Clustering-Based Factorized Collaborative Filtering

Nima Mirbakhsh
Department of Computer Science
Western University, London, ON, Canada
smirbakh@uwo.ca

Charles X. Ling
Department of Computer Science
Western University, London, ON, Canada
cling@csd.uwo.ca *

ABSTRACT

Factorized collaborative models show a promising accuracy and scalability in recommendation systems. They employ the latent collaborative information of users and items to achieve higher accuracy of recommendation. In this paper, we propose a new approach to improve the accuracy of two well-known, highly scalable factorized models: SVD++ and Asymmetric-SVD++. These are cutting-edge factorized models that have played a key role in the Netflix prize winner's solution. We first employ collaborative information to categorize the users and items. We then discover the shared interests between these categories. Including this new information, we extend these cutting-edge models regarding two main goals: 1) to improve their recommendation accuracies; 2) to keep the extended models still scalable. Finally, we evaluate our proposed models on two recommendation datasets: MovieLens100k, and Netflix. Our experiment shows that adding the shared interests among categories into these models improves their accuracy while maintaining scalability.

Categories and Subject Descriptors

H.3 [**Information storage and retrieval**]: Retrieval models, Information filtering, Clustering; H.2 [**Database management**]: Database Applications—*Data mining*

Keywords

Collaborative Filtering, Factorizing, Neighborhood-Aware

1. INTRODUCTION

Between 2007 to 2009, Netflix[1] hold a competition on recommendation systems for the best collaborative filtering algorithm that predicts users' ratings on a collection of its movies. For details of the provided dataset by Netflix, see [4, 1]. The goal of the competition was to achieve a RMSE below 0.8563 (10% improvement from the benchmark). Many models have been tried and proposed by almost 2000 teams who attended in this competition. Finally, a blended model including more than 50 methods has won this competition. SVD++ and Asymmetric-SVD++ [4, 5, 1] were two models that play a central role on improving the accuracy of the winner model. These models effectively include the implicit and explicit information about the users and the items into a factorized model which improves the recommendation accuracy while keeping the model scalable. To the best of our knowledge, these algorithms are still (if not the most) one of the most scalable and accurate collaborative filtering models [1].

In this paper, we employ the collaborative filtering information again to find the categories that these users and items may belong to. We then track how users inside each category rate the other categories containing the items. We expect that adding this novel information will improve the recommendation quality of the current collaborative filtering models including SVD++, and Asymmetric-SVD++. We follow these two main goals: 1) to improve the recommendation accuracy; 2) to keep the extended models scalable. We also evaluate how the quality of the clusters will affect the improvement of the prediction accuracy. Employing clustering to categorize collaborative information, and using these clusters to predict the unknown preferences has been employed before in a number of works [6, 2]. However, to the best of our knowledge there is not any previous works which consider the shared preferences between the categories. Our experiment shows that including this deduced knowledge improves the recommendation accuracy while keeping the model still scalable and practical.

2. PROPOSED MODELS

In a general CF problem, we have a set of users $U = \{u_1, u_2, \ldots, u_n\}$ and a set of items $I = \{i_1, i_2, \ldots, i_m\}$ that they are companied by a rating matrix $R = [r_{ui}]_{n \times m}$ where r_{ui} represents the rating of user u on item i. Collaborating filtering consists of predicting unknown r_{ij}s based on the known $r_{i'j'}$ s inside the rating matrix R. Matrix Factorization (MF) addresses this problem by decomposing the ratings matrix, R, into two lower dimension matrices Q and P which contain corresponded latent vectors of each user and item in the length of $k \ll m, n$. Such a model is close to the singular value decomposition (SVD) technique for finding latent vectors in information retrieval. Thus, MF and SVD refer to a same concept in this paper. The predictor

*Charles X. Ling is the contact author of this paper.
[1]http://www.netflix.com

RecSys'13, October 12–16, 2013, Hong Kong, China.

http://dx.doi.org/10.1145/2507157.2507233 .

function of a Biased Matrix Factorization (BMF) model is as follows:

$$r_{ui} = q_i^T p_u + b_i + b_u \quad (1)$$

where $q_i \in \mathbb{R}^k$ and $b_i \in \mathbb{R}$ are the corresponded vector and bias value of item i, and $p_u \in \mathbb{R}^k$ and $b_u \in \mathbb{R}$ are the corresponded vector and bias value of user u.

There are two common approaches to include neighborhood information into factorization models: 1) $item-item$ models which consider if user u is interested in item i and its similar items. 2) $user-user$ models that consider if user u and its similar users are interested in item i. However, item-item models are usually preferable as their less space and time complexity. This is because of the typical larger number of users in recommendation systems. Although, both models ignore the neighborhood information of possible categories that items and users may belong to. Thus, we first try to find the possible categories and then apply the shared interests between these categories in a number of CF models.

We first apply BMF on the known ratings to learn the latent vectors of each user and item. Kmeans then is applied on these latent vectors with different selection of K (number of clusters) to find possible categories of items and users. Rating matrices are typically sparse in recommendation systems. Hence, using latent vectors helps to reduce the complexity of clustering these large and sparse datasets. After finding these clusters (categories), we then correspond each a latent vector. We consider the shared ratings between these categories in a new matrix R^*. Lets assume C_u and C_i as the clusters that contain user u and item i. In this new rating matrix, every $r^*_{C_u,C_i}$ reflects the average rating of the users inside the category C_u on the items inside the category C_i. We define $n' < n$ and $m' < m$ as the number of categories for users and items respectively.

2.1 Clustering-Based SVD++

As discussed earlier, neighborhood aware models employ the similarities between users and items to improve the recommendation accuracy. However, they usually need to compute all pairwise similarities between items or users, which its complexity grows quadratically with the input size [5]. Koren [4, 5] solves this limitation by factoring the neighborhood model, which scales both item-item and user-user implementations linearly with the size of the data [5]. Thus, he effectively integrates implicit and explicit neighborhood information about the users and items to extend the pure SVD model (the BMF model in 1). He assumes all the feedbacks from the users no matter what the feedbacks are, as the implicit information, and considers the ratings as the explicit neighborhood information. He defines $N(u)$ and $R(u)$ as the sets of all implicit and explicit feedbacks from user u. However, $N(u)$ is assumed equal to $R(u)$ in his experiment on the Netflix dataset[4]. In SVD++, Koren corresponds each item i with two latent vectors $q_i, y_i \in \mathbb{R}^k$, and models each user with a corresponded latent vector $p_u \in \mathbb{R}^k$ plus the sum of the latent vectors of all the items inside $N(u)$. Thus, he defines the SVD++'s predictor function as follows:

$$r_{ui} = q_i^T (p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j) + b_i + b_u \quad (2)$$

Employing these implicit feedbacks in the SVD model has shown a promising improvement of prediction accuracy in practice[5]. We expect that employing implicit information about the categories that users and items belong to is useful as well. Thus, we assign two latent vectors $q^*_{C_i}, y^*_{C_i} \in \mathbb{R}^k$ to each category of items, and a latent vector $p^*_{C_u} \in \mathbb{R}^k$ to each category of users which reflects their direct and implicit effect on the ratings. To add the effect of the category in this model, we change the equation 2 as follows:

$$r_{ui} = \left((1-\alpha)q_i + \alpha q^*_{C_i}\right)^T \left(((1-\alpha)p_u + \alpha p^*_{C_u}) + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} \left((1-\alpha)y_j + \alpha y^*_{C_j}\right) \right) + b_i + b_u \quad (3)$$

We call this new model "CB-SVD++" in the experiment section.

2.2 Clustering-Based Asymmetric-SVD++

In the Asymmetric SVD++, Koren employs the explicit feedbacks, $R(u)$, into the SVD++ model and proposed a factorized item-item model as follows:

$$r_{ui} = q_i^T \left(p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j + |R(u)|^{-\frac{1}{2}} \sum_{j \in R(u)} (r_{uj} - b_u - b_j)x_j \right) + b_i + b_u \quad (4)$$

where $x_j \in \mathbb{R}^k$. To apply the shared interest among categories into this model, we assign three latent vectors $q^*_{C_i}, y^*_{C_i}, x^*_{C_i} \in \mathbb{R}^k$ to each category of items, and a latent vector $p^*_{C_u} \in \mathbb{R}^k$ to each category of users which reflect their direct, implicit, and explicit effect on the ratings. The new predictor function is as follows:

$$r_{ui} = \left((1-\alpha)q_i + \alpha q^*_{C_i}\right)^T \left(((1-\alpha)p_u + \alpha p^*_{C_u}) + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} \left((1-\alpha)y_j + \alpha y^*_{C_j}\right) + |R(u)|^{-\frac{1}{2}} \sum_{j \in R(u)} (r_{uj} - b_u - b_j) \left((1-\alpha)x_j + \alpha x^*_{C_j}\right) \right) + b_i + b_u \quad (5)$$

We call this model "CB-ASVD++" in our experiment. In all the discussed models, the parameters are determined by minimizing the associated regularized squared error function through gradient descent.

3. EXPERIMENT RESULTS

In section 2, we propose a clustering-based approach to improve current CF models. We apply our extending approach on two cutting-edge CF methods with respect to improve their prediction accuracies while keeping their scalability. In this section, we setup our experiment on two well-known recommendation datasets to validate our hypothesis about these extensions[1]. MovieLens100k [3], and Netflix dataset which contains over 100 million ratings from 480189 users who has rated 17770 movies. MovieLens100k contains 100,000 ratings from 943 users on 1682 movies where each user has rated at least 20 movies [3]. The package includes five randomly 80%/20% splits of dataset into training and

[1]The implementation package is publicly accessible at: https://sites.google.com/site/nmirbakhsh/projects/CBSVDpp

test sets. Netflix dataset contains over 100 million ratings from 480189 users who has rated 17770 movies. We run each algorithm 5 times on the datasets to remove the effect of random initialization of latent vectors on the predictions. Thus, our reported RMSE results are the average RMSEs of these 5 runs.

3.1 Categories

We start by applying BMF on both datasets to find their users' and items' latent vectors. These latent vectors (learned on the train sets) then are used for the clustering purpose. Kmeans is applied on the achieved latent vectors to find possible categories of items and categories of users inside the datasets. It is expected that items (users) with similar latent vectors are similar in reality as well. We guess different selection of possible categories by changing in the number of clusters. It usually achieves in 5-10 tries. Finally, by trying different size of clusters on the proposed models, the number of clusters that achieves less RMSE (higher accuracy) on the validation set will be selected as the best choice of m', and n'.

Table 1 shows the movies inside a number of found clusters on the Netflix dataset. As shown, it seems that movies in same genre and almost similar years of production tend to be in same clusters. For instance, 'category 1499' includes different versions of 'Lord of the Rings' and 'Harry Potter'. These movies are both in 'Adventure' and 'Fantasy' genres[2] and also have released between 2001 to 2004. Or, 'category 1028' contains a number of classical movies in the 'Adventure' genre. As no information about the users is provided, so the clusters of users cannot be judged. Lets remind that the movies' names (identities) are not used anywhere in this experiment. These names are only employed for better demonstration of the clusters. On the other hand, the quality of the clusters is effective on the prediction accuracy of the extension models. For instance, the 'CB-SVD++' model results a RMSE of 0.90564 for a random assignment of the clusters ($m' = n' = 100$) on the MovieLens100k dataset. Comparing this value with the achieved RMSE of 0.89928 which is the result of employing the kmeans clusters of the latent vectors, it can be seen that the low quality of clusters will decrease the prediction accuracy of our extensions.

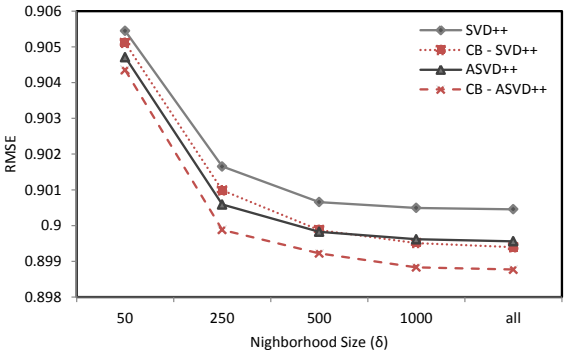
Table 1: The table shows the movies inside a number of formed clusters on the Netflix dataset. As shown, it seems that movies in same genre and almost similar years of production tend to be in same clusters.

Category ID	Movie Name (Production Year)
Category 1499	Harry Potter and the Prisoner of Azkaban(2004), Harry Potter and the Sorcerer's Stone(2001), Lord of the Rings: The Two Towers: Bonus Material(2002), Harry Potter and the Chamber of Secrets(2002), Lord of the Rings: The Fellowship of the Ring: Bonus Material(2001), Lord of the Rings: The Return of the King: Bonus Material(2003)
Category 1028	Treasure Island(1950), Bend of the River(1952), The Far Country(1955), Hondo(1953), Night Passage(1957), Rio Bravo(1959), Sink the Bismarck(1960), Sahara(1943), They Were Expendable(1945), Run Silent(1958), Rio Grande(1950), She Wore a Yellow Ribbon(1949), Destination Tokyo(1943)]

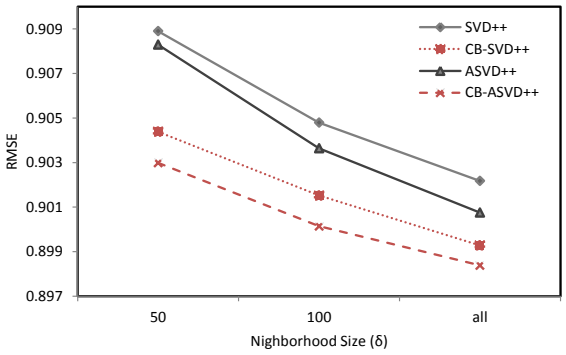
[2]http://www.imdb.com

Table 2: The RMSE results of applying clustering-based extension of a number of factorized CF methods on the Netflix dataset. As the results show, with increasing k the extensions result better improvements.

Models	k=50	k=100	k=200
SVD	0.90658	0.90510	0.90479
SVD++	0.90045	0.89952	0.89910
CB-SVD++	0.89939	0.89852	0.89803
ASVD++	0.89955	0.89907	0.89904
CB-ASVD++	0.89876	0.89770	0.89721



(a) Netflix



(b) MovieLens100k

Figure 1: The accuracy of the proposed clustering-based models applying on the two datasets. It shows that our proposed extensions outperform both 'SVD++' and 'Asymmetric-SVD++' on these datasets ($k = 50$ is used to achieve these results).

3.2 Comparison

In our all extensions an α value is used to control the balance between direct influence of users and items, and the influence of the categories that they belong to on the prediction function. Figure 2 shows how the accuracy of 'CB-ASVD++' changes with different selection of α. For $\alpha = 0.0$, this model does not consider the effect of the categories so the result RMSE is similar to the 'ASVD++"s accuracy as expected. Also, the shared interests among the categories are too general to be employed lonely for the recommendation purpose. Thus, the accuracy is not improved or even gets worse for very large selection of α. A validation set is used to determine the best selection of α in our experiment.

The Table 2 shows the RMSE results of applying clustering-based extension on a number of factorized CF methods on the Netflix dataset. As the results show, with increasing k the extensions result better improvements. However, another factor which highly affects the accuracy of the neighborhood-aware methods is the neighborhood size that is applied on the model. Lets define the maximum number of employing neighborhood as δ. Figure 1 illustrates how increasing δ results a better prediction accuracy. It also shows that employing the deduced information about the categories that the items and users belong to, is independent from the explicit and implicit feedbacks. Adding the deduced information about the shared interests among the categories improves the prediction quality for any neighborhood size in both datasets. For the MovieLens dataset, the clustering-based extension of 'SVD++' ('CB-SVD++') shows a better prediction accuracy than 'Asymmetric-SVD++' which employs the explicit feedbacks. For employing full set of neighbors, this advantage can be seen on the Netflix dataset too.

The extension models have almost the same complexity as the non-extended models. However, they add a preprocessing complexity in the clustering step. As Table 3 illustrates, the learning time of the extensions is less than twice of the non-extended models on the MovieLens100k dataset. The clustering was also not much time consuming because we perform the clustering on the low dimension latent vectors. For instance, the clustering of the Netflix's users takes less than a hour using the Rapidminer[3] software in our PC with 3.30 GHz CPU. Both extended and non-extended models also take similar number of epochs for converging in the learning time. Thus, the extended models keep the scalability of those models which was our second goal.

4. DISCUSSIONS AND CONCLUSIONS

Because of the disadvantages of the user-user and integrated models, in this paper we only focused on the item-item models. However, we have tried the clustering-based extension method on those models. A similar improvement of prediction results have been observed for those extensions. For the integrated model which is presented in [5], the improvement was even slightly higher. For instance, the basic integrated model achieves a RMSE of 0.89558 on the MovieLens100k dataset. By applying our extension on this model, the RMSE falls to 0.89298. However, the space and time complexity of this model increases dramatically by selecting larger number of δ. It is really difficult to replicate the blended model of the Netflix prize winner's solution. However, based on the central role of SVD++ and Asymmetric-SVD++ in that blended model, we expect that applying our extensions will slightly improve its accuracy.

To summarize, we propose a new approach to improve the accuracy of two well-known highly scalable factorized models: SVD++ and Asymmetric-SVD++. We first employ collaborative information to categorize users and items. We then track how users inside each category rate the other categories containing the items. Including this new information, we extend these cutting-edge models regarding two main goals: 1) to improve their recommendation accuracies. 2) to keep the extended models still scalable. Finally, we evaluate our proposed models on two recommendation systems datasets: MovieLens100k, and Netflix. Our experiment shows that applying the shared interests among the categories into these models improve their accuracies. These extensions also have the same complexity as the non-extended models.

[3]http://www.rapidminer.com

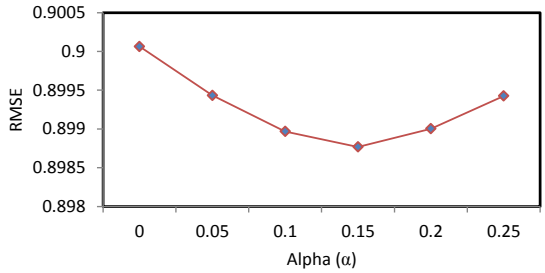


Figure 2: The accuracy of the proposed CB-ASVD++ model applying on the Netflix dataset. It shows how changing α affects the final predictor's accuracy.

Table 3: A comparison between the learning time of the extended and non-extended models on the MovieLens100k dataset (millisecond/epoch). As shown, the extensions increase the learning time to less than twice of the non-extended models' learning times.

SVD++	CB-SVD++	ASVD++	CB-ASVD++
43ms	73ms	54ms	92ms

5. REFERENCES

[1] C. Desrosiers and G. Karypis. A comprehensive survey of neighborhood-based recommendation methods. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, *Recommender Systems Handbook*, pages 107–144. Springer US, 2011.

[2] M. Gueye, T. Abdessalem, and H. Naacke. A cluster-based matrix-factorization for online integration of new ratings. In *Journées de Bases de Données Avancées (BDA)*, pages 1–18, 2011.

[3] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '99, pages 230–237, New York, NY, USA, 1999. ACM.

[4] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, pages 426–434, New York, NY, USA, 2008. ACM.

[5] Y. Koren. Factor in the neighbors: Scalable and accurate collaborative filtering. *ACM Trans. Knowl. Discov. Data*, 4(1):1:1–1:24, Jan. 2010.

[6] B. Xu, J. Bu, C. Chen, and D. Cai. An exploration of improving collaborative recommender systems via user-item subgroups. In *Proceedings of the 21st international conference on World Wide Web*, WWW '12, pages 21–30, New York, NY, USA, 2012. ACM.