Inf Syst Front
DOI 10.1007/s10796-016-9668-4

Leveraging clustering to improve collaborative filtering

Nima Mirbakhsh[1] · **Charles X. Ling**[1]

Abstract Extensive work on matrix factorization (MF) techniques have been done recently as they provide accurate rating prediction models in recommendation systems. Additional extensions, such as neighbour-aware models, have been shown to improve rating prediction further. However, these models often suffer from a long computation time. In this paper, we propose a novel method that applies clustering algorithms to the latent vectors of users and items. Our method can capture the common interests between the cluster of users and the cluster of items in a latent space. A matrix factorization technique is then applied to this cluster-level rating matrix to predict the future cluster-level interests. We then aggregate the traditional user-item rating predictions with our cluster-level rating predictions to improve the rating prediction accuracy. Our method is a general "wrapper" that can be applied to all collaborative filtering methods. In our experiments, we show that our new approach, when applied to a variety of existing matrix factorization techniques, improves their rating predictions and also results in better rating predictions for cold-start users. Above all, in this paper we show that better quality and more quantity of these clusters achieve a better rating prediction accuracy.

Keywords Collaborative filtering · Recommendation system · Matrix factorization

✉ Nima Mirbakhsh
nmirbakhsh@gmail.com

Charles X. Ling
cling@csd.uwo.ca

[1] Western University, London, ON, Canada

1 Introduction

Extensive work on matrix factorization (MF) techniques have been done recently as they provide accurate rating prediction models in recommendation systems (Koren et al. 2009; Steck 2010a; Mirbakhsh and Ling 2015). MF techniques characterize both users and items by latent vectors inferred from rating patterns. Recommendation methods based on MF show a good rating prediction accuracy and scalability (Koren et al. 2009). Several neighborhood-aware extensions on matrix factorization technique such as Asymmetric SVD (Koren and Bell 2011) and SVD++ (Koren 2008) improve rating prediction accuracy even further. Still employing clustering on the set of users and items has been a basic and practical solution in recommendation systems (Connor and Herlocker 1999). In this paper, we integrate the advantages of both disciplines to achieve a higher rating prediction accuracy.

We propose a novel method that captures the common interests between the cluster of users and the cluster of items in a latent space. In a new model, we make a "coarse" matrix where each cluster of users (items) is considered one user (item). We then employ a MF technique on this "coarse" matrix to predict if both the user and its related user-cluster are interested in the item and its related item-cluster.

Our experimental results show that our new approach, when applied to a variety of existing MF techniques, including biased matrix factorization (BMF) (Koren et al. 2009) and Asymmetric SVD (AsySVD) (Koren and Bell 2011), improves their rating prediction accuracy. We also evaluate how the quality of these clusters will affect the improvement of rating prediction accuracy which is commonly measured by root mean square error (RMSE).

Employing clustering to categorize collaborative information, and using these clusters to predict the unknown

Published online: 29 June 2016

preferences, has been employed before in a number of works such as (Xu et al. 2012; George and Merugu 2005; Jamali et al. 2011; Gueye et al. 2011). However, many of these methods mainly focus on the individual clusters and ignore the shared interest between the clusters. Or, they proposed expensive probabilistic models which cannot be easily integrated with the state of the art collaborative filtering methods. In contrast, our extension approach is easy to implement and can be applied as a "wrapper" on many other collaborative filtering methods.

This paper is an extended version of the paper that was published in the 2013 ACM Recommender System conference (RecSys'13) (Mirbakhsh and Ling 2013), and includes the following additions:

1. We introduce a new method (Section 3.1), Clustering-Based Matrix Factorization, as an extension of biased matrix factorization technique which is originally proposed by Koren in (2009). This technique is simpler than CB-AsySVD or CB-SVD++ (our former proposed techniques), requires less tuning, and achieves a good RMSE results.
2. In this paper we conduct additional experiments for cold-start users (Section 4.3) where CBMF achieves a good result for those users.
3. In Sections 3.3 and 4.4.2 we also produce large number of clusters for users and items in different sizes. Our experiment shows that employing more clusters in different sizes improves the rating prediction accuracy of our proposed CBMF model.
4. In Section 4.4.1, we report four different clustering algorithms including expectation maximization and K-Means in our experiment. This also evaluates the impact of clustering algorithms and their quality on improving the rating prediction accuracy.

The remainder of the paper is structured as follows. In Section 2 we overview basic definitions, the works related to factorized collaborative filtering methods, and recommendation models based on clustering. Section 3 describes our proposed model to extend current factorized methods in collaborative filtering. In Section 4 we describe our empirical experimental results. Finally, we summarize the findings of the evaluation and define future work directions in Section 5.

2 Previous works

Clustering is the task of grouping a set of instances in a way that instances in the same cluster, are more similar to each other than to those in other clusters. Several clustering methods has been proposed such as expectation maximization(EM) (Bishop 2006), hierarchical clustering (Witten and Frank 2005), density-based clustering (Witten and Frank 2005), etc.

Recommendation systems are key parts of the information and e-commerce ecosystem (Ekstrand et al. 2011), where businesses offer an enormous number of items through different channels such as World Wide Web (WWW), Smart TVs, Digital Screens, etc. Therefore, users can have trouble finding the products that they are looking for. Recommendation systems address this problem by providing powerful methods which enable users to filter through large information and product space based on their preferences.

The fundamental assumption behind Collaborative Filtering(CF) is if users agree about the relevance of some items, then they will likely agree about other items. For instance, if a group of users likes the same things as Mary, then Mary is likely to like the things they like which she has not seen yet (Ekstrand et al. 2011). 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 are accompanied by a rating matrix $R = [r_{ui}]_{n \times m}$ where r_{ui} represents the rating of user u on item i. Collaborative filtering consists of predicting unknown ratings based on the known ratings inside the rating matrix R.

Recommendation accuracy can be defined as the performance of a recommendation system on producing helpful recommendations for users. There are several evaluation functions to measure this accuracy. Rating prediction accuracy, which is usually evaluated by root mean squared error (RMSE), is a popular way to measure a recommendation system's accuracy (Koren 2008). This metric compare the outcome rating predictions of a recommendation system with the ratings actually observed. Clearly, smaller differences between the predicted ratings and the actual ratings are more desired.

RMSE is defined as the square root of the mean of the square of all of the error. RMSE is perhaps the most popular metric that have been used in evaluating accuracy of predicted ratings. For a given testing set T, the recommendation system generates predicted ratings $\hat{r}_{ui}$ for testing cases $< u, i >$ where the true ratings r_{ui} are known. RMSE is then computed as follows:

$$RMSE = \sqrt{\frac{1}{|T|} . \sum_{<u,i> \in T} (r_{ui} - \hat{r}_{ui})^2} \quad (1)$$

A smaller RMSE means a higher rating prediction accuracy for a recommendation system.

To predict these unknown ratings in R, matrix Factorization (MF) techniques can be employed to decompose R into two lower dimension matrices Q and P, which contain corresponding latent vectors of each user and item in the length of $l \ll m, n$ (Fig. 1.a). Such a model is based

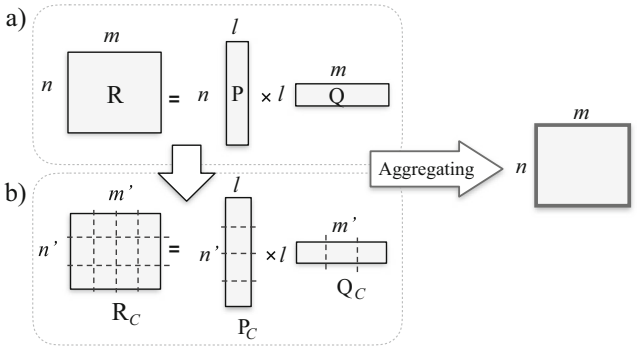

Fig. 1 Aggregation of the two level of predictions: a) Factorizing rating matrix R into latent matrices P and Q. b) Clustering P and Q, and generating cluster-level rating matrix R_C. n, m, and l are the number of users, items and latent vectors respectively. n' and m' also represent the number of clusters for users and items

on the singular value decomposition (SVD) technique for finding latent vectors in information retrieval. SVD does not work for sparse rating matrices which contain unknown ratings (missing values). Thus, techniques based on MF have been proposed to factorize rating matrix R solely based on its partial known ratings.

To find proper P and Q, a learning algorithm can be started by a random initialization of these matrices. In every learning step, it then tries to change the initialized variables in a way that $Q^T P$ converges to the known values of R. In the prediction case, the product of learned matrices will be used to predict the unknown ratings. MF techniques characterize every user and item by corresponding them a latent vector. Assume $q_i \in \mathbb{R}^l$ as the corresponding vector of item i and $p_u \in \mathbb{R}^l$ as the corresponding vector of user u. It is supposed that the dot product of these vectors result in r_{ui}. Also, a bias value is typically corresponded to each user and item to reflect their mean ratings. Adding these biases, the rating prediction function will change to:

$$\hat{r}_{ui} = q_i^T p_u + b_{ui} \tag{2}$$

where

$$b_{ui} = b_u + b_i$$

b_u represents the mean rating for user u and b_i represents the mean rating for item i. In practice, we have popular items with high ratings and also generous users who always rate the items higher than normal. Thus, these two variables reflect these biases. Consequently, the found latent vectors for users and items represent the pure interests of users on items once we remove these biases from the ratings. This method is called biased matrix factorization (BMF) (Koren et al. 2009) technique in recommendation systems.

Let's define the error as the actual rating minus the predicted value in each step, e_{ui}. In the learning phase, the goal will be minimization of root mean square error. Regularizer values prevent over-fitting on the model and keep the latent values small. The final minimization function is as follows:

$$min \left(\sum_{(u,i)\in R} (r_{ui} - q_i^T p_u - b_{ui})^2 + \sum_u (\lambda_1 . \|p_u\|^2 + \lambda_2 . \|b_u\|^2) + \sum_i (\lambda_1 . \|q_i\|^2 + \lambda_2 . \|b_i\|^2) \right) \tag{3}$$

where $\|.\|$ indicates the Euclidean norm of the vectors. These latent vectors can be learned using the stochastic gradient descent technique as is described in Koren et al. (2009).

Employing clustering to categorize collaborative information, and using these clusters to predict the unknown preferences has been employed before in a number of works (Xu et al. 2012; Gueye et al. 2011). Gueye et al. (2011) try to add the effect of clusters into a biased matrix factorization technique. The predictor function that they use is as follows:

$$\hat{r}_{ui} = q_i^T . p_u + \mu_{C_{G(i)}} + b_{u,C_{G(i)}} + b_i$$

where $C_{G(i)}$ returns the group that item i belongs to, $\mu_{C_{G(i)}}$ is the average ratings in $C_{G(i)}$, and $b_{u,C_{G(i)}}$ is the bias of user u for the group of items C_{G_i}. George and Merugu (2005) also use the same technique to improve the rating prediction accuracy employing the rating averages for users, items, user-clusters, and item-clusters.

In addition, Xu et al. (2012) employ the clusters in a different way to improve the accuracy of rating predictions. They cluster items and users into subgroups where each user or item can belong to more than one cluster. Their main idea is to apply number of collaborative filtering algorithm in each subgroup and then merge the prediction results together. However, they have to consider the clusters reasonably large to possess enough known ratings in each subgroup for the learning process. This restriction causes their prediction model to ignore the fine relations among users and items in smaller clusters (As is discussed in Section 3.3). Desrosiers et al. in Desrosiers and Karypis (2011) present a complete survey on neighbourhood-based recommendation methods that covers many other extended matrix factorization techniques for recommendation.

Jamali et al. (2011) and Beutel et al. (2014) propose promising probabilistic models to co-cluster users and items and then consider their cluster-level preferences to improve the rating prediction accuracy. However, because of the expensive nature of probabilistic modelling they have to employ heuristics to reduce both the learning time and the consumed memory.

Töscher et al. (2008) present a neighbourhood-aware matrix factorization technique, which includes neighbourhood

information in a matrix factorization technique. Their proposed algorithm computes three predictions for every user-item pair: a prediction function r_{ui}^{MF} based on matrix factorization technique; a prediction function r_{ui}^{user} based on an user-neighbourhood model; and finally a prediction r_{ui}^{item}, which is based on an item-neighbourhood model. A combination of these three predictions is defined as the final rating prediction of this algorithm. The rating prediction r_{ui}^{user} is computed as follows:

$$r_{ui}^{user} = \frac{\sum_{v \in U_J(u)} c_{uv}^{user} r_{vi}}{\sum_{v \in U_J(u)} c_{uv}^{user}}$$

where $U_J(u)$ denotes the set of J users with highest correlation to user u. These correlations are reached by counting the number of co-rating of users (co-being-rated of items). The paper has a similar approach in computing r_{ui}^{item}. It is mentioned by Töscher et al. (2008) that this neighbourhood-aware method improves the rating prediction accuracy. However, it is still sensitive to the choice of J. By increasing J the training time will be increased accordingly.

Koren in (2010) presents a neighbourhood aware recommendation model named "Non-Factorized Asymmetric SVD". It includes the following prediction function inside its recommendation model:

$$\begin{aligned}\hat{r}_{ui} &= \mu + b_u + b_i + |R(u)|^{-\frac{1}{2}} \sum_{j \in R(u)} (\hat{r}_{uj} - b_{uj}) w_{ij} \\ &\quad + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} c_{ij} \end{aligned} \tag{4}$$

where the weight from item j to item i is denoted by w_{ij}, and c_{ij} is an added offset between these two items.

As a new method, he then models these w_{ij} and c_{ij} in a factorized approach (Koren 2010). He factors item-item relationships by associating each item i with three vectors: $q_i, x_i, y_i \in \mathbb{R}^l$. It represents the similarity weight between items as $q_i^T x_i$, and $q_i^T y_j$ to represent c_{ij}. The proposed factorized item-item model is as follows:

$$\begin{aligned}\hat{r}_{ui} &= \mu + b_u + b_i + |R(u)|^{-\frac{1}{2}} \sum_{j \in R(u)} (\hat{r}_{uj} - b_{uj}) q_i^T x_i \\ &\quad + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} q_i^T y_j \end{aligned} \tag{5}$$

where $R(u)$ contains all the items which have been rated by user u, and $N(u)$ contains all items for which u provided an implicit preference. He does the same modelling for the users, and then proposes a fusion of item-item and user-user models in a single model. The integrated model uses the following predictor function:

$$\begin{aligned}\hat{r}_{ui} &= \mu + b_u + b_i + |R(u)|^{-\frac{1}{2}} \sum_{j \in R(u)} (\hat{r}_{uj} - b_{uj}) q_i^T x_i \\ &\quad + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} q_i^T y_j + |R(i)|^{-\frac{1}{2}} \sum_{v \in R(i)} (\hat{r}_{vi} - b_{vi}) p_u^T z_v \end{aligned} \tag{6}$$

where $|R(i)|$ contains all the users that have rated item i, and z_v represents the implicit feedback of user v. There are larger numbers of users compared to items in common recommendation systems scenarios. Thus, user-user methods are much more computationally expensive compared to item-item models in practice. Several advantages of the item-item models have been compared with user-user, and integrated models in (Koren 2010).

These neighbourhood-aware factorization techniques result in a promising rating prediction accuracy in practice (Koren 2010). However, they suffer from a long computation time. This is because, a larger neighbourhood size ($|N(u)|$, $|R(u)|$) needs to be employed in these models in order to achieve a fine rating prediction accuracy. Employing a larger number of neighbourhood size in Eqs. 6, 4 and 5 increases their complexity and training time. In addition, all these techniques achieve a poor recommendation accuracy for users with low rating profiles, called *cold-start* users. For these reasons, in this paper we leverage the advantages of clustering methods beside the factorization techniques to improve rating prediction accuracy without adding much complexity.

Cold-starts are users who newly enter into a recommendation system and the system does not know much about their preferences. Consequently, the system is unable to present any valuable personalized recommendations to them Rashid et al. (2008). Cold-start users are one of the major challenges in recommendation systems (Konstan and Riedl 2012; Rashid et al. 2008). Utilizing the neighbourhood information is a common solution to improve recommendation accuracy for cold-start users. In our experiment (Section 4.3), we show that adopting our proposed cluster-level rating predictions improves the rating prediction accuracy for cold-start users as well in spite of their few known ratings.

Note that all these methods are learned with respect to improving rating prediction accuracy. In recent years, multiple methods have been proposed which are learned to improve personalized item ranking for users (Ning and Karypis 2012; Rendle et al. 2009; Steck 2010b). However, these methods achieve low accuracy of rating prediction (which usually is presented by RMSE metric), they

result in significant improvements for top-N recommendation accuracy. We do not expect that our model can compete with those models regarding top-N recommendation tasks as we learn our model with respect to different criteria. However, we have recently published (Mirbakhsh and Ling 2015) a clustering-based ranking model which employ the same clustering-based wrapper model and outperforms those models regarding top-N recommendation tasks.

3 The proposed models

Employing clustering is a classic approach for dividing the set of users and items into different categories and making similarity-based recommendations for each of these categories. However, those methods mainly focus on the individual clusters and ignore the shared interest between the clusters. It is highly valuable to know how users with the same tastes have shown their interests on a set of similar items. For instance, it is important to know if younger users are interested in movies in drama genre or comedies. Thus, as soon as we guess that user u belongs to the cluster of younger users we can employ these cluster-level interests to predict user u's rating on movies in these two genre. Using the shared preferences of the categories has two advantages over the common neighbourhood models:

1. It generalizes the users' interests and the items' features regarding the clusters that they belong to. For example, user u may belong to the cluster 'Adults' and item i may belong to the cluster 'Cartoons'. Thus, in addition to considering if user u is interested in item i, it deliberates if the cluster 'Adults' shares any preferences with the cluster 'Cartoons'.
2. It considers deeper similarities. Item-item models check if user u is interested in item i and its similar items. User-user models also check if user u and its similar users are interested in item i. In our approach, we check to see if user u and its similar users are interested in item i and its similar items.

Applying clustering methods on rating matrix R to cluster users and items is extremely costly because of the large number of users and items. In this paper, we first apply a biased matrix factorization technique on the known ratings to produce latent matrices P and Q (Fig. 1a). K-means is then applied to these matrices to generate cluster of items and users (Fig. 1b). Rating matrices are typically sparse in the recommendation systems context. Hence, using latent vectors helps to reduce the complexity of clustering these large and sparse datasets because 1) there is no sparsity in these latent vectors, 2) they are in a much lower dimensionality. Our results (Section 4.2) show clearly that employing these clusters in better qualities (better clustering method) and more quantities (multiple clustering) significantly improve the rating prediction accuracy (Section 4.1).

We consider the shared ratings between these produced clusters in a "coarse" matrix R_C. In this new rating matrix, every r_{C_u,C_i} represents the average rating of the users inside the user cluster C_u on the items inside the item cluster C_i, as follows:

$$R_C = \begin{array}{c} \\ C_{u_1} \\ C_{u_2} \\ \vdots \\ C_{u_{n'}} \end{array} \begin{array}{c} \begin{array}{cccc} C_{i_1} & C_{i_2} & \dots & C_{i_{m'}} \end{array} \\ \begin{pmatrix} r_{11} & r_{12} & \dots & r_{1m'} \\ r_{21} & r_{22} & \dots & r_{2m'} \\ \vdots & \vdots & \ddots & \vdots \\ r_{n'1} & r_{n'2} & \dots & r_{n'm'} \end{pmatrix} \end{array}$$

where $n' < n$ and $m' < m$ are the number of clusters for users and items respectively (Fig. 1b). Because of the small number of known ratings, r_{C_u,C_i} is not accurate enough for all pairs of clusters. To resolve this issue, we consider r_{C_u,C_i} as observed, if we have enough observed ratings between clusters C_u and C_i. After calculating the cluster-level known ratings in R_C, any collaborative filtering methods such as a matrix factorization technique can be applied on this coarse matrix to predict the unknown ratings in R_C.

Eventually, we employ the cluster-level rating predictions from R_C to extend a number of collaborative filtering methods. In the following sub sections, we employ R_C to improve the rating predictions of biased matrix factorization (Koren et al. 2009) technique in Section 3.1, and Asymmetric SVD (Koren and Bell 2011) in Section 3.2. Our extension approach is simple to implement and can be adopted to extend many other collaborative filtering algorithms.

3.1 Clustering-based matrix factorization

In this method we employ a biased matrix factorization technique (Koren et al. 2009) to predict the unknown cluster-level ratings in R_C. We consider each cluster as an individual and correspond each of these individuals a latent vector of length l and a bias value. Thus, we can factorize coarse matrix R_C into two latent matrices Q_C and P_C, and the predictor function for r_{C_u,C_i} will be as follows:

$$r_{C_u,C_i} = q_{C_i}^T p_{C_u} + b_{C_i} + b_{C_u} \tag{7}$$

C_i is the item cluster that item i belongs to, C_u is the user cluster that user u belongs to, and q_{C_i} and p_{C_u} are the corresponding latent vectors of these two clusters. Thus, instead of predicting a rating for pairs of users and items, it predicts a rating for their clusters. This helps to reduce the sparsity of the rating matrix R. Although the cluster-level

ratings are too general to be employed solely for the prediction purpose. Thus, we use a fusion of the traditional rating predictions from rating matrix R and our cluster-level rating predictions from matrix R_C as follows:

$$r_{ui} = T_\alpha(q_i, q_{C_i})^T T_\alpha(p_u, p_{C_u}) + T_\beta(b_u, b_{C_u}) + T_\beta(b_i, b_{C_i}) \tag{8}$$

where T_α is a trade-off function defined as follows:

$$T_\alpha(x, y) = (1-\alpha).x + \alpha.y \tag{9}$$

$0 \leq \alpha \leq 1$ controls the weights between the traditional rating predictions and the cluster-levels rating predictions (same for β and T_β). We name this fusion form Clustering-Based Matrix Factorization (CBMF) technique in our experimental results.

3.2 Integrating the clusters with various collaborative filtering methods

As is mentioned in Section 1, our extension approach is easy to implement and can be applied in most collaborative filtering methods. There are two common approaches in neighbourhood aware matrix factorization techniques: 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. In the literature (Töscher et al. 2008; Koren 2010), an integration of both item-item and user-user models sometimes has been applied in the final prediction function to achieve a better rating prediction accuracy. However, item-item models are usually preferable as they have less space and time complexity. This is because of the typically larger number of users in recommendation systems. Neighborhood aware models need to compute all pairwise similarities between items or users, and its complexity grows quadratically with the input size (Koren 2010). Koren in (2008, 2010) solves this limitation by factoring the neighbourhood model, which scales both item-item and user-user implementations linearly with the size of the data (Koren 2010). He defines $N(u)$ and $R(u)$ as the sets of all implicit and explicit feedback from user u. However, $N(u)$ is assumed equal to $R(u)$ in his experiment on the Netflix dataset (Koren 2008).

In SVD++, Koren corresponds each item i with two latent vectors $q_i, y_i \in \mathbb{R}^l$, and models each user with a corresponding latent vector $p_u \in \mathbb{R}^l$ plus the sum of the latent vectors of all the items inside $N(u)$. He defines the SVD++'s prediction function as follows:

$$r_{ui} = q_i^T \left(p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j \right) + b_{ui} \tag{10}$$

In the Asymmetric SVD, Koren employs the explicit feedback, $R(u)$, into the SVD++ model and proposes a factorized item-item model as follow:

$$r_{ui} = q_i^T \left(p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j + |R(u)|^{-\frac{1}{2}} \sum_{j \in R(u)} (r_{uj} - b_{uj}) x_j \right) + b_{ui} \tag{11}$$

Thus, in this model each user is modelled with a latent vector $p_u \in \mathbb{R}^l$, normalized summation of the implicit latent vectors ($|N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j$), and normalized summation of the explicit latent vectors times their weights ($|R(u)|^{-\frac{1}{2}} \sum_{j \in R(u)} (r_{uj} - b_{uj}) x_j$).

We leverage our generated cluster-level ratings in matrix R_C to extend Asymmetric SVD technique. Similarly, we assign three latent vectors $q_{C_i}, y_{C_i}, x_{C_i} \in \mathbb{R}^l$ to each cluster of items, and a latent vector $p_{C_u} \in \mathbb{R}^l$ to each cluster of users. The new predictor function is as follow:

$$\begin{aligned} r_{ui} = {} & T_\alpha(q_i, q_{C_i})^T \left(T_\alpha(p_u, p_{C_u}) + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} T_\alpha(y_j, y_{C_j}) \right. \\ & \left. + |R(u)|^{-\frac{1}{2}} \sum_{j \in R(u)} (r_{uj} - b_u - b_j) T_\alpha(x_j, x_{C_j}) \right) \\ & + T_\beta(b_u, b_{C_u}) + T_\beta(b_i, b_{C_i}) \end{aligned} \tag{12}$$

We call this model CB-AsySVD.

3.3 Employing more clusters

Clustering items and users in different sizes of clusters may results in a variety of informative clusters. For instance, clustering movies into 10 clusters would capture more general similarities among items such as genre. Although, clustering the movies into 500 clusters would distinguish movies based on slight differences such as the year of production. Koren in Koren et al. (2009) shows that employing more parameters in the prediction model will improve rating prediction accuracy. Thus, we expect to improve rating prediction accuracy by employing more clusters in different sizes. In Section 4.4.2, we show that employing more clusters will improve the prediction accuracy of our proposed CBMF model further. However, adding more clusters will increases the complexity of this model.

3.3.1 Learning the parameters

In our both proposed CBMF and CB-AsySVD models, the parameters are determined by minimizing the associated regularized squared error function through gradient descent.

Algorithm 1 presents the learning process of our proposed CBMF model.

Algorithm 1 Our proposed CBMF's updating formula

% Parameters are randomly initialised.
for limited number of epochs **do**
  for all known $r_{ui} \in R$ **do**
    $\hat{q}_i \leftarrow T_\alpha(q_i, q_{C_i})$
    $\hat{p}_u \leftarrow T_\alpha(p_u, p_{C_u})$
    $\hat{b}_{ui} \leftarrow T_\beta(b_{ui}, b_{C_u C_i})$
    $\hat{r}_{ui} \leftarrow \hat{b}_{ui} + \hat{q}_i^T \hat{p}_u$
    $e_{ui} \leftarrow r_{ui} - \hat{r}_{ui}$
    % Perform gradient step:
    $q_i \leftarrow q_i + \gamma_1(e_{ui}.\hat{p}_u - \lambda_1.q_i)$
    $p_u \leftarrow p_u + \gamma_1(e_{ui}.\hat{q}_i - \lambda_1.p_u)$
    $q_{C_i} \leftarrow q_{C_i} + \gamma_2(e_{ui}.\hat{p}_u - \lambda_2.q_{C_i})$
    $p_{C_u} \leftarrow p_{C_u} + \gamma_2(e_{ui}.\hat{q}_i - \lambda_2.p_{C_u})$
    $b_i \leftarrow b_i + \gamma_3(e_{ui} - \lambda_3.b_i)$
    $b_u \leftarrow b_u + \gamma_3(e_{ui} - \lambda_3.b_u)$
    $b_{C_i} \leftarrow b_{C_i} + \gamma_4(e_{ui} - \lambda_4.b_{C_i})$
    $b_{C_u} \leftarrow b_{C_u} + \gamma_4(e_{ui} - \lambda_4.b_{C_u})$
  end for
end for

4 Experiment results

In this section, we provide a discussion of our experiment to validate our hypothesis about these extensions. The MovieLens100k dataset was collected by the GroupLens Research Project at the University of Minnesota. It contains 100,000 ratings from 943 users on 1,682 movies where each user has rated at least 20 movies (Herlocker et al. 1999). This package includes five random splits of dataset into training and test sets (80 % for the training set and 20 % for the test set). We employ these provided training and test sets (u1, u2, .., u5) in our evaluation. The Netflix dataset contains over 100 million ratings from 480,189 users who have rated 17,770 movies. In both datasets, ratings are in a range of $[1, 5]$. Both datasets are very sparse as we know only 1 % of ratings and 99 % of ratings are unknown. We run each algorithm five times on these datasets to remove the effect of random initialization of latent vectors on the predictions. Thus, our reported results are the averaged result of these five runs.

In the following subsections, we first describe our clustering process in Section 4.1. We then employ these found clusters in our proposed methods and evaluate their rating prediction accuracy (Section 4.2). In Section 4.3, we evaluate the impact of our proposed clustering-based methods in rating prediction accuracy for cold-start users. Finally, in Section 4.4, we present additional experiments that evaluates the impact of the quality and the quantity of clusters on improving rating prediction accuracy.

4.1 Clustering users and items

The rating matrix R is usually extremely large and sparse. Thus, clustering this matrix directly is costly and less effective. For this reason, we cluster the latent vectors instead, as their lower dimensionality and higher density. We start by applying a biased matrix factorization technique on both datasets to find their latent vectors (matrices P and Q). These latent vectors (learned on the train sets) are then used for the clustering purpose. Figure 2 illustrates a demonstration of 12 movies in MovieLens dataset based on their generated latent vectors ($l = 2$). As expected, similar items end up with similar latent vectors after the factorization process. For instance, in Fig. 2 multiple cartoons including 'Cinderella' and 'Toy Story' have closer latent vectors compare to other movies in other genre.

We try four different clustering methods, and as shown in Section 4.4.1, K-Means results in the best prediction accuracy in both datasets. Expectation maximization achieves almost the same rating prediction accuracy but with much more complexity.

We try different sizes of clusters on the proposed models, the number of clusters that achieves smallest RMSE on the validation set will be selected as the best choices of m', and n' (Table 3). Eventually, $n' = m' = 100$ is selected for the MovieLens100k dataset. Also, $n' = 1000$ and $m' = 500$ are selected for the Netflix dataset. Based on our experiment,

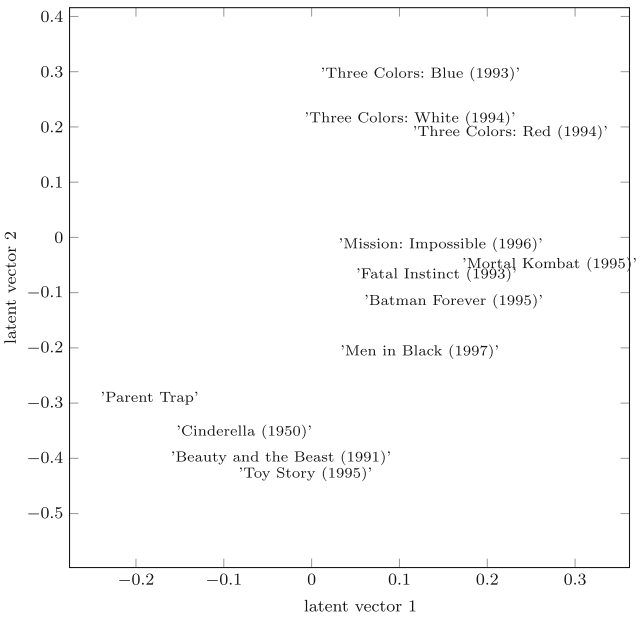

Fig. 2 A demonstration of 12 movies in the MovieLens dataset based on their generated latent vectors. The rating matrix R is factorized into matrices P and Q with only two latent vectors ($l = 2$)

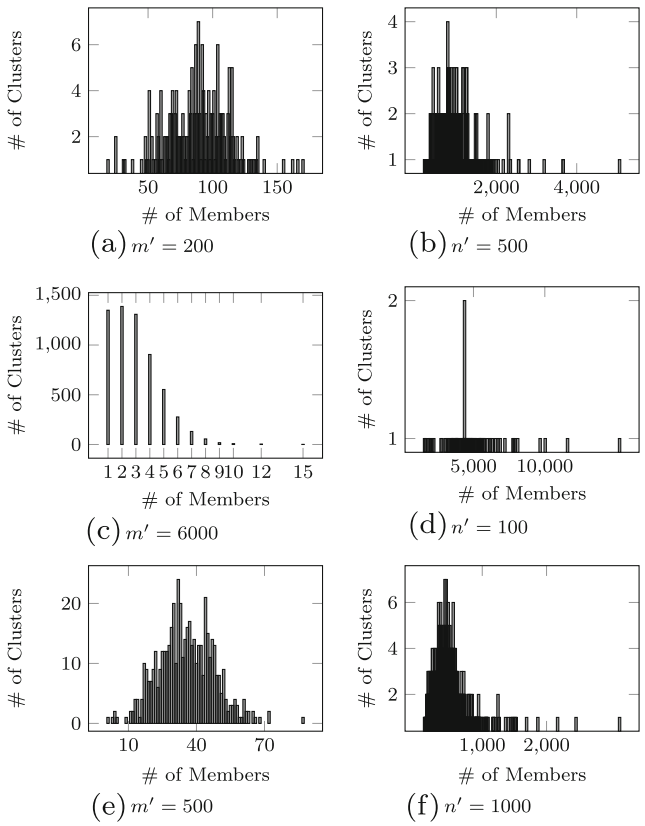

Fig. 3 Distribution of clusters with different number of members in the Netflix dataset

the clusters should not be too broad or too small. Figure 3 illustrates the distribution of clusters of items and users with different numbers of members in the Netflix dataset. It seems our final selected number of clusters (Fig. 3e and f) achieves a normal-like distribution with fewer too large and too small clusters.

Table 1 shows the movies inside a number of found clusters in the Netflix dataset. As shown, it seems that movies in the same genre and almost similar years of production tend to be in the same clusters. For instance, 'cluster 295' includes different versions of 'Lord of the Rings' and 'Star Wars' movies accompanied by a number of other movies in 'Adventure' and 'Fantasy' genres[1] such as 'The Matrix'. 'cluster 344' contains different versions of 'X-Men', 'Spider Man' and 'Harry Potter' movies. 'cluster 420' includes a number of documentaries and live concerts. As no information about the users is provided, the clusters of users cannot be judged. Note that the movies' names (identities) are not used anywhere in this experiment; these names are only employed for a better demonstration of the clusters.

Finding a reasonable number of clusters for each dataset has been always a subject of debate (Balijepally et al. 2011). Obviously the clusters are not perfect and there are always items that are wrongly clustered. Evaluating the clusters is subjective and depends on the view of the evaluator. However, based on the author's point of view at least two thirds of the clustered items seem meaningful.

4.2 Rating prediction accuracy

As mentioned earlier, we integrate the traditional rating predictions from rating matrix R and our proposed cluster-level rating predictions from coarse matrix R_C as our final rating predictions. Two variables α and β control the influence of the rating predictions from R and cluster-level rating predictions from R_C. More specifically, α manages the influence of cluster-level interests and β manages the influence of cluster-level biases. For $\alpha = \beta = 0.0$, the extended models (CBMF and CB-AsySVD) consider the traditional rating predictions solely. Thus, the resulting RMSE is similar to the accuracy of the original models (BMF and AsySVD) as expected. A validation set is used to determine the best selection of these variables in our experiment by testing different values for α and β. Table 2 illustrates the RMSE result of CBMF model by selecting different combinations of α and β in the MovieLens dataset. $\alpha = 0.5$ and $\beta = 0.8$ is selected as the optimum values in our experiment for this dataset (Table 3).

Figure 4 illustrates a comparison between the RMSE results of the biased matrix factorization technique (Koren et al. 2009) and our proposed clustering-based matrix factorization technique for different selections of l. As is shown, CBMF outperforms the BMF even when BMF employs higher l. Additionally, Fig. 5 shows a comparison between the accuracy of four techniques: biased matrix factorization (BMF) (Koren et al. 2009), Asymmetric SVD (AsySVD) (Koren 2008), and our clustering-based extensions of them. It shows that CBMF achieves a smaller RMSE compared with the non-extended neighbourhood-aware model (AsySVD) when this model does not employ enough large numbers of neighbours. Neighbourhood-aware models are usually sensitive to the selection of the neighbourhood size. By employing larger neighbourhood size, the RMSE decreases. However, this practice results in a higher complexity and a lower comprehensibility for these models. Note that we could not show the error bars in this figure because the overlaps of the error bars between CBMF and both AsySVD and CB-AsySVD techniques.

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

Table 1 The movies inside a number of formed clusters in the Netflix dataset

Cluster ID	Movie Title (Production Year)
cluster295	Lord of the Rings: The Return of the King: Extended Edition: Bonus Material (2003), Lord of the Rings: The Two Towers: Extended Edition (2002), Firefly (2002), Raiders of the Lost Ark (1981), Star Wars: Clone Wars: Vol. 1 (2004), Star Wars: Episode VI: Return of the Jedi (1983), Lord of the Rings: The Two Towers: Bonus Material (2002), Lost: Season 1 (2004), Aliens: Collector's Edition: Bonus Material (1986), Batman Begins (2005), Lord of the Rings: The Two Towers (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), Battlestar Galactica: The Miniseries (2003), Crouching Tiger (2000), Star Wars: Episode IV: A New Hope (1977), Lord of the Rings: The Fellowship of the Ring (2001), Band of Brothers (2001), Battlestar Galactica: Season 1 (2004), Star Wars: Episode V: The Empire Strikes Back (1980), The Matrix (1999), The Indiana Jones Trilogy: Bonus Material (2003), Lord of the Rings: The Return of the King: Extended Edition (2003), The Lord of the Rings: The Fellowship of the Ring: Extended Edition (2001), Indiana Jones and the Last Crusade (1989), Lord of the Rings: The Return of the King (2003), Star Wars Trilogy: Bonus Material (2004)
cluster344	Pirates of the Caribbean: The Curse of the Black Pearl: Bonus Material (2003), X2: X-Men United: Bonus Material (2003), X-Men (2000), Harry Potter and the Prisoner of Azkaban (2004), Harry Potter and the Sorcerer's Stone (2001), Monsters (2001), Harry Potter and the Chamber of Secrets (2002), X2: X-Men United (2003), The Incredibles: Bonus Material (2004), Harry Potter and the Prisoner of Azkaban: Bonus Material (2004), Spider-Man 2 (2004), X-Men: Bonus Material (2000), Beauty and the Beast: Special Edition: Bonus Material (1991), Harry Potter and the Chamber of Secrets: Bonus Material (2002), Harry Potter and the Sorcerer's Stone: Bonus Material (2001), Farscape: The Peacekeeper Wars: Bonus Material (2004), Spider-Man (2002), Pirates of the Caribbean: The Curse of the Black Pearl (2003)
cluster420	ABC Primetime: Mel Gibson's The Passion of the Christ (2004), Jay Jay the Jet Plane: Good Friends Forever (2003), Lassie: The 50th TV Anniversary Collector's Edition (1954), Jesus and His Times (2000), Jesus of Nazareth (1977), Wiseguy: Between the Mob and a Hard Place (1989), Jesus and the Shroud of Turin (1999), The Bible Collection: Moses (1996), National Geographic: Inside American Power: The White House (1996), Barney: Come on Over to Barney's House (2000), Chicago: AE Live by Request (2003), Love Comes Softly (2003), The Commish: Season 1 (1991), Oliver Twist (1999), The Ultimate National Geographic WWII Collection (2004), Twin Towers (2003), The Gospel of John (2003), Hans Brinker (1962), Lorna Doone (2000), The Miracle Maker: The Story of Jesus (2000), Jeremiah: The Bible (1998), Thomas and Friends: Calling All Engines (2005), The Bible Collection: Jacob (1994), New York Firefighters: The Brotherhood of 9/11 (2002), Parineeta (2005), Genesis: The Way We Walk: Live in Concert (2001), Billy Joel: The Essential Video Collection (2001), National Geographic: Ambassador: Inside the Embassy (2002), Joshua (2002), The Barkleys of Broadway (1949), Michael W. Smith: Live in Concert: A 20 Year Celebration (2004), National Geographic: Inside American Power: Air Force One (2001), Bear in the Big Blue House Live! (2002), Piano Grand: A Smithsonian Celebration (2000), Samantha: An American Girl Holiday (2004), Chris Botti and Friends: Night Sessions: Live in Concert (2002), Denise Austin: Personal Training System (2004), Walt: The Man Behind the Myth (2001), The Story of Jesus for Children (1979), Joseph: King of Dreams (2000), In Old Chicago (1937), Visions of Greece (2002)

The extension models have almost the same complexity as the non-extended models. However, they add a pre-processing complexity in the clustering step. For instance, the training time of the extended models are less than twice of the non-extended models in the MovieLens100k dataset. Clustering was also not considerably time consuming because we perform the clustering on the low dimension latent vectors. For instance, the clustering of users in the

Table 2 Resulting RMSE by applying CBMF with different values for α and β on a validation set in MovieLens dataset

α / β	$\beta = 0$	$\beta = 0.2$	$\beta = 0.4$	$\beta = 0.6$	$\beta = 0.8$	$\beta = 1$
$\alpha = 0$	0.922	0.921	0.921	0.920	0.919	0.931
$\alpha = 0.2$	0.914	0.914	0.913	0.912	0.911	0.921
$\alpha = 0.4$	0.913	0.913	0.912	0.911	0.909	0.920
$\alpha = 0.6$	0.914	0.914	0.913	0.912	0.911	0.922
$\alpha = 0.8$	0.918	0.917	0.917	0.916	0.918	0.933
$\alpha = 1$	0.927	0.927	0.927	0.928	0.931	1.072

Table 3 Resulting RMSE by applying CBMF with different numbers of item clusters, m', and different numbers of user clusters, n', on a validation set in MovieLens dataset

n' / m'	m' =10	m' =50	m' =100	m' =150	m' =200
n' =10	0.9140	0.9115	0.9106	0.9113	0.9114
n' =50	0.9140	0.9107	0.9105	0.9109	0.9110
n' =100	0.9135	0.9101	0.9097	0.9101	0.9104
n' =150	0.9112	0.9110	0.9109	0.9112	0.9145
n' =200	0.9144	0.9113	0.9115	0.9112	0.9116

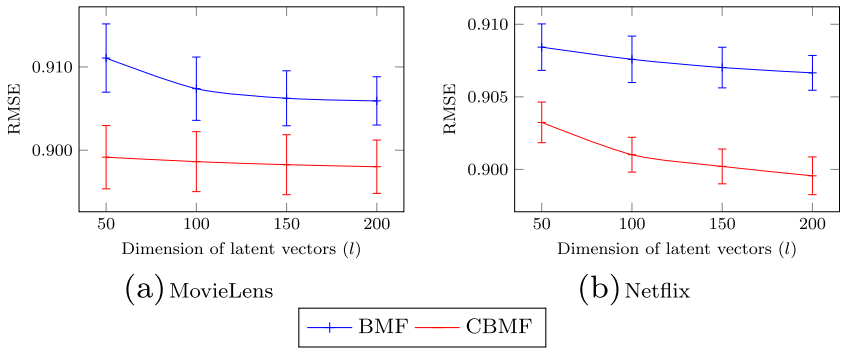

Fig. 4 A comparison between BMF technique and our proposed CBMF technique for different selection of l (dimension of latent vectors)

Netflix dataset takes less than a hour using the Rapidminer software in our PC with 3.30 GHz CPU. Thus, the extended models maintain the scalability of those models.[2]

We apply a t-test to ensure the significance of our results. In the MovieLens dataset, we train the extended and non-extended models five times with different initialized parameters and test it for this dataset's five standard validation sets (u1, u2, .. , u5). Thus, we have 25 results in total for each trained model. Our student t-test for two independent RMSE means between our proposed cluster-based matrix factorization (CBMF) technique and biased matrix factorization (BMF) technique shows a significant difference at $p < 0.01$. In this one-tailed t-test the t-value is 2.888078 and the p-value is 0.003183. For the Netflix dataset, we have the same significant results where the t-value is 12.35604 and the p-value is 0.00001. The employed parameters in the learning process of Algorithm 1 are listed in Appendix.

4.3 Cold-start users

Using the neighbourhood information is a common solution to improve recommendation accuracy for cold-start users. However, even those models have to employ the sparse rating matrix R to find the similar users and items to leverage their feedback. Our produced coarse matrix R_C reduces the sparsity of rating matrix R using all those found similarities between users and items. Thus, R_c can more effectively generalize any minor feedback from cold-start users to improve their related recommendation accuracy.

For example, assume that a cold-start user u has rated movie "Lord of The Rings". Using neighbourhood information we can only find out that user u may like similar movies such as "Star Wars". However, many similarities between movies and users will be ignored because of the sparsity of rating matrix R. On the other hand, our clustering level modelling of movies adds much information into the prediction model such as the higher level similarities between movies including their abstract genre, etc. Thus, our R_C can extract more information from any minor feedback from cold-start user u. Once these clusters are formed, known ratings in the clusters can help to patch the unknown ratings of cold-start users.

Cold-start users are examined separately in our experiment to see if our extension models improve the accuracy of their recommendations. Figure 6 illustrates the RMSE results of our proposed model for cold start users in both datasets. The horizontal axis represents the increase of the known ratings for users. As is shown, our extended models effectively improve the prediction accuracy for this set of users with few known ratings. As is shown in Fig. 6a and b, all extended models improve the RMSE of cold-start users in both datasets. For instance, in the MovieLens dataset, CBMF model results in an almost 0.97 of RMSE for user with fewer than 25 known ratings, which is as good as the RMSE result of BMF model for users who have fewer than 50 known ratings.

4.4 Sub-experiments

In this section we explain two sub experiments; We first try four different clustering algorithms on our proposed model to examine the influence of the clusters' quality on our proposed model prediction accuracy (Section 4.4.1). Using this experiment we also select the clustering algorithm that achieve the best prediction accuracy as our default clustering method. We then vary the number of clusters in the selected clustering algorithm to create clusters in different sizes. In Section 4.4.2, we show that adding more clusters in variety of sizes will improve the rating prediction accuracy of our proposed CBMF model further. The following experiments also evaluate the impact of the quality and the quantity of clusters on improving rating prediction accuracy.

4.4.1 Different clustering methods

We employ four different clustering methods to examine the effect of clustering quality on our proposed model's rating

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

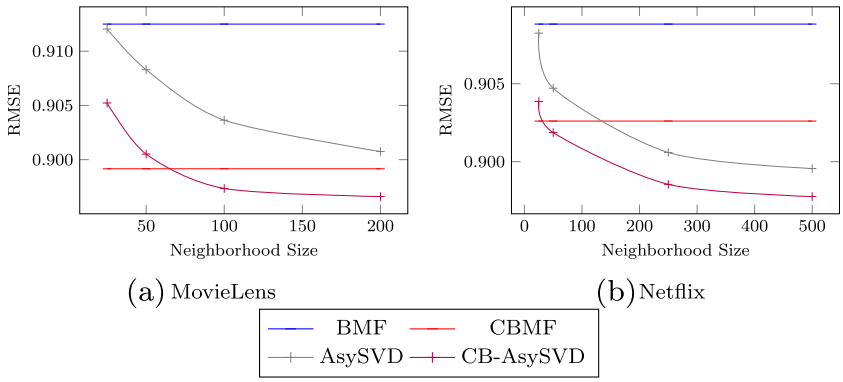

Fig. 5 The accuracy of the proposed clustering-based models applying on the two datasets ($l = 50$ is used to achieve these results)

prediction results. We apply following clustering methods on both datasets:

- **K-Means:** It is based on partitioning data into K clusters that each data point belongs to the cluster with the nearest mean (Bishop 2006). This is a well-known clustering method that achieves a high quality of clusters.
- **Expectation Maximization (EM):** It is an expensive but high quality clustering algorithm (Bishop 2006). EM models the data points with a fixed number of Gaussian distributions, those are initialized randomly and their parameters are iteratively optimized to fit better to the data points.
- **Density-Based K-Means:** It is a fast version of K-Means that tries to find the clusters based on the density of data points in a region (Witten and Frank 2005).
- **Farthest First:** It is another fast version of K-Means that in each step places each cluster center at the point farthest from the existing cluster center. This process results in less readjustments and greatly speed up the clustering model. However, it mostly achieves lower quality of clustering (Witten and Frank 2005).

We employ two data mining software packages to perform these clustering tasks in our experiments: Rapidminer (Rapidminer 2016) and Weka (Weka 3: Data mining software in java 2016). By employing the found clusters from these four clustering methods in our CBMF model, we examine the effect of clustering quality on CBMF's resulting RMSE (Fig. 7). As expected, Farthest First model achieves lower accuracy as it provides a poor clustering performance. Density-Based K-Means also is a fast version of K-Means which provides a slightly faster but lower quality of clustering. Thus, Density-Based K-Means shows a lower rating prediction accuracy compared to the original K-Means. Also, in both datasets, EM and K-Means result in almost the same rating prediction accuracy. However, EM has more complexity compare to K-Means. Thus, we select K-Means as the default clustering method in our experiment.

We also observed similar results for our proposed CB-AsySVD model. In addition, CB-AsySVD model results in a RMSE of 0.90523 for a random assignment of the clusters ($m' = n' = 100$) in the MovieLens100k dataset, while it achieves 0.89668 by employing K-Means's found clusters. Therefore, by increasing the quality of clusters, our proposed CBMF and CB-AsySVD models achieve a better rating prediction accuracy.

4.4.2 Employing more clusters

As is mentioned in Section 3.3, different numbers of clusters capture different levels of similarity between users and item.

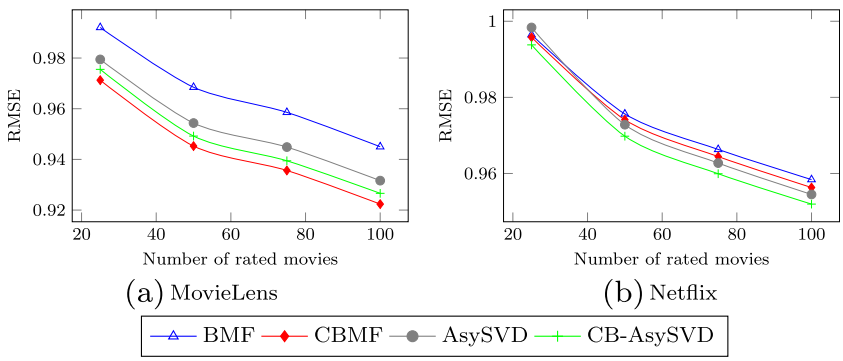

Fig. 6 A comparison over the RMSE of the extended and non-extended models for the cold-start users

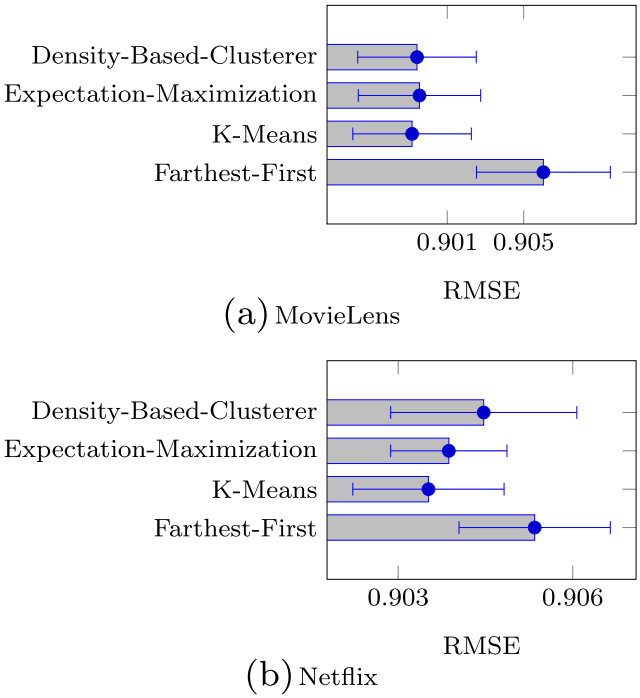

Fig. 7 Applying different clustering methods in users and items latent spaces in both datasets and its effect on the CBMF's result

For instance, by clustering movies into 10 clusters in the MovieLens dataset, we expect to find similar movies based on a general feature such as genre. However, by increasing the number of clusters to 200, we expect to distinguish between movies based on more concrete differences such as the production year. Thus, we expect to improve rating prediction accuracy even further by employing all these clusters with variety of sizes into our proposed CBMF model. For making these clusters in the MovieLens dataset, we start by applying K-Means on users and items latent spaces to find 10 clusters $n' = m' = 10$, we then gradually increase m' and n' until $n' < n/4$ and $m' < m/4$.

As Fig. 8 shows by employing more clusters in variety of sizes, rating prediction accuracy improves slightly in the MovieLens dataset. However, we do not use all these clusters in our final CBMF model because they increase the complexity of the proposed models and improve the rating prediction accuracy only slightly.

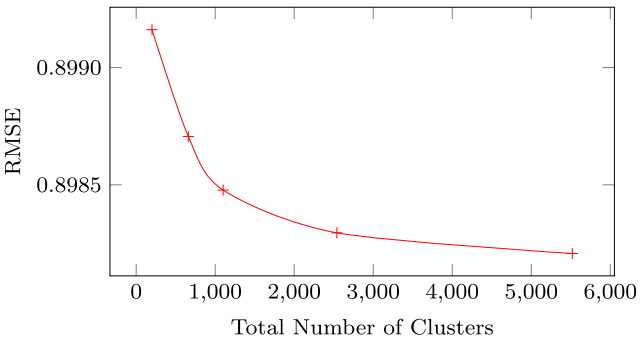

Fig. 8 Applying clustering multiple times with different number of clusters and employ those found clusters in CBMF model. By employing more clusters with variety of sizes, rating prediction accuracy improves slightly

5 Discussions and conclusions

To summarize, accuracy and scalability are two main challenges to design an effective recommendation system. Current neighbourhood-aware collaborative filtering models have to include a large neighbourhood size to achieve a desirable rating prediction accuracy. However, employing a large neighbourhood size results in a high complexity and poor comprehensibility. In this paper, we propose a novel method that applies a clustering algorithm to the latent vectors of users and items. We capture the common interests between the cluster of users and the cluster of items in a latent space. A matrix factorization technique is then applied to this cluster-level rating matrix to predict the future cluster-level interests. We then leverage these cluster-level interests to improve rating prediction accuracy of a number of well-known collaborative filtering methods. Our method is a general "wrapper" that can be applied to all other collaborative filtering methods. Additionally, interpreting the achieved clusters can add more comprehensibility to the generated recommendations. We employ RMSE evaluation metric in our experiment on two well-known datasets: the Netflix dataset, and the MovieLens100k dataset. Our experiment shows that utilizing our formed cluster-level ratings improves rating prediction accuracy. Above all, in this paper we show that better quality and more quantity of these clusters achieve a better rating prediction accuracy.

Acknowledgments This work was made possible by the facilities of the Shared Hierarchical Academic Research Computing Network (SHARCNET: www.sharcnet.ca) and Compute/Calcul Canada. The authors would like to thank the reviewers of the 2013 ACM Recommender System conference (RecSys'13) for their valuable comments.

Appendix

In this section, we list all the parameters that we employ in the learning process of our proposed CBMF model in our experiment in the datasets (Algorithm 1). Table 4 shows these selected values.

Table 4 Employed parameters in Algorithm 1 in the datasets

Parameter	Netflix	MovieLens
n'	1000	100
m'	500	100
α	0.1	0.5
β	0.7	0.8
γ_1	0.005	0.005
γ_2	0.002	0.005
γ_3	0.005	0.005
γ_4	0.0005	0.00002
λ_1	0.01	0.035
λ_2	0.1	0.065
λ_3	0.0001	0.0001
λ_4	0.055	0.0001

References

Rapidminer (2016). http://www.rapidminer.com. Accessed.

Weka 3: Data mining software in java (2016). http://www.cs.waikato.ac.nz/ml/weka/. Accessed.

Balijepally, V., Mangalaraj, G., & Iyengar, K. (2011). Are we wielding this hammer correctly? A reflective review of the application of cluster analysis in information systems research. *Journal AIS*, *12*(5). http://aisel.aisnet.org/jais/vol12/iss5/1.

Beutel, A., Murray, K., Faloutsos, C., & Smola, A.J. (2014). *Cobafi: Collaborative bayesian filtering*. NY, USA: ACM. doi:10.1145/2566486.2568040.

Bishop, C.M. (2006). *Pattern recognition and machine learning (information science and statistics)*. NJ, USA: Springer-Verlag New York, Inc.

Connor, M., & Herlocker, J. (1999). Clustering items for collaborative filtering. In *Proceedings of the ACM SIGIR Workshop on Recommender Systems*. Berkeley, CA.

Desrosiers, C., & Karypis, G. (2011). A comprehensive survey of neighborhood-based recommendation methods, In Ricci, F., Rokach, L., Shapira, B., & Kantor, P.B. (Eds.) *Recommender Systems Handbook* (pp. 107–144). US: Springer. doi:10.1007/978-0-387-85820-3_4.

Ekstrand, M.D., Riedl, J.T., & Konstan, J.A. (2011). Collaborative filtering recommender systems. *Foundations Trends Human-Computer Interaction*, *4*(2), 81–173. doi:10.1561/1100000009.

George, T., & Merugu, S. (2005). A scalable collaborative filtering framework based on co-clustering. In *Proceedings of the Fifth IEEE International Conference on Data Mining, ICDM '05* (pp. 625–628). DC, USA: IEEE Computer Society. doi:10.1109/ICDM.2005.14.

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

Herlocker, J.L., Konstan, J.A., Borchers, A., & Riedl, J. (1999). 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* (pp. 230–237). NY, USA: ACM. doi:10.1145/312624.312682.

Jamali, M., Huang, T., & Ester, M. (2011). A generalized stochastic block model for recommendation in social rating networks. In *Proceedings of the Fifth ACM Conference on Recommender Systems, RecSys '11* (pp. 53–60). NY, USA: ACM. doi:10.1145/2043932.2043946.

Konstan, J.A., & Riedl, J.T. (2012). Recommender systems: from algorithms to user experience. *User Modeling and User-Adapted Interaction*, *22*, 101–123. doi:10.1007/s11257-011-9112-x.

Koren, Y. (2008). 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* (pp. 426–434). NY, USA: ACM. doi:10.1145/1401890.1401944.

Koren, Y. (2010). Factor in the neighbors: Scalable and accurate collaborative filtering. *ACM Transactions Knowledge Discovery Data*, *4*(1), 1:1–1:24. doi:10.1145/1644873.1644874.

Koren, Y., & Bell, R. (2011). Advances in collaborative filtering, In Ricci, F., Rokach, L., Shapira, B., & Kantor, P.B. (Eds.) *Recommender Systems Handbook* (pp. 145–186). US: Springer. doi:10.1007/978-0-387-85820-3_5.

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, *42*(8), 30–37. doi:10.1109/MC.2009.263.

Mirbakhsh, N., & Ling, C.X. (2013). Clustering-based factorized collaborative filtering. In *Proceedings of the 7th ACM conference on Recommender systems, RecSys '13* (pp. 315–318). NY, USA: ACM. doi:10.1145/2507157.2507233.

Mirbakhsh, N., & Ling, C.X. (2015). Improving top-n recommendation for cold-start users via cross-domain information (accepted to publish) the Transactions on Knowledge Discovery from Data (TKDD).

Ning, X., & Karypis, G. (2012). Sparse linear methods with side information for top-n recommendations. In *Proceedings of the Sixth ACM Conference on Recommender Systems, RecSys '12* (pp. 155–162). NY, USA: ACM. doi:10.1145/2365952.2365983.

Rashid, A.M., Karypis, G., & Riedl, J. (2008). Learning preferences of new users in recommender systems: an information theoretic approach. *SIGKDD Exploration Newsletter*, *10*(2), 90–100. doi:10.1145/1540276.1540302.

Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2009). Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI '09* (pp. 452–461). Virginia, US: AUAI Press. http://dl.acm.org/citation.cfm?id=1795114.1795167.

Steck, H. (2010). Training and testing of recommender systems on data missing not at random. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '10* (pp. 713–722). NY, USA: ACM. doi:10.1145/1835804.1835895.

Steck, H. (2010). Training and testing of recommender systems on data missing not at random. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '10* (pp. 713–722). NY, USA: ACM. doi:10.1145/1835804.1835895.

Töscher, A., Jahrer, M., & Legenstein, R. (2008). Improved neighborhood-based algorithms for large-scale recommender systems. In *Proceedings of the 2nd KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition, NETFLIX '08* (pp. 4:1–4:6). NY, USA: ACM. doi:10.1145/1722149.1722153.

Witten, I.H., & Frank, E. (2005). *Data mining: Practical machine learning tools and techniques, second edition (morgan kaufmann series in data management systems)*. CA, USA: Morgan Kaufmann Publishers Inc.

Xu, B., Bu, J., Chen, C., & Cai, D. (2012). An exploration of improving collaborative recommender systems via user-item subgroups. In *Proceedings of the 21st international conference on World Wide Web, WWW '12* (pp. 21–30). NY, USA: ACM. doi:10.1145/2187836.2187840.

Nima Mirbakhsh is the lead Data Scientist at Arcane Inc. He receives his PhD in computer science at Western University in 2015. He has published several peer-reviewed research papers. In 2015, he won the TalentEdge Fellowship Program grant from Ontario Centres of Excellence (OCE) for two years.

Charles X. Ling has been a Faculty member for over 25 years. He is a Professor in Computer Science, and Distinguished Professor in Science at Western University. His research focuses on (Big) Data Analytics, machine learning, and data mining applications. He has published over 150 peer-reviewed research papers. Recently, he applies his research to mobile health, and created GlucoGuide, which won the First Prize in Diabetes Research Day at the Schulich School of Medicine & Dentistry of Western. He is the CEO and Founder of GlucoGuide Corp.