relevance of items or/and users in a system, making other views of items that can provide with much more rich and complementary information of the items to be wasted. Thus, the quality of recommendation may be not so satisfactory. For example, if there are three items $item_a$, $item_b$, $item_c$. And they are of great similarity. However, $item_a$ and $item_b$ are similar to each other in one view, while $item_a$ and $item_c$ are relevant to each other in another view. Suppose that $User_i$ like $item_a$ so much, then the single-view clustering content-based recommendation may only recommend either $item_b$ or $item_c$ to $User_i$. Whatever prediction the system may make, it will miss one of the users' interests, so this recommendation system cannot cover a wild area.

Furthermore, in classic recommendation system, apps usu-

0730-3157/18/$31.00 ©2018 IEEE
DOI 10.1109/COMPSAC.2018.00064

ally use only one of the two common methods to recommend items for users. As either of the two methods has worked well in practical applications, whether integrating the two common methods into a new novel method can make another great progress or not is a considerably valuable topic to study, which means that how to combine the CF method with the content-based recommendation method is an important consideration.

So in order to address the limitation that have been mentioned before and integrate the two common method into a new one with high quality recommendation, we propose a novel recommendation model called Multi-Content Clustering Collaborative Filtering (MCCCF). We extend Multi-View Clustering into Content-Based Recommendation to discover the latent semantic relevance between items or/and users, then use KNN algorithm to get the value of the similarity of them. The proposed model can apply the similarity matrix on the traditional CF methods so that the data sparsity problem can be significantly alleviated. The novel MCCCF model can make the recommendation system with higher quality and more stable performance. We applied our MCCCF model on real-world datasets to conduct a series of experiments. The results demonstrate our proposed model outperforms traditional recommendation methods comparatively.

The main contributions of this paper are:

- Applying Multi-View Clustering into Content-Based Recommendation method, which can dig out the features of items or/and users more fully so that the correlations between them can be obtained more broadly.
- Integrating Collaborative Filtering method and Content-Based method so that the data sparsity problem in CF method can be solved. Proposing a novel recommendation model called MCCCF model.
- Applying our MCCCF model on several real-world datasets for experiments to demonstrate that our proposed model has a significantly better performance than the traditional methods.

II. Related Work

Before introducing our MCCCF model, we should discuss some preliminary studies. First we state out the problem about the Content-Based Recommendation and CF approach. Then we discuss the multi-view clustering.

A. Collaborative Filtering

Collaborative Filtering (CF) is one of the most classic algorithms in Recommendation System, which can learn users' preference and give its prediction by using the historical interaction information of users. The most well-known practice of CF is the commodity recommendation in Amazon.com [6] which can be briefly described as using users' previous shopping data to recommend new items to them.

With all of the representative literature, the classic Collaborative Filtering model has been extended to be capable to handle more complicated situation. Yet, there still are some challenges like cold start problem or data sparsity problem that need to be resolved.

B. Content-Based Recommendation

Another common recommendation model is called Content-based Recommendation which is aimed at recommending items for users through the analysis of characteristic features extracted from items and/or users. This type of method can be briefly viewed as three processes: features extracting, users' preference learning and recommending. And in the process of features extracting, some kind of technique to transmit the describing data of items to numerical data which can be more easily recognized will be needed.

In the past few years, research on CF models has become more in-depth and more multi-angle. Besides the classic CF model, a great number of new content-based recommendation algorithms have been proposed and achieved good performance [1], [2], [10]–[16].

Though Content-Based Recommendation has been developed into many varieties and its performance has been more and more satisfactory, it still face with the problem that the features of the items or/and users cannot be dug out deeply enough.

C. Multi-View Clustering

In practical applications, many kinds of real data appear in multiple views. Each view can be used for data mining independently. In detail, the intrinsic features of many real-world data sets are composed of different views [15]. For example, a document can be translated into a multilingual version, and each version can be as a view. A website can be classified according to its content, domain name or its hyperlinks. It can be seen that the descriptions of data sets in these different views are often compatible and complement each other. So instead of relying on a single view for learning analysis, integrating the attributes of multiple views together for learning clearly has better results. In the past decade, many multi-view clustering algorithms have been proposed, such as multi-view k-means [17], Multi-NMF [9] and CoNMF [18]. Experiments have shown that their performance is better than that of single-view clustering.

We believe that multi-view clustering can make progress in Content-Based Recommendation. In this paper, we choose the CoNMF method to be the multi-view clustering part of our model.

III. Preliminary Study

As we have mentioned before, the two common traditional approaches of recommendation both have their own shortcomings when they are used into practice. For the classic CF methods, data sparsity is one of the biggest problem it can meet. In a new started system, users may not comment/view/like anything or they just have viewed a small amount of items which is extremely rare compared to the information concluded in the whole system. This means that the system does not have enough historical interaction to make some more accurate predictions.

However, the data sparsity problem can be solved by content-based recommendation which is proposed by Wang

C [19]. But there comes another two problems: what if the content-based recommendation cannot dig out the latent semantic correlations of items or/and users deeply enough; And whether the content-based recommendation can be integrate into CF methods to handle its problem or not.

About the first problem, we find that multi-view clustering is a good way to learn the relevance of items. Traditional single-view clustering just classify items from one point of view of the items, which may lead to a huge waste of other views. For example, in an online shopping system like Amazon, the description of items given by the merchant can be one point of view to be clustered, the description given by the buyers who have already bought the items surely means a lot for clustering. If both of the views are considered, the correlations between items can be mined considerably deeper than the situation of single-view. So multi-view clustering will help a lot in content-based recommendation. As for the second problem, we will introduce the solution in IV.

IV. Multi-Content Clustering CF

As we have mentioned in previous sections that the traditional existing recommendation models may be faced with either the data sparsity or extracting feature insufficiently problem. In this paper, we propose a novel recommendation model called MCCCF. The proposed model applies multi-view clustering on traditional content-based recommendation, which makes it more capable to dig out the correlations in a system deeply; Further, we merge the multi-content-based recommendation with the CF approach, and use the relevance that we have obtained to solve the data sparsity problem.

Suppose that there are n_v views of attribution of the items in one system. The first thing we do is to reduce the dimension which also means extracting features from each view of attribution. In order to achieve this, we put VSM (Vector Space Model) and the TF-IDF technique into use. After the extracting process, we can generate a feature matrices for every single view n_v, which contains the weight data about the features of every views, and this will be represented as $\{V^{(1)}, V^{(2)}, ..., V^{(n)}\}$.

However, the features extracted from views of items always conclude a large amount of noise which need to be eliminated such as the punctuations, the stop words and some common/ambiguous words like a.k.a. etc. To deal with this, we use the sklearn tool-kit in python and only retain the useful features.

Since we have noise-filtered features of items $\{V^{(1)}, V^{(2)}, ..., V^{(n)}\}$, we make the multi-view clustering on them to attain the latent correlations between items. In II, we have mentioned three multi-view clustering methods which can be applied. In this paper we choose CoNMF alogrithm. Similar to traditional NMF, it has the objective function which can be optimized by the update rules [18]:

$$
\begin{aligned}
H^{(s)} &\leftarrow H^{(s)} \frac{{W^{(s)}}^T V^{(s)}}{{W^{(s)}}^T W^{(s)} H^{(s)}} \\
W^{(s)} &\leftarrow W^{(s)} \frac{\lambda_s V^{(s)} H^{(s)} + \sum_{t=1}^{n_v} \lambda_{st} W^{(t)}}{\lambda_s W^{(s)} H^{(s)} H^{(s)T} + \sum_{t=1}^{n_v} \lambda_{st} W^{(s)}}
\end{aligned}
\tag{1}
$$

in which, the coefficient matrix $W^{(s)}$ is the results of clustering, where the entry $W_{ij}^{(s)}$ denotes the weights of $clus_j$ to $item_i$. Since we have the clustering results, we need to find out the correlations between the items. The KNN algorithm is a great way to work out. Therefore, we apply KNN on the results and generate the nearest neighbour matrices for each view which can be represented as $\{NN^{(1)}, NN^{(2)}, ..., NN^{(n)}\}$. For the $item_i$, $NN_{ij}^{(s)}$ denotes the j-th relevant item to it in view s. Table I gives the structure of the matrices:

TABLE I
Structure of Matrices

ItemName	No.1 Neighbor	No.2 Neighbor	No.3 Neighbor
Avenger	Avengers 2	Iron Man	Captain America
Batman	The Dark Knight	Batman v Superman	Batman 3
Spider-Man	Spider-Man 2	The Amazing Spider	Spider-Man Man?HomeComing

After all the process above, we get the deeper latent correlations among the items which can be used in the subsequent process. Note that though the correlations are at the item level, they mean a lot to the user interaction modeling.

In traditional CF approach, the users' historical interaction data sometimes can be extremely rare which will lead the recommendation engine to a relatively poor performance. Consider a situation that users rate items: there can be such a user-item rating matrices where the horizontal axis denotes the users existing in the system and the vertical axis denotes the items, the entry value of this matrices is a rating that is given by one user to one item. The detail can be found in Fig. 1.

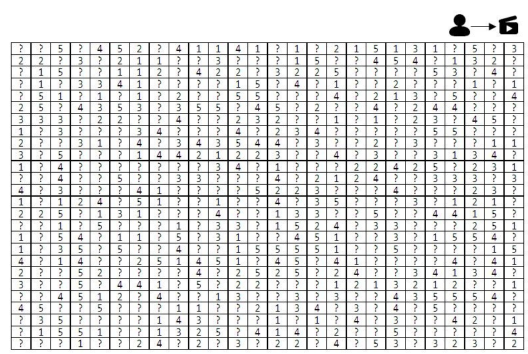

Fig. 1. User Item rating matrix

From Fig. 1, we can see that there is a large number of users who just rate rare items (This kind of case is common in real word apps or websites, users may not interested in rating or they have just registered for a short time). The ratings can be seen as one type of users' interaction data with the system which is the key point of the CF approach. The limits of this will make the quality of recommendation decline drastically.

However, consider such a situation that users tend to give the similar ratings on highly similar items. In real world films and television comments site (Douban, IMDB, Netflix, etc.), if $User_i$ rates *Iron Man* for 5 stars, then he may give out the similarly high ratings for *Iron Man* 2 and $TheAvengers$ to a large extent; and if he dislikes *Kill Bill Vol.1*, then he may just give a low rating on *Kill Bill Vol.2* to a great degree.

In other words, if we can obtain the relevance among items in our system, we can then use this to make a retrospective analysis for users' previous history data and complement the blanks where users did not truly give out their comments. Though the complements we make are subjective to some extent, it is significant indeed. And as we have mentioned before, the correlations we need are the results of running multi-view clustering and KNN on our items which we can obtain from the previous process of our model.

In detail, our MCCCF model applies multi-view clustering on features of items, and run KNN means on the clustering result to generate a series of correlation matrices which can be used in CF approach to complement the user historical interaction matrices. Fig. 2 gives the process of the complementing in our model, if $User_i$ rates *Iron Man* for 5 stars, and he never rates the neighbours of *Iron Man* that can be found in the correlation matrices, which is *Iron Man 2* and *The Avengers*. And the proposed model will complement the blank ratings as 5 stars.

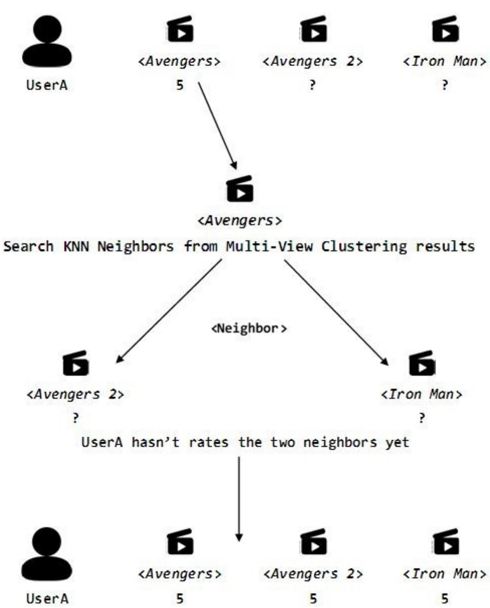

Fig. 2. An example of completing rate

Also, for every user-item entry, our model will check if the item has some neighbours that are never commented by users, if so, the blank user-neighbour entry will be complemented. And in this way, the condition that the user-item entries are rare can be solved, which means the data sparsity problem will be settled. Formally, the proposed model can be briefly described as the following five steps: (1) Features Extracting; (2) Noise Filtering; (3) Multi-view Clustering; (4) KNN Matrices Generating; (5) Complementing Interaction Matrices. The detailed process is shown in Fig. 3.

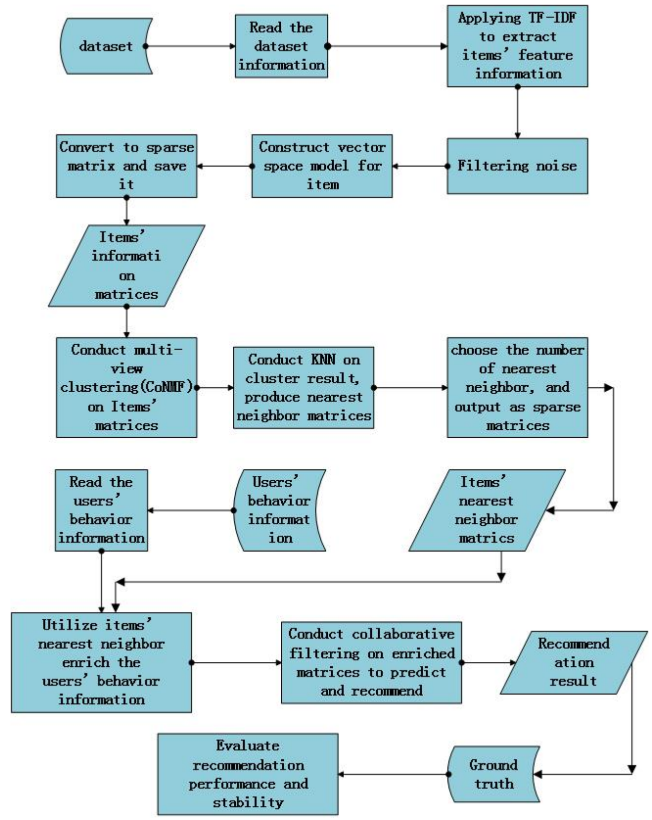

Fig. 3. the Framework of Proposed MCCCF

To summarize, the proposed MCCCF model integrate the content-based recommendation into CF approach, use the latent correlation of items in system to complement the user interaction matrices so that the data sparsity problem in the CF approach can be solved, and the quality of recommendation can be improved. In **Algorithm 1**, we give out the overview of our model.

A. *Advantages of MCCCF*

The model in this paper merges multi-view content-based recommendation into classic collaborative filtering to solve the problems existing in traditional recommendation algorithms. On one hand, the proposed model takes advantage of multi-view clustering to extract features of items in online system so that the condition of information wasting will be excluded to a great extent. Besides, the complements of features can help dig out the correlations of items more deeply so that the later modelling procedure can be more accurate. On the other hand, the proposed model fits real online systems well. With

Algorithm 1 Multi-Content Clustering CF(MCCCF)

Input:

Multi-view item information matrix $\{F^{(1)}, F^{(2)}, ...F^{(n_t)}\}$, user behaviour matrix R, parameter about neighbour number $\{N_1, N_2, ...N_v\}$, cluster number k and under grouped number K.

Output:

Prediction and recommendation matrix PR of users' behaviour and nearest neighbour matrix of item $N = \{NN^{(1)}, NN^{(2)}, ..., NN^{(n_t)}\}$.

1: normalize each view's matrix $F^{(s)}$.
2: apply noise filtering to feature information, then generate the product feature information matrix $\{V^{(1)}, V^{(2)}, ..., V^{(n_v)}\}$.
3: initialize $V^{(s)} = W^{(s)}H^{(s)}$.
4: **while** CoNMF doesn't converge to minimum value and current iteration number is less than threshold **do**
5:     **for** each s from 1 to n_v **do**
6:         utilize (1) to solve $W^{(s)}$ and $H^{(s)}$ iteratively.
7:     **end for**
8: **end while**
9: **return** $\{W^{(s)}\}$, $H\{^{(s)}\}$
10: calculate nearest neighbour of $\{W^{(s)}\}$ and return N.
11: utilize N to enrich R, and return R.
12: conduct collaborative filtering on R, and return PR.

the spreading of internet, the large scale of datasets in online system will definitely make the user-item matrix rarely sparse. As we have described above, model in this paper can handle that kind of problem relatively well.

V. EXPERIMENT

In this section, to evaluate the performance and quality of the proposed model, we contrast it against some traditional recommendation methods. We use two sets of real world datasets to experiment, along with two classic algorithms to be compared with.

A. Dataset

In this paper, we use the following two datasets to make the performance evaluation of the proposed model:

Movie-Len: Consists of 2 billion rates, along with 465000 tags generated for 27000 movies and 138000 users who rated the movies, note that the rates all range from 1 to 5. In this experiment, we randomly select 500 users who have rated more than 30 movies with the 1000 related movies and their tags to make evaluation. This dataset was published in Apr. 2015 with times of updating and modifying, which is advanced enough to be processed.

Hetrec2011: Consists of 86000 rates from IMDB and Rotten Tomatoes, along with a number of tags and 2113 users. We still randomly pick 500 users who have rated more than 30 movies with the 1000 related movies and their tags to be experimented.

B. Baselines

In experiments, we bring the following two classic Collaborative Filtering methods into compare:

NMF: A classic matrix decomposition algorithm. In Collaborative Filtering, we can transform the missing rates prediction problem into regression problem with machine learning. With NMF, the user-item rating matrix can be decomposed into user matrix and item matrix with the cluster features of them, which can regress through dot product with the missing part available.

SVD: Another classic matrix decomposition algorithm based on singular value. Same as NMF, it can decompose user-item rating matrix into user matrix, item matrix and singular matrix which can be regress through dot product.

The details of the above algorithms are too detailed to be discussed in this paper.

C. Evaluation Method

MAE (Mean Absolute Error): The averaged value of absolute error which can be used to reflect the practical performance of prediction well. The lower value of MAE indicates a better performance of the model. (2) shows the details.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|observed_i - predicted_i| \tag{2}$$

RMSE (Root Mean Squared Error): The mathematical expectation of the squared value of error between real rates and predicted rates. The lower value of RMSE also indicates a better performance of the model. (3) shows the details.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(observed_i - predicted_i)^2} \tag{3}$$

D. Experimental Settings and Results

In this evaluation, we divided the rates data into two groups with ratio of 2:1 to be recognized as training set and testing set. Random grouping and selecting is the point to make sure that the evaluation is fair and real so that it will not always show the preferred results because of specific data. We ran this experiment for 10 times and then took the average values as the final results. And the parameter of group number is set as 20. Note that, in this experiment, we have data with two views: names and tags, and the weights of both the two views are set as 1. About the weights, we have mentioned before, it can adjust to the practical use.

TABLE II
EXPERIMENT RESULT

DataSet	Evaluation Method	RMSE	MAE
HetRec2011	NMF	2.6943	2.4801
	SVD	2.7901	2.5571
	MCCCF	**1.0527**	**0.7882**
Movie-Len	NMF	2.9452	2.7291
	SVD	3.0095	2.7781
	MCCCF	**1.1187**	**0.8458**

Table II shows the results of our evaluation. From Table II we can see, the proposed model has an excellent performance compared with the two baselines. Both the MAE and RMSE value of our model are lower than the two classic algorithms significantly, which means that the proposed model reaches a higher quality of prediction. The proposed method can reach a considerable performance. It has settled the data sparsity problem in traditional Collaborative Filtering and it also has a better move in the mining of the latent relevance of items.

Formally, with the previous introduction of our model and the results of the experiments we can see: first, the proposed recommendation model takes advantage of multi-view clustering to dig out the latent relevance of the items more deeply so that the correlation between items can be more visible and accurate. Further, it uses this correlation to handle the sparsity problem in collaborative filtering so that the recommendation quality can be improved significantly.

Conclusion and Future Work

Recommendation System is playing a very important role in nowadays online systems. As more and more top applications begin to use the intelligent auto-push system to push the latent preferred items to users, how to further improve the quality and stability of that system has become a popular field to research.

In this paper, we analyse the shortcomings of the existing methods of recommendation: cannot handle the data sparsity problem. Therefore we propose a novel Multi-Content Clustering Collaborative Filtering (MCCCF) model. The proposed model merges the multi-content-based recommendation into collaborative filtering, it combines the features of items and the historical interaction data that users generated previously to learn and recommend. In detail, on one hand, our model uses multi-view clustering, which clusters the items with multiple views, to learn the latent structure of the data which can be viewed as a deeper correlation between items provided by the complementary features from views. On the other hand, the proposed model takes advantage of this correlation to complement the interaction logs in the system to handle the data sparsity problem, such that the users can be better modeled to be recommended.

Experiments on real world datasets with the proposed model and baselines show that our model can improve the quality of recommendation with the deeply mined relevance indeed. In future work, we will aim at extend our research to deep learning and parameters learning to enlarge the scale of the relevance analysis in our model to make a better quality of recommendation.

References

[1] A. M. Elkahky, Y. Song, and X. He, "A multi-view deep learning approach for cross domain user modeling in recommendation systems," in *The International Conference*, 2015, pp. 278–288.

[2] T. Hofmann, "Collaborative filtering via gaussian probabilistic latent semantic analysis," vol. 13, no. 4, pp. 259–266, 2003.

[3] S. M. Kakade and D. P. Foster, "Multi-view regression via canonical correlation analysis," in *Conference on Learning Theory*, 2007, pp. 82–96.

[4] A. Blum, "Combining labeled and unlabeled data with co-training," in *Eleventh Conference on Computational Learning Theory*, 1998, pp. 92–100.

[5] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization." *Nature*, vol. 401, no. 6755, pp. 788–91, 1999.

[6] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering," *IEEE Internet Computing*, vol. 7, no. 1, pp. 76–80, 2003.

[7] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted boltzmann machines for collaborative filtering," in *International Conference on Machine Learning*, 2007, pp. 791–798.

[8] D. Ramage, P. Heymann, C. D. Manning, and H. Garcia-Molina, "Clustering the tagged web," 2009, pp. 54–63.

[9] J. Liu, C. Wang, J. Gao, and J. Han, *Multi-View Clustering via Joint Nonnegative Matrix Factorization*, 2013.

[10] R. M. Bell and Y. Koren, "Scalable collaborative filtering with jointly derived neighborhood interpolation weights," in *IEEE International Conference on Data Mining*, 2007, pp. 43–52.

[11] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2008, pp. 426–434.

[12] A. S. Das, M. Datar, A. Garg, and S. Rajaram, "Google news personalization: scalable online collaborative filtering," in *International Conference on World Wide Web*, 2007, pp. 271–280.

[13] Y. Wang, S. Wang, N. Stash, L. Aroyo, and G. Schreiber, "Enhancing content-based recommendation with the task model of classification." in *Knowledge Engineering and Management by the Masses - International Conference, Ekaw 2010, Lisbon, Portugal, October 11-15, 2010. Proceedings*, 2010, pp. 431–440.

[14] L. Zhang and Y. Zhang, "Discriminative factored prior models for personalized content-based recommendation," in *ACM International Conference on Information and Knowledge Management*, 2010, pp. 1569–1572.

[15] S. Ren, S. Gao, J. Liao, and J. Guo, "Improving cross-domain recommendation through probabilistic cluster-level latent factor model," in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015, pp. 4200–4201.

[16] M. Jiang, P. Cui, X. Chen, F. Wang, W. Zhu, and S. Yang, "Social recommendation with cross-domain transferable knowledge," *IEEE Transactions on Knowledge & Data Engineering*, vol. 27, no. 11, pp. 3084–3097, 2015.

[17] X. Cai, F. Nie, and H. Huang, "Multi-view k-means clustering on big data," in *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, 2013, pp. 2598–2604. [Online]. Available: http://www.aaai.org/ocs/index.php/IJCAI/IJCAI13/paper/view/6979

[18] X. He, M. Y. Kan, P. Xie, and X. Chen, "Comment-based multi-view clustering of web 2.0 items," in *International Conference on World Wide Web*, 2014, pp. 771–782.

[19] C. Wang and D. M. Blei, "Collaborative topic modeling for recommending scientific articles," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2011, pp. 448–456.