IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.7, Special Issue.4, Feb. 2019 E-ISSN: 2347-2693

Movie Recommendation Model Using Stochastic Gradient Descent For Collaborative Filtering In Social Media Mining

C. Premila Rosy

Department of Computer Science, Idhaya College for Women, Kumbakonam, Tamilnadu, India

Available online at: www.ijcseonline.org

***Abstract*—**Nowadays, many people appetite to watch TV-shows or - series anytime and anywhere they want. In recent years, online TV has experienced exponential growth. Netflix is one of the parties that jumped into the world of online streaming services. In this effort, many subsist movie recommendation approaches learn a user ranking model from user feedback with respect to the movie's content. Unfortunately, this approach suffers from the sparsity problem inherent in SMR data. Collaborative filtering (CF) is the workhorse of recommender engines since it can perform feature learning on its own, meaning it learns for itself what features to use. CF can be split into Memory-Based Collaborative Filtering and Model-Based Collaborative filtering. Here compare results from memory-based CF, model-based CF and third approach which uses an algorithm called 'Stochastic gradient descent' for collaborative filtering. The propose stochastic gradient descent algorithm using movie recommender system. In this propose system use movie lens dataset, one of the most common datasets used to implement and test recommender engines. It contains 100,000 movie ratings from 943 users and a selection of 1682 movies. Evaluate the results using the Root Mean Squared Error (RMSE) and Mean Absolute Error(MAE).

***Keywords*—**Movie Recommendation System, Memory-Based Collaborative Filtering, Model-Based Collaborative Filtering, Stochastic Gradient Descent

I. INTRODUCTION

Nowadays, many people appetite to watch TV-shows or - series anytime and anywhere they want. In recent years, online TV has experienced exponential growth. Netflix is one of the parties that jumped into the world of online streaming services. Netflix, which was begin in 1999 as an online video shop, has become the most-used, and a still strong growing American online streaming provider specialized in video-on-demand distribution. Currently, they are alive in over 190 countries all over the world with over 100 million subscriptions. Recently, the number of Netflix sub-scrimptions within the United States exceeded the number of subscriptions for regular paid cable TV. Regarding customer satisfaction in general, recommendations based on the usersbehaviour has faced an important role in the e-commerce customer satisfaction. Many web shops, like Amazon and Alibaba, use recommender techniques to recommend items to their visitors, which are items that are similar to the one they searched for, or they have bought recently.

Next to that, advance systems are also widely used by online travel agencies like booking.com and Expedia, so that visitors can discover their ultimate holiday destination match, based on their search behavior, historic bookings, or similar users. Not only is Netflix using recommender systems to improve customersatisfaction, but also because people are bad in choosing between many options. From consumer analysis Netflix has conducted, it suggested that an ordinary Netflix user loses it interest after 60 seconds of choosing or reviewed more than 10 to 20 titles in detail. There-fore, Netflix developed a advance system over the years, which exists of various algorithms that are combined into an ensemble method. Netflix is a company that handles a big collection of television programs and movies, by streaming it at any time via online (computers or TV). This firm is profitable because the users do a monthly payment to get access to the platform. However, the clients can abort their subscriptions at any time. Therefore, it is vital for the business to keep the users hooked to the platform and not to lose their interest. This is where recommendation [1] [15] [16] systems start to play an important role, it is pivotal to provide valuable suggestions to users. The recommendation systems are increasing their popularity among the service providers, because they help to increase the number of items sold, offer a diverse selection of items, the user satisfaction increases, as well as the user fidelity to the company, and they are quite helpful to have a better understanding of what the user wants. Then, it is easier to lead the user to make better decisions from a wide variety of cinematographic products. The recommender systems take into account not only information about the users but also about the items they consume; comparison with other products, and so on and so forth. Nevertheless, there are many algorithms available to perform a recommendation [16] system. For instance, (i) Popularity, where only the most popular items are recommended (ii) Collaborative Filtering[5], which looks for patterns in the user activity to produce user-specific recommendations; (iii) Content-based Filtering, the recommendation [8][16]of items with similar information the user has liked or used in the past; (iv) Hybrid Approaches.

II. RELATED WORK

The rapid proliferation of online social networks and the rise of streaming movie services, people are changing their movie selection habits. Instead of picking movies off a shelf at a store, entertainment seekers decide what to watch digitally based, partially, on the online recommendations [4] of their friends. A movie watcher, in turn, can then recommend a movie or show to their friends via social media. As a result, Social-aware Movie Recommendation systems (SMRs) have been developed to provide relevant movie recommendations to a target audience. The benefits of movie recommendation systems are well-recognized today, and major SMR Web sites like Netflix, IMDB, and Douban continue to grow rapidly. The popularity of tagging systems accommodates a great opportunity to improve the performance of item recommendation. Although abide approaches use topic modeling to mine the semantic information of items by grouping the tags labeled for items, they overlook an important property that tags link users and items as a bridge. Thus, these methods cannot deal with the data sparsity without commonly rated items (DS-WO-CRI) problem, limiting their recommendation performance. Towards clarify this challenging problem, we propose a novel tag and rating based collaborative filtering (CF) model [19] for item recommendation, which first uses topic modeling to mine the semantic information of id for each user and for each item respectively, and then incorporates the semantic information into matrix factorization to factorize rating information and to capture the bridging feature of tags and ratings [14] between users and items. The context-aware [3] hierarchical Bayesian method[20]. First, we advance the use of spectral clustering for user-item sub grouping, so that users and items in similar contexts are grouped. We then advance a novel hierarchical Bayesian model [10] that can make predictions for each user-itemsubgroup, our model incorporate not only topic modeling to mine item content but also social matrix factorization [7] to handle ratings and social relationships. Deep Learning approach to map users and items to a latent space where the similarity between users and their preferred items is maximized. The general framework named NCF, short for neural network- based Collaborative Filtering[19]. NCF is generic and can accurate and generalize matrix factorization under its framework. To supercharge NCF classic with non-linearities, we propose to leverage a multi-layer perception to learn the user-item interaction function.

The new MF method aimed at learning from implicit feedback [6]effectively while satisfying the requirement of online learning. The learning algorithm that efficiently optimizes the implicit [17] [18] MF model without imposing a uniform-weight restriction on missing data. The algorithm is unique in its sparsity encouraging property and can easily cope with many non-informative features. Secondly, the training is based on Variation Bayes inference that is less prone to over-fitting and does not require cross validation. While we are not first to present a Variational Bayes MF model, MF-EFS is different than these previous works. Combine heterogeneous user feedback [9] by transforming both explicit and implicit feedback into a unified pair wise preference-based representation, which can then be used as training data for matrix factorization models[12]. The second part of this article is focused on the study of time-aware recommendation. Relevance Feedback (RF) to automatically adjust the intra weights within each modality and the inter weights among different modalities based on user click-through. We also design a voting approach by tracking the users specific browsing behaviours on video shots to estimate the feature weights associated with the individual shots. Improved assumption, group Bayesian [20] personalized ranking (GBPR), via introducing richer interactions among users. Conventional collaborative filtering [11] algorithms is the search for neighbours among a large user population of potential neighbours. Item-based algorithms avoid this bottleneck by exploring the relationships between items first, rather than the relationships between users. A novel recommendation model based on a joint matrix factorization (JMF) that factorizes the user-item (user-movie) matrix, while also exploiting the contextual information as additional regularization terms.The hybrid movie recommendation approach via social tags and preferred ratings. First, we extract normalize, and recondition social tags according to user preference. The user preference is based on social content annotation, which includes tags and ratings. Then, our model can benefit from unifying the potential capability of a personalized scoring system (e.g., singular value decomposition [SVD] of a matrix) and a tagging system (e.g., the preference-topic model, tag normalization and reconditioning). Finally, in terms of the recommendation results, our hybrid method has outperformed the existing user-based collaborative filtering (CF) algorithms including the user-based CF, the CF model, and the topic model-based CF model. The cross-space affinity learning (CSAL) algorithm to learn the affinity measure across different spaces with heterogeneous structures. The basic idea is to first construct cross-space tensors (CSTs) from these heterogeneous spaces to represent the correlation [2] between them, and the cross-space affinity [13] can then be learned by exploiting a set of order must on the affinity from a training pool.

III. PROPOSED APPROACH

The propose system, random movies were determined from the movie dataset to the users in the test set. In other words, this system does not consider the historic rating behaviour of the user. Note that the set from which movies were elect, were excluding the already seen movies. This was complete to prevent recommending already seen movies. The Memory-based method uses user to user and item to item correlations based on rating behaviour to predict ratings and recommend items for the users in coming also called as Neighbourhood-Based CF. this mechanism handling users" rating data to compute similarity between users and/or items is used for making recommendations.

A) Memory-Based Collaborative Filtering

Memory-Based CF mechanism is handling in many commercial systems as it is easy to implement and is effective. *User-item filtering* The collaborative filtering technique is the user-based technique. Rather of searching similar movies as one has seen in the previous section, user-based collaborative filtering will search for similar users. The first step of this technique is to find a neighbourhood of similar users and then aggregate the ratings of these users to form a prediction. To acquisition the k nearest neighbours of a given user u, similarity measures like the Pearson correlation coefficient or the Cosine similarity is used. For user-based collaborative filtering, the Cosine similarity is handling again, as described in the item-based collaborative filtering section. However, the items are now replaced with users. This now becomes $simcosine$

$$sim_{cosine}\,(\vec{x},\vec{y}) = \frac{\vec{x}\cdot\vec{y}}{\|\vec{x}\|^2\|\vec{y}\|^2} \qquad (1)$$

Whereux and uy are two users, x and y are the row vectors that mean the two user's profile ratings. After the neighbourhood of effective user ua is created, the top k users are picked and will be represented as the set (a) of active user ua. Next, a guess of a specific movie for active user ua can be made by averaging the ratings of the same movie of the users in (a). In a formula, this would be written as

$$\hat{r}_{aj} = \frac{1}{|N(a)|}\sum_{j\in N(a)} r_{ij} \qquad (2)$$

$j\in(a)$ Whereraj is the predicted rating for active user ua of movie j and rij is the predicted rating for user$i\in N(a)$ of the same movie j. *Item-item filtering* After building a key random recommender, more advanced techniques were implemented to create more explainable and accountable recommendations. The item-based collaborative filtering technique is a well-known and widely handling recommender technique. Item-based collaborative filtering is a approach that produces recommendations based on the relationship between items (in this research: movies) inferred from the rating matrix. The first step of this approach is to calculate the $n\times n$ similarity matrix **S** that contains all item-to-item similarities. In this process, a given similarity measure is used, for ex-ample Pearson correlation and Cosine similarity. In this research, Cosine similarity is used as proposed the Cosine similarity is defined by the following formula, where ix and iy are two items, x and y are the row vectors that perform the two items ratings: $simcosine$

$$sim_{cosine}\,(\vec{x},\vec{y}) = \frac{\vec{x}\cdot\vec{y}}{\|\vec{x}\|^2\|\vec{y}\|^2} \qquad (3)$$

Next, it is constant to store only the k most similar items of an item to reduce the size of **S**, so it becomes a $n\times k$ matrix where $k\ll n$. The k items which are better similar to item i are denoted by vector (i). The second step is to calculate the certain recommendations based on **S**. This is complete by calculating a weighted sum of the users rating for the corresponding items, according to the following formula.

$$\hat{r}_{ai} = \frac{1}{\sum_{j\in w(i)} S_{ij}}\sum_{j\in w(i)} S_{ij}\, r_{ai} \qquad (4)$$

$j\in(i)$ In this formula, rai is the estimate rating of user a for item i and Sij is the similarity between item i and j. In addition, item j must be in (i), which is decide as a subset of $S(i)$ that contains all known ratings of user a that are in $S(i)$.

B. Model-Based Collaborative Filtering

Model-Based Collaborative Filtering algorithm uses RS information to create a model that generates the recommendations. Unlike Memory-Based CF, Model-based CF does not use the whole dataset to compute predictions for real data. There are various model-based CF algorithms including Bayesian [20]Networks, Clustering Models, and Latent Semantic Models such as Singular Value Decomposition (SVD) for dimensionality reduction of rating matrix. The goal of this approach is to uncover latent factors that explain observed ratings.

- Ithasan n X m matrix consisting of the ratings of n users and m items. Each element of the matrix (i, j) mean how user i rated item j. Since we are task with movie ratings, each rating can be expected to be an integer from 1-5 (reflecting one-star ratings to five-star ratings) if user i has rated movie j, and 0 if the user has not estimate that particular movie.
- For each user, we need to recommend a set of movies that they have not seen yet (the movie rating is 0). To do this, we will aftermath use an approach that is similar to weighted K-Nearest Neighbours.
- For each movie j user i has not seen yet, we find the set of users U who are similar to user i and have seen movie j. For each similar user u, it takes u„s rating of movie j and multiply it by the cosine similarity of user i and user

u. Sum up these weighted ratings, category by the number of users in U, and we get a weighted average rating for the movie j.

Finally, we set the movies by their weighted average rankings. These average rankings serve as an estimate for what the user will rate each movie. Movies with higher average rankings are highly likely to be favoured by the user, so we will recommend the movies with the highest average rankings to the user.

Singular Vector Decomposition based Collaborative Filtering

Another form of collaborative filtering, one that is not neighbourhood-based" like one has seen in this section so far, is the Singular Vector Decomposition (SVD) Collaborative Filtering technique. In general, SVD is used to avoid the number of features of a data set. For recommender systems, one is only activity in the matrix factorization part where one keeps the same dimensionality. Roughly said, matrix factorization is the action of finding matrices whose product is the rating matrix. In formula, it is

$$A = USV^T \quad (5)$$

where A is the given $n{\times}m$ matrix, U is the $n{\times}n$ matrix containing the eigenvectors ofAAT, S is the $n{\times}m$ matrix containing the square root of the eigen values associated with AAT on its diagonal, and V is the $m{\times}m$ matrix that contains the eigenvectors of ATA. So, let us estimate that each movie i is associated with a vector qi and each user u is associated with a vectorpu. This means that for a given movie i, the elements of qi measure the extent of interest the user has in items that are high on the corresponding factors. The same holds for a accord user u and its corresponding vectorpu. When one takes the dot product of these vectors, one will get the approximated rating $rui$of user u for movie i. In formula, it is

$$\hat{r}_{ui} = q_i^T p_u \qquad (6)$$

A method to cause the rating matrix denser, and thus make it easier to compute the movie vector qi and profile vectorpu, is to use an imputation technique. However, the downside of handling imputation is that it might distort the data considerably. Hence, an alternative method is using only the ratings that are available. Besides, this method also clear over fitting by using a regularized model. This is complete by minimizing the regularized squared error on the training set such that:

$$min_{\bar{q}^*,p^*} \sum_{(u,i)\in k} (r_{ui} - q_i^T p_u)^2 + \lambda(\|q_i\|^2 + \|p_u\|^2) \qquad (7)$$

In this formula, K is the set that abide of all rui that are in the training set. In addition, by using this formula, one is apt to learn from previous ratings and do it in such way that it generalizes these previous ratings so it is able to predict future ratings as well. It prevents over fitting by the continualλ which restricts the degree of regularization. In order to find the actual qi and $pu$to predict therui, the formula above must be minimized. This can be complete using a stochastic gradient descent optimization. To be exact, for each accord rating in the training set, the system calculates the prediction error. The formula yields:

Next, the qi and pu are changed such that

$$q_i \leftarrow q_i + \gamma(e_{ui} \cdot p_u - \lambda \cdot q_i) \qquad (9)$$

$$p_u \leftarrow p_u + \gamma(e_{ui} \cdot q_i - \lambda \cdot p_u) \qquad (10)$$

Where$eu,=ru,v-PuqvT$, which represents the difference between predicted score and actual score and γ is the learning rate.

C. Stochastic Gradient Descent

The low-rank matrix decomposition aims to divide rating matrix into two low-rank factor matrices, and uses the product of these two factor matrices to approximate the original rating matrix. The approximation is to small the error between the predicted and the original rating matrix. More specifically, considerate the user number is m, the number of items is n, and rating matrix $R(m{\times}n)$ that records the preference of the uth user on the Vth item at the (u,v) entry, ru,v. The key to the problem is to find two low-rank matrices,$(m{\times}d)$and $Q(n{\times}d)$, that makes PQTapproximate the rating matrix R. The optimization problem is equal to the following Formula (1):

$$min_{P,Q} \sum_{(u,v)\in R} \left(r_{u,v} - P_u {q_v}^T\right)^2 + \lambda_P \|P_u\|^2 + \lambda_Q \|q_v\|^2 \qquad (11)$$

Where Pu means the uth line of matrix P, qvTmeans the vth line of matrix Q. Simply Considering$PuqvT$is the predicted score to users. While the decomposition is easy to generate over-fitting problem, so add the normalization factors λP and λQ into the objective function, which try to avoid over fitting effect. The core idea of SGD is to select a ru,vin R, then finds out the corresponding factor vector Pu from the user factor matrix P, qv from the item factor matrix Q, calculates the predicted score $PuqvT$, and updates parameters according to the following two rules.

$$P_u \leftarrow P_u + \gamma\left(e_{u,v} q_v - \lambda_P P_u\right) \qquad (12)$$

$$q_v \leftarrow q_v + \gamma\left(e_{u,v} P_u - \lambda_Q q_v\right) \qquad (13)$$

Where $eu,=ru,v-PuqvT$, which represents the difference between predicted score and actual score and γ is the learning rate. *Synchronous Operation* Conflicts occur frequently when worker threads need to work together at critical resources. In order to avoid the conflict problems when workers need to obtain the next free block or updating the sharing data, it needs synchronous operation (Algorithm 4) among workers. A high available distributed data management framework. It guarantees strong consistency of data in a distributed environment. We adopt Zoo Keeper to realize distributed sharing lock, achieving the synchronous operation among workers. For the purpose of calculating the entire RMSE of test sets, we also use Zoo Keeper to realize a producer-consumer queue and to calculate each block's RMSE parallel using its watcher registration and asynchronous notification mechanism.

Although worker threads may encounter conflicts when they attempt to obtain the sharing lock and those failing to get the lock have to wait for the lock to be released, the execution procedure of SGD algorithm on each block which occupies most of the time can perform without keeping the sharing lock, which largely avoids the conflicts among workers. And the task that occupies the lock will release it in a very short period of time, so the waiting time for those attempting to get the sharing lock will not to be long even when conflict occurs. It greedy that our strategy to those failing to get the sharing lock is to have the thread sleep for a while before attempting to obtain the lock again.

Data Sharing

After DSGD performing a rule, it has a information synchronous process before processing the next rule. However, there is no data synchronous complication to FPSGD which runs in a shared memory system. In DCE, in order to avert the synchronous process, data needing to be shared can be stored in a high-speed public storage system. In our experiment, we selection Memcached, a high-performance storage system based on memory, and all the workers need to store and update the sharing data in the Me cached cluster. We grid rating and factor matrices into blocks and allow a number to each block. Then appreciation encapsulates a block number and its content into a key-value pair and stores it into Memcached Cluster. Each thread attempts to get a block that is free and with the minimum number of being executed on SGD algorithm. And according to the block number, thread will collectMemcached to get the content of corresponding rating blocks, factor blocks and other parameters. After the execution of SGD on each block, the new value should be amending into Memcached Cluster again. According to the limitation of Memcached that the maximum size of its storing value is 1M map it will compress those exceeded value to put them into Memcached. And we build a first-level index to those value size over 1M after compression so that they can be stored properly. Though this method solves the limited, it needs more than once to get those exceeded value that still need to use first-level index after the compression operation.

IV. Results and Discussion

The experimented with a popular database, the MovieLens100k dataset. The MovieLens data set contains 100,000 ratings (1–5 scales) from 943 users on 1682 movies (items), where each user has rated at least 20 movies. Evidently, accurate rating predictions indirectly result in relevant recommendations to users. Hence, prediction performance is an essential aspect of recommender systems from a business perspective. We therefore mainly focus on optimizing the prediction performance of the models. To measure this, we calculate the root mean square error (RMSE) and the mean absolute error (MAE) metrics. Suppose that we want to evaluate a models prediction performance on the dataset$S \in \{Validaton, test\}$.

$$RMSE(S) = \sqrt{\frac{1}{|O_S|} \sum_{(u,i) \in O_S} (r_{ui} - \hat{r}_{ui})^2} \qquad (14)$$

And

Both measures are error based, which means that lower values indicate better performance. The RMSE more strongly penalizes large errors compared to the MAE. Including both measures allows us to distinguish models that make a few large errors from models that consistently have a small prediction error. The RMSE and MAE are metrics that measure the prediction performance on individual user-item pairs. Besides this, the quality of a recommender system is also characterized by various aspects of the top-n lists it is able to produce. Examples of these aspects include novelty and diversity. The novelty of recommendations refers to the degree of newness of the items in the recommendation list. Commonly, the newness of a recommended item is inversely related to the similarity with the items that the active user has already rated in the past. The diversity of recommendations is related to the extent to which the recommended items are dissimilar. Another characteristic is the personality of the top-n lists, which refers to the degree of diversity among the top-n recommendations to different users.

Table 1: Comparison of RMSE

Techniques	Dataset	
	Train	Test
User Collaborative Filtering	2.81	2.91
Item Collaborative filtering	3.24	3.37
singular value decomposition	2.31	2.74
Stochastic Gradient Descent	0.62	1.0

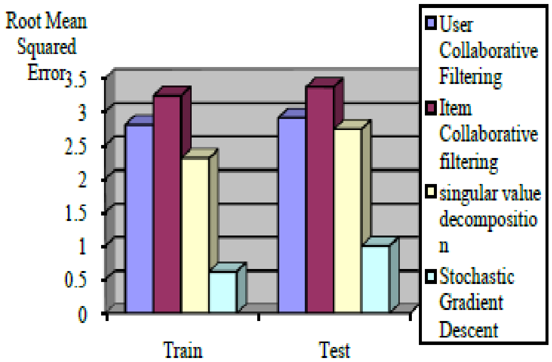

Figure 1: Comparison of RMSE train/test for different recommenders

Table 2: Comparison of Techniques

Techniques	No of Nodes			
	User Collaborative Filtering	Item Collaborative filtering	singular value decomposition	Stochastic Gradient Descent
Train	2.45	2.83	2.0	0.62
Test	2.61	2.91	2.51	0.71

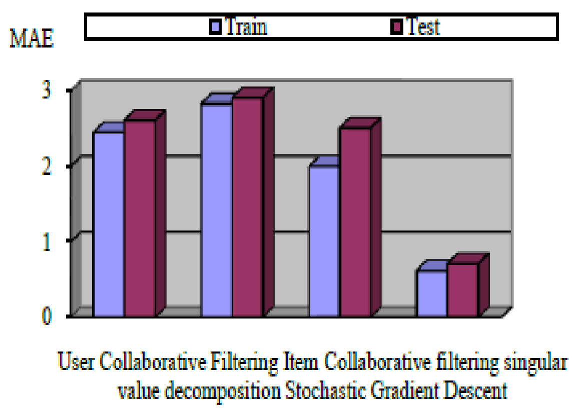

Figure 2 Comparison of MAE by train/test for different recommenders.

V. CONCLUSION

Recommender systems are a able new technology for extracting additional value for a business from its user databases. These systems aid users find items they want to buy from a business. Recommender systems profit users by enabling them to find items they like. Conversely, they relief the business by generating more sales. Recommender systems are rapidly becoming a crucial tool in E-commerce on the Web. In this propose system, a movie recommendation mechanism within Netflix will be built. The dataset that is used here consists of over ML100K. The two main types of recommender systems are Content-Based Filtering and Collaborative Filtering (CF). In general, Collaborative filtering (CF) is the workhorse of recommender engines since it can perform feature learning on its own, meaning it learns for itself what features to use. CF can be category into Memory-Based Collaborative Filtering and Model-Based Collaborative filtering. MovieLens dataset, one of the most common datasets used to implement and test recommender engines. It contains 100,000 movie ratings from 943 users and a selection of 1682 movies.

REFERENCES

[1] L. Canini, S. Benini, and R. Leonardi. Affective recommendation of movies based on selected connotative features. Circuits and Systems for Video Technology, IEEE Transactions on, 23(4):636–647, 2013.

[2] C. Chen, X. Zheng, Y. Wang, F. Hong, and D. Chen. Capturing semantic correlation for item recommendation in tagging systems.In AAAI, pages 108–114, 2016.

[3] C. Chen, X. Zheng, Y. Wang, F. Hong, Z. Lin, et al. Contextaware collaborative topic regression with social matrix factorization for recommender systems. In AAAI, volume 14, pages 9–15, 2014.

[4] A. M. Elkahky, Y. Song, and X. He.A multi-view deep learning approach for cross domain user modeling in recommendation systems. In WWW, pages 278–288, 2015.

[5] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua.Neural collaborative filtering. In Proceedings of the 26th International Conference on World Wide Web, pages 173–182. International World Wide Web Conferences Steering Committee, 2017.

[6] X. He, H. Zhang, M.-Y.Kan, and T.-S. Chua. Fast matrix factorization for online recommendation with implicit feedback. In Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, pages 549–558. ACM, 2016.

[7] N. Koenigstein and U. Paquet. Xbox movies recommendations: Variationalbayes matrix factorization with embedded feature selection. In RecSys, pages 129–136. ACM, 2013.

[8] N. N. Liu, L. He, and M. Zhao. Social temporal collaborative ranking for context aware movie recommendation. TIST, 4(1):15, 2013.

[9] T. Mei, B. Yang, X.-S. Hua, and S. Li. Contextual video recommendation by multimodal relevance and user feedback. TOIS, 29(2):10, 2011.

[10] W. Pan and L. Chen. Gbpr: Group preference based bayesian personalized ranking for one-class collaborative filtering. In IJCAI, volume 13, pages 2691–2697, 2013.

[11] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl.Item-based collaborative filtering recommendation algorithms. In WWW, pages 285–295. ACM, 2001.

[12] Y. Shi, M. Larson, and A. Hanjalic.Mining contextual movie similarity with matrix factorization for context-aware recommendation. TIST, 4(1):16, 2013.

[13] J. Tang, G.-J.Qi, L. Zhang, and C. Xu.Cross-space affinity learning with its application to movie recommendation. TKDE, 25(7):1510–1519, 2013.
[14] S. Wei, X. Zheng, D. Chen, and C. Chen.A hybrid approach for movie recommendation via tags and ratings. Electronic Commerce Research and Applications, 18:83–94, 2016.
[15] M. Yan, J. Sang, and C. Xu. Unified youtube video recommendation via cross-network collaboration. In Proceedings of the 5th ACM on International Conference on Multimedia Retrieval, pages 19–26. ACM, 2015.
[16] X. Zhao, G. Li, M. Wang, J. Yuan, Z.-J.Zha, Z. Li, and T.-S.Chua. Integrating rich information for video recommendation with multitask rank aggregation. In ACM Multimedia, pages 1521–1524. ACM, 2011.
[17] X. Zhao, J. Yuan, R. Hong, M. Wang, Z. Li, and T.-S.Chua.On video recommendation over social network. In International Conference on Multimedia Modeling, pages 149–160. Springer, 2012.
[18] H. Li, R. Hong, D. Lian, Z. Wu, M. Wang, and Y. Ge, "A relaxed ranking-based factor model for recommender system from implicit feedback," in Proceedings of IJCAI‟16, 2016, pp. 1683–1689.
[19] C. Wang and D. M. Blei, "Collaborative topic modeling for recommending scientific articles," in Proceedings of KDD‟11. ACM, 2011, pp. 448–456.
[20] D. Lian, Y. Ge, N. J. Yuan, X. Xie, and H. Xiong, "Sparse bayesian content-aware collaborative filtering for implicit feedback," in Proceedings of IJCAI‟16. AAAI, 2016.