Joint User Attributes and Item Category in Factor Models for Rating Prediction

Jiang Wang[1], Yuqing Zhu[2], Deying Li[1](✉), Wenping Chen[1], and Yongcai Wang[1]

[1] School of Information, Renmin University of China, Beijing 100872, China
{jiangw,deyingli,wenpingchen,ycw}@ruc.edu.cn

[2] Department of Computer Science, California State University at Los Angeles, Los Angeles, CA 90032, USA
yzhu14@calstatela.edu

Abstract. One important problem of recommender system is rating prediction. In this paper, we use the movie rating data from MovieLens as an example to show how to use users' attributes to improve the accuracy of rating prediction. Through data analysis, we observe that users having similar attributes tend to share more similar preferences and users with a special attribute have their own preferred items. Based on the two observations, we assume that a user's rating to an item is determined by both the user intrinsic characteristics and the user common characteristics. Using the widely adopted latent factor model for rating prediction, in our proposed solution, we use two kinds of latent factors to model a user: one for the user intrinsic characteristics and the other for the user common characteristics. The latter encodes the influence of users' attributes which include user age, gender and occupation. On the other hand, we jointly use user attributes or item category information and rating data for calculating similarity of users or items. The similarity calculating results are used in our proposed latent factor model as a regularization term to regularize users or items latent factors gap. Experimental results on MovieLens show that by incorporating users' attributes influences, much lower prediction error is achieved than the state-of-the-art models. The prediction error is further reduced by incorporating influences from item category popularity and item popularity.

Keywords: Recommendation · Rating prediction · Matrix factorization

1 Introduction

With the rapid increasing of web information, recommender system is becoming more and more popular since it is good for both consumers and service or product provider. On one hand, it helps consumers to alleviate the information overload problem. On the other hand, it brings better sales to product or service provider. Diversity recommender systems have been designed to meet the needs of different platforms in recent years. Such as commodities recommendation for

S.B. Navathe et al. (Eds.): DASFAA 2016, Part I, LNCS 9642, pp. 277–296, 2016.
DOI: 10.1007/978-3-319-32025-0_18

E-commerce websites, friends recommendation for social network platform and point-of-interest recommendation for location-based social networks (LBSN).

Depending on the application, different recommendation problems have been defined and studied. The top-N item recommendation and rating prediction are two most widely studied categories of recommendation problems. On one hand, top-N item recommendation tasks aim to recommend a user a list of items that she may be interested in. For example, in movie websites, movie recommendation aims to recommend unseen movies to users. Rating prediction, on the other hand, is to predict the preference rating of a user to a product or service that she has not rated before. The products or services with high predicted ratings are recommended to users. In this study, we are interested in the movie rating prediction problem with user attribute data from MovieLens.

In recent years, social information is widely used in the recommendation system and the performance has been improved a lot. One popular method using social information in recommendation systems is SocialMF [1]. SocialMF is proposed by adding the social regularization term to the matrix factorization objective function. The additional social regularization term ensures that the distance of the latent vectors of two users or items will become closer if two users or items are more similar. Commonly, users' or items' similarity are computed based on user-item ratings, which is not applicable for new users or items. In our dataset, both user attributes and item categories information are available. For item rating prediction, an interesting question here is: Are the user attributes and item categories information helpful to similarity calculation and can they improve the accuracy of rating prediction?

To answer the question above, we conducted data analysis on MovieLens's movie rating data. We observe that users having similar attributes tend to share more similar user preferences and users with certain attribute have their own favorite items, while other users without that attribute may not be interested in those items. Based on these observations, we incorporate user attributes influences into our movie rating prediction model which is based on the widely adopted latent factor model realized by Matrix Factorization (MF). Together with influences from other factors including item category, user neighbor, item neighbor, item popularity and item category popularity, we show that the proposed model outperforms state-of-the-art baselines including Biased MF, RSVD and Social MF [1,2], measured by both Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). To the best of our knowledge, this is the first study exploiting user attributes as latent factor in item rating prediction. To summarize, the main contributions arise from this study are as follows:

- We conduct data analysis and observe that users having similar attributes tend to share more similar user preferences and users with certain attribute have their own favorite items, while other users without that attribute may not be interested in those items. This is an important observation that could be useful for not only item rating prediction, but also related studies, e.g., item recommendation and user attribute analysis from rating data.

- We directly model the influence of user attributes into movie rating prediction using matrix factorization. Specifically, for each user, we use two different latent factors to represent his or her intrinsic and common characteristics respectively. Every user's intrinsic latent factors are unique and determined by himself or herself. While common latent factors of a user are determined by his or her attributes' type and every attribute has a latent factors. On the whole, a user's latent factors are modeled as the linear combination of his or her intrinsic latent factors and all attributes' latent factors. In our recommender system, we have also considered other factors including item popularity and item category popularity. To the best of our knowledge, this is the first study that models user attribute influence as latent factors into rating prediction, while previous studies only take a attribute as a bias.
- We conducted extensive experiments to evaluate the effectiveness of incorporating influence of user attributes and other factors in item rating prediction and compared the prediction accuracy of the proposed model with an array of strong baseline models. The experiment results show that using both user or item profile information and rating sequence information for selecting appropriate neighbor can further improve the performance and different user attributes have imparity status in item rating prediction.

The rest of this paper is organized as follows. We review the related work in Sect. 2. The data analysis is reported in Sect. 3. The details of the proposed item rating prediction model is presented in Sect. 4, Experimental evaluation is showed in Sect. 5. Finally, Sect. 6 concludes this paper.

2 Related Work

The collaborative filtering (CF) technique has become more and more popular in personalized recommender task [3–6] since CF methods are domain independent and only require the past activities history of users, i.e. user-item rating matrix, to make recommendations. According to different means of utilizing the user-item rating matrix, collaborative filtering approaches are usually divided into two kinds of categories [7]: memory-based CF and model-based CF.

Memory-based CF, also called as neighbor-based methods, use the history user rating behaviors data for recommendations. The basic idea of memory-based CF is to find similar users or items for target user or item by using similarity measures. Once the neighborhoods are formed, memory-based CF usually takes a weighted sum of ratings given by their neighbors (target user' neighbors or target item' neighbors) as a prediction. user-based CF [7,8] and item-based CF [9,10] are two typical kinds of memory-based method. User-based CF predicts the ratings based on the opinions of target user's neighbors, which have similar item preferences with target user. On the other hand, item-based CF provide prediction based on the ratings of items that is similar to target item.

In contrast with memory-based CF, which utilize entire user-item matrix to provide recommendations for target users, model-based CF make use of statistical and machine learning techniques to learn a predictive model from training data. The predictive model can characterize the rating behaviors of target users. Then model-based CF use the trained model to predict the unobserved ratings, rather than directly utilize the entire user-item matrix to compute predictions. Latent factor model is one of the most successful CF models, in which users and items are jointly mapped into a shared latent space of much lower dimensionality. As the most successful realization of latent factor model, matrix factorization (MF) [1,11,12] has been successfully applied to various recommender systems including location rating prediction for Jiepang [13] and Yelp [14], event recommendation for Meetup [15] and Douban [16], and personalized tweet tag recommendation [17].

Among various MF models proposed, SVD++ [11] is probably one of the most successful models. This model integrates the implicit feedback information from a user to items (e.g., based on user' s purchase history or browsing history). More specifically, the user vector of latent factors in this model is complemented by the latent factors of the items to which the user has provided implicit feedback. Recently, Ma [1] proposed a social regularization MF method, named Social MF, to employ the similar and dissimilar relationships between users and items to improve recommendation accuracy. The similarity between items is measured based on their ratings using Pearson' s correlation or cosine similarity. It is usually difficult to reflect the real similarity of two users or items that only using the rating information to measure similarity. In many literatures, many other user-/item-specific attributes are introduced to calculating user similarity or item similarity. In contrast with rating data, user attribute information, for example, age, gender, occupation for a user, can represent the characteristics more accurately. Item has a similar situation with user. In our experiments, user attribute information and item category information are introduced to calculating similarity, and the proposed models achieve better rating prediction accuracy than Social MF [1] indicating that adding the influence from attributes neighborhood is more effective than the influence only from neighbors chosen by rating-based similarity measures.

Based on MF, influence from other aspects of users or items besides the ratings can be flexibly and easily modeled. For example, Koenigstein et al. [18] incorporated rich item bias into MF model to capture the taxonomy information of music. Each music has multiple types of information such as track, album, artist and genre. In their proposed model, MF was extended by adding shared bias parameters for items linked by common taxonomy. Moreover, some other work has also shown that popularity is helpful in improving the recommendation accuracy [19,20]. To the best of our knowledge, user or item attributes are introduced to most of literatures as biases. However, few work focus on taking user or item attributes information as latent factors to improve the quality of recommendation. In our problem setting, we consider the attributes from a user further elaborates about the rating. We model users' attributes as latent factors and incorporate the attributes into rating prediction. Next, we conduct data analysis of the MovieLens data.

3 Data Analysis

3.1 Dataset

Our study is based on the popular released MovieLens Dataset[1]. The data was collected through the MovieLens web site[2] during the seven-month period from September 19th, 1997 through April 22nd, 1998. It contains 100,000 user-item ratings (scale from 1 to 5) rated by 943 users on 1,682 items.

This data has been cleaned up - users who had less than 20 ratings or did not have complete demographic information were removed from this data set. A user contains a unique id, name, age, gender and occupation. This data have 21 occupations and a user only has one of them. An item has a unique id, name, release date, URL and its categories. This data have 19 categories and an item may have more than one category. However, we don't use released date and URL in our study. A rating tuple contains user id, item id, rating from 1 to 5 stars and date. In this study, we do not use the date feature for its less relevance to our research problem. Table 1 reports the minimum, maximum, and average number of ratings per user and per item respectively.

Table 1. Statistics of dataset movieLens

Statistics	User	Item
Min. Num. of Ratings	20	1
Max. Num. of Ratings	737	583
Avg. Num. of Ratings	106.04	59.45

3.2 Observations

To gain a better understanding of users' rating behaviors, we study the rating data introduced above. More specifically, we study the influence of user attributes for rating behavior at two levels: item-level and rating-level. We make two observations from the data analysis: (a) Users have similar attributes tend to share more similar user preferences; and (b) Users with certain attribute have their own favorite items, while other users without that attribute may not be interested in those items. These patterns are independent of any particular user. We now report the details of the data analysis.

Item Level. To better study users rating behaviors at rating level, we cluster the users into groups by user attributes. Users having same attribute value are clustered in same group. As an example, we cluster users into female group and male group by gender. In order to observe the item level rating behaviors more intuitively, we measure the Rating Count Difference(RCD) of each item i for

[1] http://www.grouplens.org/system/files/ml-100k.zip.

[2] movielens.umn.edu.

two user groups us_j and us_k, these two user groups have different attribute value a_j and a_k, respectively. Considering the user number gap of those two user groups, using Absolute Rating Count Difference(ARCD) is unfair. Hence we use Normalized Rating Count Difference(NRCD) to measure RCD, defined as follows.

$$NRCD_i(us_j, us_k) = \frac{\frac{\sum\limits_{t\in I} |us_{kt}|}{\sum\limits_{t\in I} |us_{jt}|} \cdot \mid us_{ji} \mid - \mid us_{ki} \mid}{\max\limits_{t\in I}\left(\frac{\sum\limits_{t\in I} |us_{kt}|}{\sum\limits_{t\in I} |us_{jt}|} \cdot \mid us_{ji} \mid - \mid us_{ki} \mid\right)} \tag{1}$$

Where $NRCD_i(us_j, us_k)$ is normalized rating count difference of user group us_j and user group us_k for item i, I is the set of all items, us_{ji} and us_{ki} denote the two sets of users who have rated item i, and $\mid \bullet \mid$ denotes the cardinality of a set.

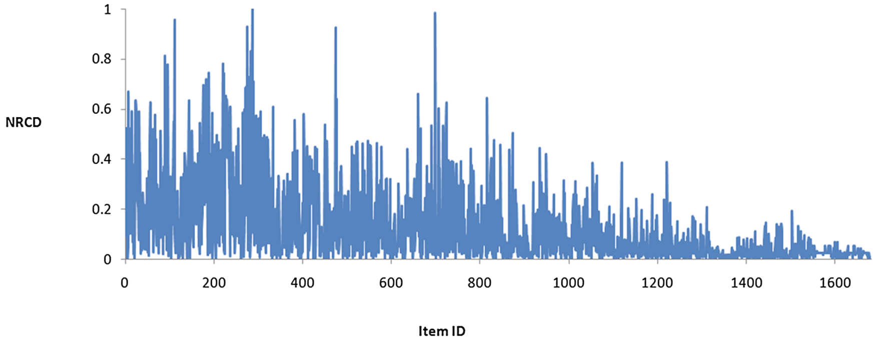


Fig. 1. Normalized Rating Count Difference(NRCD) of female and male

Figure 1 plots the rating count difference(NRCD) of female and male. As shown in Fig. 1, female's rating count for each item shows great difference to male's. Specifically, about 13 % items' NRCD is over 0.3 and average of NRCD is about 0.13. Another interesting phenomenon is about 45 % items' NRCD less than 0.05. These two phenomena indicate that female and male share similar preferences for most of items, while for some particular items, their preferences show huge gap. In other words, users' gender has a certain degree of influence for users' rating behaviors. For age (3 age groups) and occupation (21 occupations), Rating count difference have similar circumstance with gender. As a result of the limitation of space, we omit the NRCD figures of age groups and occupations. Table 2 lists top-5 ARCD items of young adults and mid adults. Item 286 has the max ARCD of 213, while the max NRCD of items is 515. This is because NRCD and ARCD are two measure methods which have a certain relation, but inequitable. Any way, these two measure methods both indicate user attributes have influence of user's rating behaviors in another way.

Table 2. Top-5 ARCD items of Young Adults(YA) and Mid Adults(MA)

Item ID	RC YA	RC MA	NRCD	ARCD
286	87	300	0.67	**213**
269	54	206	0.55	152
515	13	151	**0.83**	138
211	19	152	0.74	133
483	37	169	0.57	132

Table 3. Statistics of Student(S) and Educator(E)

Statistics	Item id	Group	Diff
Max. of Avg Ratings Difference	919	-	1.46
Min. of Avg Ratings Difference	257,etc.	-	0
Avg. of Avg Ratings Difference	-	-	0.32
Max. of Avg Ratings	170	S	4.93
Max. of Avg Ratings	48	E	4.69

Rating Level. At the rating level, we analyze the average of each item's rating for diverse user groups. Without loss of generality, we use student group and educator group as an example. These two user groups clustered according occupation. To avoid the influence of sparse data, we don't consider items which are less than 5 user rated in either student or educator. After the preprocessing, only 500 items remind. But we surprising discover that about 30 % items' average difference are larger than 0.5. Especially, the max gap reaches 1.46 for item 919. More detailed statistics of these two groups are summarized in Table 3.

We also compare others user group pairs, they have similar circumstance with student and educator group. From these analysis in rating level, we can get a similar conclusion with item level — user attributes have certain influence for user rating behaviors.

Motivated by the two observations, we propose incorporating the above attribute characteristics for rating prediction, so as to improve item recommendation accuracy.

4 Rating Prediction

Rating prediction is a basic problem in recommendation system and has been widely studied in literature. In this paper, we use r_{ui} to denote the rating that user u gives to item i (i.e., a movie), and r_{ui} is in the range of 1 to 5 stars with more stars indicating higher preference. Given the existing ratings made by M users to N items, the task is to predict the unknown rating $\hat{r}_{ui}$, if user u has not

Table 4. Notations and semantics

N_i	Set of neighbors of item i
K	Set of (u,i) pairs with known r_{ui} ratings
T	Set of (u,i) pairs using test
s_{uf}	Similarity of user(item) u and user(item)f
s^a_{uf}	Attribute similarity of user(item) u and user(item)f
s^r_{uf}	Rating similarity of user(item) u and user(item)f
r_{ui}	Observed ratings of user u to item i
$\hat{r}_{ui}$	Predicted ratings of user u to item i
μ	Mean of all known r_{ui} ratings
b_u	Bias parameters for user u
b_i	Bias parameters for item i
$\mathbf{p}_u$	Latent factors of user u
$\mathbf{q}_i$	Latent factors of item i
$\mathbf{a}_{ua}$	Latent factors of user u's age group
$\mathbf{g}_{ug}$	Latent factors of user u's gender
$\mathbf{o}_{uo}$	Latent factors of user u's occupation
γ	Similarity weighting parameter
ρ_i	Normalized popularity of item i
γ_i	Popularity weighting parameter for item i
τ_c	Normalized popularity of item category c
γ_c	Popularity weighting parameter for item category c

rated item i before. In the following, we first briefly introduce matrix factorization and then present our proposed model by incorporating various influences into the prediction. Table 4 lists the notations used in this paper.

4.1 Matrix Factorization

Our proposed method is based on the latent factor model realized by matrix factorization. Matrix factorization map users and items into a joint latent space with dimension $f << min(M; N)$. The inner product of a user vector $\mathbf{p}_u \in R^{f \times 1}$ and an item vector $\mathbf{q}_i \in R^{f \times 1}$ is used to approximate the user's preference to the item (see [21] for a detailed introduction of matrix factorization). Accordingly, the predicted rating of user u to item i is computed using

$$\hat{r}_{ui} = \mathbf{p}_u^T \mathbf{q}_i, \tag{2}$$

where $\mathbf{p}_u$ and $\mathbf{q}_i$ can be learned from the user-item rating matrix with known ratings. However, users may have certain degree of biases: some users are more lenient and some are very strict about ratings. Similarly, items may also have some degree of biases because of location or branding for example. To achieve more accurate rating prediction, Biased MF extends the basic matrix factorization by considering the biases,

$$\hat{r}_{ui} = \mu + b_u + b_i + \mathbf{p}_u^T \mathbf{q}_i, \tag{3}$$

where μ is the average rating of all known ratings, b_u and b_i are the user bias and item bias, respectively. Learning the unknown parameters $\mathbf{p}_u$, $\mathbf{q}_i$, b_u and b_i is an optimization problem to minimize the regularized squared error on the set of known ratings K.

$$\min_{p^\star, q^\star, b^\star} \sum\nolimits_{(u,i)\in K} (r_{ui} - \hat{r}_{ui})^2 + \lambda_1(\parallel \mathbf{p}_u \parallel^2 + \parallel \mathbf{q}_i \parallel^2) + \lambda_2(b_u^2 + b_i^2)$$

In this equation, λ_1 and λ_2 are regularization parameters used to avoid overfitting. Both stochastic gradient descent (SGD) and alternating least squares (ALS) algorithms can be used to solve the optimization function and learn the parameters [2,21]. In this paper, we adopt SGD to learn the parameters following the algorithm presented in [2].

4.2 Incorporating Neighborhood Influence

Inspired by [1,22], user or item neighbors information can help rating prediction. In the case of missing explicit neighbor information in MovieLens, we use top-N similar users of target user instead of his or her neighbors information and then plug in those similar users to the aforementioned matrix factorization framework. There are several methods we can borrow in the literature to compare the similarity between two users. In this paper, we jointly adopt rating similarity and attribute similarity, which is defined as:

$$\begin{aligned}
s_{uf} &= \gamma \cdot s_{uf}^a + (1-\gamma) \cdot s_{uf}^r \\
s_{uf}^r &= \frac{\sum_{k\in I(u)\cap I(f)} (r_{uk} - \bar{r}_u) \cdot (r_{fk} - \bar{r}_f)}{\sqrt{\sum_{k\in I(u)\cap I(f)} (r_{uk} - \bar{r}_u)^2} \cdot \sqrt{\sum_{k\in I(u)\cap I(f)} (r_{fk} - \bar{r}_f)^2}} \\
s_{uf}^a &= \frac{1}{3}(s_{uf}^{age} + s_{uf}^{gender} + s_{uf}^{occupation})
\end{aligned}$$

where s_{uf} is the similarity between user u and f, s_{uf}^r indicates the rating similarity and we adopt Pearson Correlation Coefficient(PCC) to calculate it. $I(u)$ is a set of items that rated by user u, and $\bar{r}_u$ represents the average rate of user u. s_{uf}^a represents the attribute similarity and it's a linear combination of age, gender and occupation similarity. Age similarity s_{uf}^{age} is a normalized value of the age difference of user u and f. Gender similarity s_{uf}^{gender} is equal to 1 if the gender of user u and f are the same, otherwise it is a small smooth value. The calculation method of occupation similarity $s_{uf}^{occupation}$ is similar to s_{uf}^{age}. And parameter γ

is introduced to balance the importance between rating and attribute similarity. Incorporating neighborhood influence, we proposed the matrix factorization objective function as follow:

$$\min_{p^\star,q^\star,b^\star} \sum_{(u,i)\in K} (r_{ui} - \hat{r}_{ui})^2 + \beta_1 \sum_{f\in N(u)} s_{uf} \parallel \mathbf{p}_u - \mathbf{p}_f \parallel^2 + \lambda_1(\parallel \mathbf{p}_u \parallel^2 + \parallel \mathbf{q}_i \parallel^2) + \lambda_2(b_u^2 + b_i^2)$$

where β_1 is the regularization parameter, and $N(u)$ represents user i's neighbors. In this method, the neighbors information is employed in designing the neighbor regularization term to constrain the matrix factorization objective function. The neighbor regularization term also indirectly models the difference of users' tastes. More specifically, if user u has a high similarity with user f, this regularization term actually indirectly minimizes the distance between latent vectors $\mathbf{p}_u$ and $\mathbf{p}_f$.

From the above, since we define the implicit user social information as the similar users, we can naturally extend this idea to take advantages of the implicit item social information, which can be found through the similar items. The similarity calculation method of items are similar to users, and also includes rating similarity and attribute similarity. Attribute similarity is calculated by cosine similarity of two items' category vector. The Social Regularization method described above is a very general approach, and it can be easily extended to incorporate the item social information. The objective function can be formulated as:

$$\begin{aligned}\min_{p^\star,q^\star,b^\star} & \sum_{(u,i)\in K} (r_{ui} - \hat{r}_{ui})^2 + \beta_1 \sum_{f\in N(u)} s_{uf} \parallel \mathbf{p}_u - \mathbf{p}_f \parallel^2 + \beta_2 \sum_{f'\in N(i)} s_{if'} \parallel \mathbf{q}_i - \mathbf{q}_{f'} \parallel^2 \\ & + \lambda_1(\parallel \mathbf{p}_u \parallel^2 + \parallel \mathbf{q}_i \parallel^2) + \lambda_2(b_u^2 + b_i^2)\end{aligned}$$

4.3 Incorporating User Profile Influence

Based on our observations in Sect. 3.2, user's attributes have influence of item rating. These observations suggest that considering user attributes influence may improve the accuracy of item rating prediction.

In this paper, to model users' rating behavior, we first assume that a user's rating preference is determined by the user intrinsic characteristics and the common characteristics. Each user' intrinsic characteristics are unique, while common characteristics are shared by all users. Limited by the data set, common characteristics using in this paper contain age, gender and occupation characteristics. For a user u, we use $\mathbf{p}_u$, $\mathbf{a}_{ua}$, $\mathbf{g}_{ug}$ and $\mathbf{o}_{uo}$ to model its intrinsic, age, gender and occupation characteristics, respectively. Next, we use user profile for rating prediction in our proposed methods.

Age Influence. Analyzed in Sect. 3.2, users' age span in MovieLens is very big, the youngest is only 7 years old, while the oldest is 73. But users with similar ages show close preferences. In our model, we introduce age latent factors to exploit age groups for more accurate item rating prediction. For an age group

Table 5. Objective functions for incorporating neighborhood influence, age influence, gender and other factors

$$\min_{p\star,q\star,a\star,b\star} \sum_{(u,i)\in K} (r_{ui}-\hat{r}_{ui})^2 + \beta_1 \cdot \sum_{f\in N(u)} s_{uf} \parallel \mathbf{p}_u - \mathbf{p}_f \parallel^2 + \beta_2 \cdot \sum_{f'\in N(i)} s_{if'} \parallel \mathbf{q}_i - \mathbf{q}_{f'} \parallel^2 + \lambda_1(\parallel \mathbf{p}_u \parallel^2 + \parallel \mathbf{q}_i \parallel^2) + \lambda_2(b_u^2 + b_i^2) + \lambda_3 \parallel \mathbf{a}_a \parallel^2 \tag{9}$$

$$\min_{p\star,q\star,a\star,g\star,b\star} \sum_{(u,i)\in K} (r_{ui}-\hat{r}_{ui})^2 + \beta_1 \cdot \sum_{f\in N(u)} s_{uf} \parallel \mathbf{p}_u - \mathbf{p}_f \parallel^2 + \beta_2 \cdot \sum_{f'\in N(i)} s_{if'} \parallel \mathbf{q}_i - \mathbf{q}_{f'} \parallel^2 + \lambda_1(\parallel \mathbf{p}_u \parallel^2 + \parallel \mathbf{q}_i \parallel^2) + \lambda_2(b_u^2 + b_i^2) + \lambda_3(\parallel \mathbf{a}_a \parallel^2 + \parallel \mathbf{g}_g \parallel^2) \tag{10}$$

$$\begin{aligned} \min_{p\star,q\star,a\star,g\star,o\star,b\star} & \sum_{(u,i)\in K} (r_{ui}-\hat{r}_{ui})^2 + \beta_1 \cdot \sum_{f\in N(u)} s_{uf} \parallel \mathbf{p}_u - \mathbf{p}_f \parallel^2 \\ & + \beta_2 \cdot \sum_{f'\in N(i)} s_{if'} \parallel \mathbf{q}_i - \mathbf{q}_{f'} \parallel^2 + \lambda_1(\parallel \mathbf{p}_u \parallel^2 + \parallel \mathbf{q}_i \parallel^2) + \lambda_2(b_u^2 + b_i^2) \\ & + \lambda_3(\parallel \mathbf{a}_a \parallel^2 + \parallel \mathbf{g}_g \parallel^2 + \parallel \mathbf{o}_o \parallel^2) \end{aligned} \tag{11}$$

$$\begin{aligned} \min_{p\star,q\star,a\star,g\star,o\star,\gamma\star,b\star} & \sum_{(u,i)\in K} (r_{ui}-\hat{r}_{ui})^2 + \beta_1 \cdot \sum_{f\in N(u)} s_{uf} \parallel \mathbf{p}_u - \mathbf{p}_f \parallel^2 \\ & + \beta_2 \cdot \sum_{f'\in N(i)} s_{if'} \parallel \mathbf{q}_i - \mathbf{q}_{f'} \parallel^2 + \lambda_1(\parallel \mathbf{p}_u \parallel^2 + \parallel \mathbf{q}_i \parallel^2) \\ & + \lambda_2(b_u^2 + b_i^2 + \gamma_i^2 + \sum_{c\in C_i} \gamma_c^2) + \lambda_3(\parallel \mathbf{a}_a \parallel^2 + \parallel \mathbf{g}_g \parallel^2 + \parallel \mathbf{o}_o \parallel^2) \end{aligned} \tag{12}$$

a, it is associated with a latent vector $\mathbf{a}_a \in R^{f\times 1}$. Let u_a be the age group of user u and $\mathbf{a}_{ua}$ be the age factor of user u. By incorporating the age influence, the predicted rating $\hat{r}_{ui}$ is now defined in Eq. 4, where $\alpha_1 \in [0,1]$ is a parameter that controls the importance of age influence in rating prediction. The objective function is updated accordingly, see Eq. 9 in Table 5.

$$\hat{r}_{ui} = \mu + b_u + b_i + (\mathbf{p}_u + \alpha_1 \cdot \mathbf{a}_{ua})^T \mathbf{q}_i, \tag{4}$$

Gender Influence. Similar with age, we study the relationship between user gender and their rating behavior in Sect. 3.2. Users in same gender have smaller preference difference than those in opposite genders. To better use gender information in our model, we introduce gender latent factors to explore the influence of gender in item recommendation. Female or male is associated with a latent vector $\mathbf{g}_g \in R^{f\times 1}$. Let u_g be the gender of user u and $\mathbf{g}_{ug}$ be the gender factor of user u. By incorporating the gender influence, the predicted rating $\hat{r}_{ui}$ is now defined in Eq. 5, where $\alpha_2 \in [0,1]$ is a parameter that controls the importance of gender influence in rating prediction. The objective function is updated accordingly, see Eq. 10 in Table 5.

$$\hat{r}_{ui} = \mu + b_u + b_i + (\mathbf{p}_u + \alpha_1 \cdot \mathbf{a}_{ua} + \alpha_2 \cdot \mathbf{g}_{ug})^T \mathbf{q}_i, \tag{5}$$

Occupation Influence. In Movielens, a user belongs to one of 21 occupations. Based on our observations in Sect. 3.2, the occupation of a user may reflects the characteristics of a user, e.g., an artist may care about items' artistic quality more, while a writer may attach more importance to items' literariness. In other words, users often consider items in their own professional perspective. Intuitively, users with same occupation show similar item preference. Inspired by the above observations, we also introduce occupation latent factors in our model for getting better rating prediction. For each occupation o, it maps to a latent vector $\mathbf{o}_o \in R^{f \times 1}$. Let u_t be occupation of user u and o_{ut} be the occupation factor of user u. By incorporating the occupation influence, the predicted rating $\hat{r}_{ui}$ is now defined in Eq. 6, where $\alpha_3 \in [0, 1]$ is a parameter that controls the importance of occupation influence in rating prediction. The objective function is shown in Eq. 11 in Table 5.

$$\hat{r}_{ui} = \mu + b_u + b_i + (\mathbf{p}_u + \alpha_1 \cdot \mathbf{a}_{ua} + \alpha_2 \cdot \mathbf{g}_{ug} + \alpha_3 \cdot \mathbf{o}_{uo})^T \mathbf{q}_i, \tag{6}$$

4.4 Item Popularity and Category Influences

The aforementioned methods are standing in the user's perspective. Next we convert perspective to item, discuss two features that have been widely used in traditional collaborative filtering method, namely *popularity* and item *category popularity*. For simplicity, we model both item popularity and category popularity as a rating bias z.

$$z = \gamma_i \cdot \rho_i + \gamma_c \cdot \tau_c \tag{7}$$

In above equation, $\rho_i \in [0, 1]$ is the normalized popularity of item i, $\tau_c \in [0, 1]$ is the normalized popularity of category c. The two parameters γ_i and γ_c are the popularity weighting parameters for item i and category c respectively, both are learned from the training data. With rating bias z, the predicted rating is shown in Eq. 8. The objective function considering both item popularity and category popularity is shown in Eq. 12 in Table 5.

$$\hat{r}_{ui} = \mu + b_u + b_i + z + (\mathbf{p}_u + \alpha_1 \cdot \mathbf{a}_{ua} + \alpha_2 \cdot \mathbf{g}_{ug} + \alpha_3 \cdot \mathbf{o}_{uo})^T \mathbf{q}_i \tag{8}$$

4.5 Parameter Estimation

All the objective functions (e.g., Eqs. 9, 10, 11 and 12) in the proposed models share the same form. Next, we detail the parameter estimation for Eq. 12 (where $z = \gamma_i \cdot \rho_i + \gamma_c \cdot \tau_c$) as an example using Stochastic Gradient Descent (SGD) algorithm [2]. Let e_{ui} be the error associated with the prediction $e_{ui} = r_{ui} - \hat{r}_{ui}$. The parameters are learned by moving in the opposite direction of the gradient with a learning rate η in an iterative manner. The details of iterative formula is as follows:

$$
\begin{aligned}
b_u &\leftarrow b_u + \eta \cdot (e_{ui} - \lambda_2 \cdot b_u)\\
b_i &\leftarrow b_i + \eta \cdot (e_{ui} - \lambda_2 \cdot b_i)\\
\gamma_i &\leftarrow \gamma_i + \eta \cdot (e_{ui} \cdot \rho_i - \lambda_2 \cdot \gamma_i)\\
\forall c \in C_i \quad &: \quad \gamma_c \leftarrow \gamma_c + \eta \cdot (e_{ui} \cdot \tau_c - \lambda_2 \cdot \gamma_c)\\
\mathbf{p}_u &\leftarrow \mathbf{p}_u + \eta \cdot (e_{ui} \cdot \mathbf{q}_i - \beta_1 \cdot \sum_{f \in N(u)} s_{uf} \cdot (\mathbf{p}_u - \mathbf{p}_f) - \lambda_1 \cdot \mathbf{p}_u)\\
\mathbf{q}_i &\leftarrow \mathbf{q}_i + \eta \cdot (e_{ui} \cdot (\mathbf{p}_u + \alpha_1 \cdot \mathbf{a}_{ua} + \alpha_2 \cdot \mathbf{g}_{ug} + \alpha_3 \cdot \mathbf{o}_{uo})\\
&\qquad - \beta_2 \cdot \sum_{f' \in N(i)} s_{uf'} \cdot (\mathbf{q}_i - \mathbf{q}_{f'}) - \lambda_1 \cdot \mathbf{q}_i)\\
\mathbf{a}_{ua} &\leftarrow \mathbf{a}_{ua} + \eta \cdot (e_{ui} \cdot \alpha_1 \cdot \mathbf{q}_i - \lambda_3 \cdot \mathbf{a}_{ua})\\
\mathbf{g}_{ug} &\leftarrow \mathbf{g}_{ug} + \eta \cdot (e_{ui} \cdot \alpha_2 \cdot \mathbf{q}_i - \lambda_3 \cdot \mathbf{g}_{ug})\\
\mathbf{o}_{uo} &\leftarrow \mathbf{o}_{uo} + \eta \cdot (e_{ui} \cdot \alpha_3 \cdot \mathbf{q}_i - \lambda_3 \cdot \mathbf{o}_{uo})
\end{aligned}
$$

5 Experiments

We now conduct experiments on the MovieLens dataset to evaluate the proposed models and compare the proposed models with state-of-the-art baselines.

5.1 Experimental Setting

Dataset. We use the MovieLens dataset that has been studied in Sect. 3.1 in our experiments. For each user, we randomly select 80 % of ratings for training, and the remaining 20 % for testing. As the result, we have 80, 000 ratings to build the matrix factorization model for the prediction of the remaining 20,000 ratings. The data sparsity is 93.7 %.

Evaluation Metric. We adopt two popular evaluation metrics, namely, Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). The smaller MAE or RMSE value means better rating prediction accuracy. In the following equations, T is the set of user-item rating pairs (u, i) used in testing.

$$MAE \quad = \quad \frac{1}{\mid T \mid} \sum_{(u,i) \in T} \mid r_{ui} - \hat{r}_{ui} \mid$$

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

Baseline Methods. We compare the proposed models with the following 8 baseline methods. All the experiments were conducted on a server with Intel Xeon E5310 1.60 GHz CPU (8 cores) and 20 G memory. We implemented the algorithms in C++ with the support of Eigen libary[3] for fast vector/matrix manipulations.

1. *Global Mean:* this method predicts an unknown rating to be the average of all known ratings, *i.e.*, $\hat{r}_{ui} = u$.
2. *User Mean:* this method utilizes the mean rating of each user to predict the missing values for the corresponding user.
3. *Item Mean:* this method uses the mean rating of each item to predict the missing values for the corresponding item.
4. *RSVD:* this is the Regularized SVD method. It is equivalent with the method proposed by Salakhutdinov and Minh in [23]. The underlining distribution is assumed as Gaussian distribution. The details of this method are also introduced in Sect. 4.1.
5. *Social MF:* this model considers implicit social information between items and/or users. The implicit social information can be derived from most similar and dissimilar users/items using Pearson's correlations or cosine similarity of ratings. As our model considers neighbors influences from not only rating information but also attribute information, for a fair comparison, this method only includes rating information from most similar items in Social MF. More detailed discussion about neighbors selection is presented in Sect. 5.2.
6. *Biased MF:* this is the MF model with user and item biases briefly described in Sect. 4.1. Biased MF is widely used as a baseline in recommender systems.
   Proposed Methods. We extended Biased MF to incorporate influences from multiple factors: user age (A), user gender (G), user occupation (O), item popularity (P), and item category popularity (C). The proposed methods are denoted using the letters in parentheses to indicate the influences considered in each method.
7. *NA-MF:* this method incorporates both implicit neighborhood and user age influence (Sect. 4.3, Eq. 5).
8. *NAG-MF:* this method incorporates both implicit neighborhood, user and gender influence (Sect. 4.3, Eq. 6).
9. *NAGO-MF:* this method incorporates implicit neighborhood, user age, user gender and user occupation influence (Sect. 4.3, Eq. 7).
10. *NAGOP-MF:* this method incorporates implicit neighborhood, user age, user gender, user occupation and item popularity influence, by setting $z = \gamma_i \cdot \rho_i$ in Eq. 8 (Sect. 4.4).
11. *NAGOPC-MF:* this model incorporates all factors: implicit neighborhood, user age, user gender, user occupation, item popularity and item category popularity influence, by setting $z = \gamma_i \cdot \rho_i + \gamma_c \cdot \tau_c$ (Sect. 4.4).

[3] http://eigen.tuxfamily.org.

We also evaluate another two methods: *AGOP-MF* and *AGOPC-MF*. These two methods do not incorporate implicit neighborhood influence but incorporate influences from other factors (i.e., A, G, O, P and C) indicated by the method names.

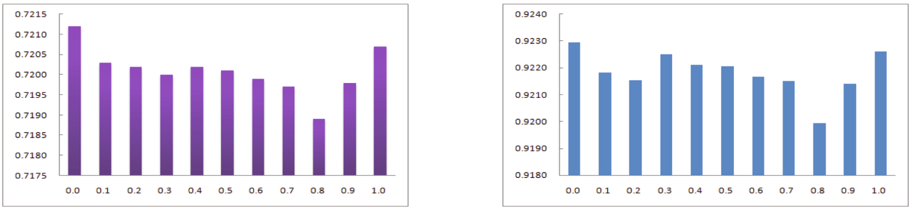
Parameter Setting. We performed 5-fold cross-validation on the training set to empirically set the hyperparameters. The number of latent factors f = 20, the weight coefficient of similarity $\gamma = 0.8$, the relative importance of age, sex and occupation influences are set to $\alpha_1 = 1$, $\alpha_2 = 1$, $\alpha_3 = 1$. The neighborhood regularization parameters are set to $\beta_1 = 0.002$, $\beta_2 = 0.002$. The regularization parameters: $\lambda_1 = 0.1$, $\lambda_2 = 0.1$, $\lambda_3 = 0.1$. The latent factors are learnt by SGD with initial learning rate $\eta = 0.01$, which decreases by a factor of 0.95 after each 10 iterations. The same parameters are used in all methods for fair comparison for all our proposed methods and the baseline methods whenever applicable. For example, the number of latent factors is also set to 20 in baseline methods *Biased MF*, *RSVD* and *Social MF*. For neighborhood influence, by default, the proposed methods use the 10 nearest neighbors for each user or item. For all the methods based on matrix factorization, the reported results are averaged over 5 runs to avoid the impact of initialization in parameter learning.

5.2 Experimental Results

We first compare the proposed methods with baseline methods and then search the best weight coefficient of similarity. Lastly, we evaluate the importance of user attributes in our proposed methods.

Method Comparison. The prediction errors measured by MAE and RMSE of all methods are reported in Table 6 with best results highlighted in boldface. We make four observations from the results.

First, incorporating attributes influence into item rating prediction greatly reduces prediction errors measured by both MAE and RMSE. All the proposed methods with attributes influence (i.e., methods 7–13) outperform all baseline



(a)MAE (b)RMSE

Fig. 2. Comparisons of different similarity weight of *NAGOP-MF* on MAE and RSME metrics.

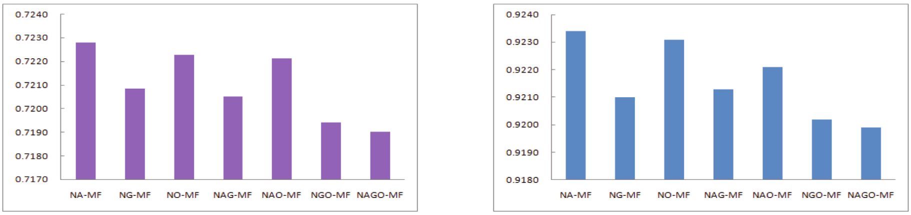
Table 6. MAE and RMSE of all methods, the lower the better.

ID	Method	MAE	RMSE
1	Global Mean	0.9435	1.1256
2	User Mean	0.8362	1.0437
3	Item Mean	0.8163	1.0229
4	RSVD	0.7492	0.9496
5	Social MF	0.7372	0.9304
6	Biased MF	0.7418	0.9468
7	NA-MF	0.7228	0.9234
8	NAG-MF	0.7205	0.9213
9	NAGO-MF	0.7193	0.9204
10	NAGOP-MF	**0.7189**	**0.9199**
11	NAGOPC-MF	0.7190	0.9201
12	AGOP-MF	0.7275	0.9321
13	AGOPC-MF	0.7275	0.9322

methods (methods 1–6). The best prediction accuracy is achieved by *NAGOP-MF* which considers neighborhood(N), user age(A), user gender(G), user occupation(O) and item popularity (P). With attribute age influence alone, *NA-MF* outperforms all baselines including state-of-the-art methods *Bias MF* and *Social MF*. This result suggests that using user attributes is of great help to item rating prediction. Further considering factors like user gender(G) and user occupation(O) leads to relatively small additional reduction in prediction errors.

Second, without incorporating neighborhood influence, *AGOP-MF* performs poorer than most methods with neighborhood influence including *NA-MF*, *NAG-MF*, *NAGO-MF*, *NAGOP-MF* and *NAGOPC-MF*. The poorer performance of *AGOP-MF* against *NA-MF* suggests that the neighborhood influence is more effective than the combination of the three factors (G, O, and P) in item rating prediction. On the other hand, the effectiveness of neighborhood influence is also reflected from the better performance of *AGOPC-MF* compared with *NAGOPC-MF*.

Third, incorporating popularity influence, *NAGOP-MF* and *NAGOPC-MF* performs better than most methods without popularity influence including *NA-MF*, *NAG-MF*, *NAGO-MF*. Such result supports our earlier discussion that considering popularity can improve rating prediction accuracy. Unnatural, method *NAGOPC-MF* incorporating item popularity(P) and item category popularity(C) influences performs poorer than *NAGOP-MF* only including item popularity influence. This may be caused by the noise resulting from using item popularity(P) and item category popularity(C) in one prediction model. Because item category popularity have certain direct relationship with item popularity.



(a)MAE (b)RMSE

Fig. 3. Comparisons of different types of methods on MAE and RSME metrics.

Last, among the 6 baseline methods, *Social MF*, the state-of-the-art methods, perform the best evaluated by both MAE and RMSE. While *Global Mean* perform the worst.

Impact of Different Similarity Weight. As described in Sect. 4.2, γ is a weight coefficient to balance the importance between rating sequence similarity and attribute similarity. Bigger γ represents attribute similarity is more important and vice versa. To explore the impact of γ on the rating prediction accuracy, we select 11 different γ from 0 to 1 to use in the *NAGOP-MF* method. The *NAGOP-MF* method is selected as the method for evaluation because it has the best performance.

Figure 2(a) and (b) respectively plot MAE and RMSE of the *NAGOP-MF* method by using different γ ranging from 0 to 1. From the figure, $\gamma = 0.8$ gives the best prediction accuracy by considering both MAE and RMSE, and only using rating similarity ($\gamma = 0$) or attribute similarity ($\gamma = 1.0$) gets worse results. These show that both rating similarity and attribute similarity are help for rating prediction, and attribute similarity are more important than rating similarity. This may be caused by that users with similar attributes can better reflect the users' close preference than those users with similar rating sequence.

Analysis of Each Attribute Factor. Age(A), gender(G), and occupation(O) factor are three main factors for rating prediction in this task. Although the results of *NA-MF*, *NG-MF*, and *NO-MF* shown in Fig. 3 can indicate their effectiveness for the task alone, the contribution of each factor to *NAGOP-MF* should also be explored. This is because combining multiple latent factors to form a unified model does not mean the results of the new model is the performance summarization of each factor.

To better understand the importance of each attribute factor, we also adopt the strategy of combining two factors from user attributes, i.e., *NAG-MF*, *NAO-MF* and *NGO-MF*. we keep using neighborhood(N) in this section because of user attribute also using in neighborhood selection, while item popular(P) is moved for the reasons that it has no relationship with user attribute. Specifically, we test the strategy of all three user attributes factors using *NAGO-MF*.

The results of them are displayed in Fig. 3. We find *NG-MF* performs clearly better than *NA-MF* and *NO-MF* both in MAE and RSME, which indicates the importance of user gender information to the task. The reason may be attributed to the dramatic preference difference between female and male as we discussed in Sect. 3.2. *NO-MF* achieves better results than *NA-MF*, which reveals age information makes a smaller contribution to *NAGOP-MF* than occupation information. This is because the number of user's occupation group is much more than age group, which leads to the occupation factors more personalized.

On the other hand, *NGO-MF* performs clearly better than the other methods with two attribute factors, which again indicates the top importance of gender, occupation take the second and age is the last. Lastly, when all the three attributes are combined together (*NAGO-MF*), the performance is further improved, which indicates each attribute is good for rating prediction.

6 Conclusion

In this work we proposed a new, simple, and efficient way to incorporate user attributes and item category on ratings prediction in several methods commonly used for matrix factorization. We firstly analyze the MovieLens dataset and find that a user's rating behavior is certainly correlated with his or her attribute type. Based on this observation, we model a user with two vectors of latent factors one for its intrinsic characteristics and the other for its common characteristics determined by his or her attribute type. The experimental results on real dataset have shown that our model is effective and outperforms several alternative methods. Other factors like user neighbor, item neighbor, item popularity and item category popularity can further improve the rating prediction accuracy. Nevertheless, using both item popularity and item category popularity in one method may bring noise and lead to bad results.

Acknowledgments. This research was supported in part by the National Natural Science Foundation of China under Grant 91124001 and the Fundamental Research Funds for the Central University, and the Research Funds of Renmin University of China under grant 2015030273.

References

1. Ma, H.: An experimental study on implicit social recommendation. In: Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 73–82. ACM (2013)
2. Koren, Y.: Factorization meets the neighborhood: A multifaceted collaborative filtering model. In: KDD, pp. 426–434. ACM (2008)
3. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. TKDE **17**(6), 734–749 (2005)

4. Koenigstein, N., Dror, G., Koren, Y.: Yahoo! music recommendations: Modeling music ratings with temporal dynamics and item taxonomy. In: RecSys, pp. 165–172. ACM (2011)
5. Verstrepen, K., Goethals, B.: Unifying nearest neighbors collaborative filtering. In: Proceedings of the 8th ACM Conference on Recommender Systems, RecSys 2014, pp. 177–184. ACM (2014)
6. Natarajan, N., Shin, D., Dhillon, I.S.: Which app will you use next?: Collaborative filtering with interactional context. In: RecSys, pp. 201–208. ACM (2013)
7. Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, pp. 43–52. Morgan Kaufmann Publishers Inc., (1998)
8. Jin, R., Chai, J.Y., Si, L.: An automatic weighting scheme for collaborative filtering. In: SIGIR, pp. 337–344 (2004)
9. Deshpande, M., Karypis, G.: Item-based top-n recommendation algorithms. ACM Trans. Inf. Syst. **22**(1), 143–177 (2004)
10. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: WWW, pp. 285–295. ACM (2001)
11. Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 426–434. ACM Press (2008)
12. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009)
13. Lian, D., Zhao, C., Xie, X., Sun, G., Chen, E., Rui, Y.: GeoMF: joint geographical modeling and matrix factorization for point-of-interest recommendation. In: Proceedings of KDD 2014, pp. 831–840. ACM (2014)
14. Hu, L., Sun, A., Liu, Y.: Your Neighbors affect your ratings: on geographical neighborhood influence to rating prediction. In: Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval, pp. 345–354. ACM (2014)
15. Pham, T.-AN., Li, X., Cong, G., Zhang, Z.: A general graph-based model for recommendation in event-based social networks. In: ICDE, Seoul, Korea, 13–17 April 2015
16. Zhang, W., Wang, J.Y.: A collective bayesian poisson factorization model for cold-start local event recommendation. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1455–1464. ACM Press (2015)
17. Feng, W., Wang, J.: We can learn your #hashtags: Connecting tweets to explicit topics. In: ICDE, Chicago, USA, 31 March-4 April 2014
18. Koenigstein, N., Dror, G., Koren, Y.: Yahoo! music recommendations: Modeling music ratings with temporal dynamics and item taxonomy. In: RecSys, pp. 165–172. ACM (2011)
19. Cremonesi, P., Koren, Y., Turrin, R.: Performance of recommender algorithms on top-n recommendation tasks. In: RecSys, pp. 39–46. ACM (2010)
20. Steck, H.: Item popularity and recommendation accuracy. In: RecSys, pp. 125–132. ACM (2011)
21. Koren, Y.: Collaborative filtering with temporal dynamics. In: KDD, pp. 447–456. ACM (2009)

22. Ma, H., Zhou, D., Liu, C., Lyu, M.R., King, I.: Recommender systems with social regularization. In: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, WSDM 11, Hong Kong, China, pp. 287–296 (2011)
23. Salakhutdinov, R., Mnih, A.: Probabilistic matrix factorization. In: Proceedings of Advances in Neural Information Processing Systems, NIPS 2007 (2007)