TopicMF: Simultaneously Exploiting Ratings and Reviews for Recommendation

Yang Bao[†] **Hui Fang** **Jie Zhang**
[†] Nanyang Business School, Nanyang Technological University, Singapore
School of Computer Engineering, Nanyang Technological University, Singapore
{baoyang@ntu.edu.sg, hfang@e.ntu.edu.sg, zhangj@ntu.edu.sg}

Abstract

Although users' preference is semantically reflected in the free-form review texts, this wealth of information was not fully exploited for learning recommender models. Specifically, almost all existing recommendation algorithms only exploit rating scores in order to find users' preference, but ignore the review texts accompanied with rating information. In this paper, we propose a novel matrix factorization model (called *TopicMF*) which simultaneously considers the ratings and accompanied review texts. Experimental results on 22 real-world datasets show the superiority of our model over the state-of-the-art models, demonstrating its effectiveness for recommendation tasks.

Introduction

Recommender systems have become a core component for today's personalized online business (e.g., Amazon), whose heart is a personalization algorithm for identifying the preference of each individual user. The most well-known algorithms utilize the collaborative filtering technique (CF), which analyzes relationships between users and interdependencies among products, in order to identify new user-item associations. Among all the CF algorithms, the most successful ones, as demonstrated by the Netflix Prize competition (Bell and Koren 2007), are the latent factor models. These models try to explain user ratings by characterizing both items and users on, say, 20 or 100 factors inferred from rating patterns. In a sense, such factors comprise a computerized alternative to the human created genes. One of the representative realizations of latent factor models are based on matrix factorization (Koren, Bell, and Volinsky 2009). In its basic form, matrix factorization characterizes both items and users by vectors of factors inferred from user-item rating matrix. High correspondence between item and user factors leads to a recommendation. Recently these methods have become popular with good scalability and predictive accuracy.

When learning the latent factor models, an assumption they take for granted is that a rating score assigned by a user to an item is determined by all factors with equal importance. Specifically, the rating is the inner product between the corresponding user and item vector. In some cases, biases for different users and items can also be incorporated (Koren, Bell, and Volinsky 2009). However, in the real scenario, this might not be the case. Based on our observations, users usually assign a rating score to an item only based on a few factors that they care specifically for this item. For example, when giving rating score to the movie *Die Hard*, different users might care about different factors. To put it in another way, factors should have different importance degrees when a user rates an item.

On the other hand, recommender systems usually predict users' preference based on their previous feedback (e.g. ratings or free-form review texts). A rating score can tell us whether a user likes or dislikes an item but cannot tell us why. In contrast, if this rating score is associated with a segment of review text, we can understand why the user likes or dislikes the item. For example, a user might give the highest rating score to the movie *Die Hard* for different reasons (e.g. fan of *Bruce Willis*, or the action movies). However, through the user's review "Bruce Willis rocks!", we can infer that this user likes this movie most possibly because he is a fan of Bruce Willis. Somewhat surprisingly, the review text has not been fully exploited for recommendation. Instead, most of the existing works on recommender systems are focused on discovering users' preferences by using the explicit rating scores while the valuable information in review texts is totally disregarded. Few studies that utilize the text information fail to connect the latent factors from textual information with those from user-item rating matrix, which greatly impacts the interpretability of the algorithms.

In this paper, aiming at bridging the two gaps mentioned above, we present a matrix factorization model, called *TopicMF*, for learning recommender models by factorizing factors using the information semantically hidden in the review texts. Specifically, we use a biased Matrix Factorization (MF) for rating prediction in recommender systems, and simultaneously adopt a topic modeling technique (i.e. Nonnegative Matrix Factorization (NMF)) to model the latent topics in review texts. We align these two tasks by using a transform from item and user latent vectors to topic distribution parameters. By doing so, we combine latent factors in rating data with topics in user-review text. Furthermore, we can cope with the different importance degrees of latent factors by adjusting the transformation function. Note that the

textual review contains richer information than a single rating score. In this case, our model can thus better handle the data sparsity (i.e. cold-start) problem than those only considering rating information. The experimental analysis on 22 real-world datasets shows that our method provides better performance than the state-of-the-art latent factor models for recommendation tasks.

Related Work

In this section, we review several related approaches to our work, including (1) latent factor-based recommender systems, (2) semantic enhanced recommender systems which have drawn a lot of attentions recently.

Due to its efficiency in dealing with large datasets and its quite effective performance for recommendation, several low-dimensional matrix approximation methods have been proposed. These methods focus on fitting the user-item rating matrix using low-rank approximations, and employ the matrix to make further predictions. The low-rank matrix factorization methods are very efficient in training since they assume that in the user-item rating matrix, only a small number of factors influence preferences, and that a user's preference vector is determined by how each factor applies to that user. Low-rank matrix approximations based on minimizing the sum-squared errors can be easily solved using Singular Value Decomposition (SVD) (Koren 2008). For example, Salakhutdinov et al. (2008a) propose a probabilistic graphical model by assuming some Gaussian observation noises on observed user-item ratings. The proposed model achieves promising prediction results. In their following work (Salakhutdinov and Mnih 2008b), the Gaussian-Wishart priors are placed on the user and item hyper-parameters. Since exact inference is intractable in the new model, a Gibbs sampling method is proposed to iteratively learn the user and item latent matrices. However, these latent factor models suffer from the general limitations: 1) the learnt latent space is not easy to interpret; 2) the assumption of equal importance of factors when generating the ratings differs from the reality.

There are also several works that attempt to combine ratings and review texts together in recommender systems, referred to as semantic enhanced recommender systems. These approaches try to interpret the latent factors factorized from the ratings. For example, Ganu et al. (2009) depend on the domain knowledge provided by human annotators to extract "*explicit*" aspects information (e.g. price) from review texts, and thus to harness rating prediction. Our work significantly differs from it as we aim to automatically learn "*implicit*" topic aspects from review text, and link them with latent factors reasonably. Also some works have considered to automatically identify review dimensions. For example, fLDA (Agarwal and Chen 2010), extended from matrix factorization, regularizes both user and item factors in a supervised manner through explicit user features and the bag of words associated with each item. It adopts discrete factors for items regularized through an LDA prior. On the contrary, in our model, both the user latent factors and item factors are optimized simultaneously with the topic parameters. Another similar work, proposed in (Wang and Blei 2011), recommends scientific articles to users based on both content of the articles and users' historic ratings. Wang et al. (2013) design a supervised topic model which simultaneously considers the textual and user-item rating information. However, these approaches differ from ours as the latent factors learned through LDA are not necessarily correlated with the user and item factors from matrix factorization on rating matrix.

The closest work to ours is proposed by McAuley and Jeskovec (2013). They relate latent rating dimensions (user and item factors) with latent review topics in their HFT model. However, the latent topics might only relate to latent user (or item) factors in the HFT, while in our approach, the latent topics correlate with user and item factors simultaneously, well reflecting the real world scenario. Besides, they learn the topics for each item (or user). In contrast, we learn the topics for each review, which can better map to users' rating behavior, and thus further increase the accuracy and interpretability of rating prediction.

Preliminaries

Problem Formulation

The problem we study is slightly different from traditional recommender systems which usually only consider the user-item rating matrix. We also take the semantic evidences into consideration. Suppose there are I users and J items. Each observed data point is a 4-tuple (i, j, r_{ij}, d_{ij}) where $i \in \{1, 2, ..., I\}$ is the user index, $j \in \{1, 2, ..., J\}$ is the item index, $r_{ij} \in \mathbb{R}$ is the rating value assigned to item j by user i, and d_{ij} is the review text that is written by user i to item j. $d_{ij} = -1$ means that there is no review text for the corresponding data point.

The problem we investigate is essentially how to effectively and efficiently predict the missing values of user-item rating matrix by employing the semantic evidences and observed user-item rating matrix.

Matrix Factorization for Recommendation

A promising method in recommender systems is to factorize the user-item rating matrix, and to utilize the factorized user-specific and item-specific matrices for further missing data prediction (Salakhutdinov and Mnih 2008a; Koren, Bell, and Volinsky 2009). In recommender systems, given $I \times J$ user-item rating matrix $R = [r_{ij}]_{I \times J}$, a low-rank matrix factorization approach seeks to approximate the rating matrix R by a multiplication of K-rank factors $R \approx U^T V$, where $U \in \mathbb{R}^{K \times I}$ and $V \in \mathbb{R}^{K \times J}$. The parameter K controls the number of latent factors for each user and item which is typically much smaller than I and J.

In this model, each user i is represented by a K-dimensional feature vector $u_i \in \mathbb{R}^K$, the i-th column of U, while each item j is represented by a K-dimensional feature vector $v_j \in \mathbb{R}^K$, the j-th column of V. The predicted rating on item j by user i is equal to the inner product of the corresponding user and item feature vectors:

$$\hat{r}_{ij} = u_i^T v_j + \mu + \beta_u(i) + \beta_v(j) \quad (1)$$

where $\beta_u(i)$ and $\beta_v(j)$ are biased terms regarding to user i and item j, respectively, and μ is the global biased term.

The objective is to compute the latent representations of the users and items given the observed rating matrix by minimizing the following regularized squared error loss:

$$\mathcal{L}_{rating} = \min_{U,V} \sum\nolimits_{i,j} c_{ij}(u_i^T v_j + \mu + \beta_u(i) + \beta_v(j) - r_{ij})^2 + \lambda_u \sum\nolimits_i \|u_i\|^2 + \lambda_v \sum\nolimits_j \|v_j\|^2 + \lambda_B(\sum\nolimits_i \beta_u^2(i) + \sum\nolimits_j \beta_v^2(j)) \quad (2)$$

where λ_u, λ_v and λ_B are the parameters controlling the strength of regularization with the purpose of avoiding overfitting, and $\|\cdot\|^2$ denotes the Frobenius norm. c_{ij} serves as a confidence parameter for rating r_{ij}, where a larger c_{ij} means that we trust r_{ij} more. $r_{ij} = 0$ can be interpreted as the user i is either not interested in item j or is unaware of it. Thus, $r_{ij} > 0$ should have higher confidence level than $r_{ij} = 0$. Similar to (Hu, Koren, and Volinsky 2008; Wang and Blei 2011), we introduce different confidence parameters c_{ij} for different rating r_{ij}. Besides, in our approach, we can further justify c_{ij} as an indicator function related to the quality of the review d_{ij}. The higher quality of the review indicates a larger c_{ij} value. Specifically, we have

$$c_{ij} = \begin{cases} a * q_{ij}, & \text{if } r_{ij} > 0 \\ b * q_{ij}, & \text{if } r_{ij} = 0 \end{cases} \quad (3)$$

where a and b are parameters satisfying $a > b \geq 0$. q_{ij} is the quality (i.e. helpfulness) of d_{ij} where $q_{ij} = 0$ if $d_{ij} = -1$ (review is unavailable). The helpfulness of each review can be moderately computed if the helpfulness votes are not available (Ghose and Ipeirotis 2011).

The locally optimal solution of U and V can be found usually with an iterative algorithm (Hu, Koren, and Volinsky 2008), where we update U and V alternately while holding the other matrix fixed. We can then use Equation 1 to predict the missing ratings of the items.

Topic Modeling

Topic modeling techniques are employed to uncover hidden "topics" in review text, where a topic is a distribution over terms (i.e. words) that is biased around those associated under a single theme. The simplest topic model - Latent Dirichlet Allocation (LDA) (Blei, Ng, and Jordan 2003) is not suitable for our problem. Instead, we use NMF for our research, since it estimates the probability distribution of each document on the hidden topics independently (Cai et al. 2008). This is the case for our research, as we bound the probability distribution of each review on the hidden topics with the corresponding latent user and item factors.

Given review dataset $[d_{ij}]_{I\times J}$ and N words (each word $n \in \{1, 2, \cdots, N\}$), let $W = [W_{d_{ij}n}]_{IJ\times N}$ denote the *word-to-review* matrix: $F_{d_{ij}n}$ is the frequency of word n in review d_{ij}. We further re-scale $W_{d_{ij}n} = W_{d_{ij}n}/T_N$, where T_N is the total number of words. Different from MF, NMF has the constraint that all entries in the decomposed factors have to be non-negative. NMF tries to find two non-negative matrix factors $\Theta \in \mathbb{R}^{IJ\times K}$ and $\Phi \in \mathbb{R}^{N\times K}$ (K is constraint to be equivalent to the number of factors in Matrix Factorization) where the product of the two matrices is an approximation of the original matrix:

$$F \approx \Theta\Phi^T \quad (4)$$

where $\Theta = (\theta_{d_{ij}k})$, $\Phi = (\phi_{nk})$, and $\theta_{d_{ij}k}$, $\phi_{nk} \geq 0$. They are determined by minimizing:

$$\mathcal{L}_{review} = \min_{\Theta,\Phi} ||\Theta\Phi^T - W||^2 = \min_{\Theta,\Phi} \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{n=1}^{N} c_{ij}(\theta_{d_{ij}}\phi_n^T - W_{d_{ij}n})^2 \quad (5)$$

where $\phi_n \in \mathbb{R}^K$ is the n-th row of Φ, and $\theta_{d_{ij}} \in \mathbb{R}^K$ is the ij-th row of Θ.

The TopicMF Model

Our TopicMF model tries to combine the idea of MF for rating prediction and NMF for uncovering latent topic factors in review texts. Particularly, we correlate the topic factors with the corresponding latent factors of both users and items.

To be more specific, as demonstrated in the Preliminaries section, we learn a topic distribution ($\theta_{d_{ij}}$) for each review d_{ij}. This actually records the extent to which each of K topics is talked by user i for item j.

Given our motivation of linking a user's rating pattern with her review pattern, we need to dependently learn the rating parameters and review parameters. On the one hand, v_j represents a set of certain properties that item j occupies, while u_i represents user i's preference towards corresponding item features. On the other hand, $\theta_{d_{ij}}$ encodes user i's preference on the item j. Therefore, we define that the $\theta_{d_{ij}}$ is simultaneously influenced by u_i and v_j (see Figure 1).

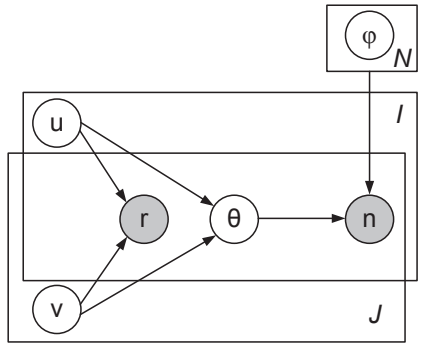

Figure 1: The relationship of parameters.

More importantly, each component of $\theta_{d_{ij}}$, e.g. $\theta_{d_{ij}k}$, is expected to be positively correlated with both the corresponding user factor and item factor (e.g. u_{ik} and v_{ik}). That is, if an item has higher *absolute*[1] value of a certain factor, or a user values higher on a certain factor (in terms of the *absolute* value), the corresponding factor is expected to be discussed more in the review. In order to capture such dependent correlation, we propose the following addition transformation function (abbreviated as *A-T*):

$$\theta_{d_{ij}k} = \frac{\exp(\kappa_1|u_{ik}| + \kappa_2|v_{jk}|)}{\sum_{k'=1}^{K}\exp(\kappa_1|u_{ik'}| + \kappa_2|v_{jk'}|)} \quad (6)$$

where parameters κ_1 and κ_2 are introduced to moderate the transformation. Intuitively, larger κ_1 means that users are

[1] In the HFT model (McAuley and Leskovec 2013), different from our model, it ignores those factors of negative bigger values, which might also be talked a lot in the corresponding reviews but in a relatively negative mood.

more attracted by the dominant properties of target items (like or dislike), while smaller κ_1 means that users tend to evenly talk about all the features of target items. Similarly, large κ_2 means that users always tend to talk about the highly important factors they care, while smaller one means that users care about all the properties and would like to discuss them in the reviews. Formally, $\kappa_1, \kappa_2 \to 1$, $\theta_{d_{ij}}$ will approach a unit vector that takes the value 1 only for the largest indexing value of u_i or v_j. Conversably, if $\kappa_1, \kappa_2 \to 0$, $\theta_{d_{ij}}$ approaches an uniform distribution. Note that through the control of κ_1 and κ_2, we fairly achieve one of our goals of relaxing the assumption of equal importance of factors.

Equation 7 is the multiplication form of transformation function (abbreviated as *M-T*) that also captures the monotonic relationship of topic factors with user and item factors.

$$\theta_{d_{ijk}} = \frac{\exp(\kappa|u_{ik}.v_{jk}|)}{\sum_{k'=1}^{K}\exp(|\kappa u_{ik'}.v_{jk'}|)} \tag{7}$$

where κ serves the same purpose as κ_1 and κ_2 in Equation 6. The two latent spaces in rating matrix R and review matrix W are thus connected via our proposed two transformation functions. Besides, the latent space in review matrix can provide semantic descriptions (i.e. topics) for each latent factor in rating matrix.

The objective of our model is to learn the optimal U and V for accurately modeling users' ratings, but at the same time, obtain most likely topics according to reviews with the constraint of transformation function. Thus, we reach the following objective function for our TopicMF model using NMF for uncovering hidden topics in Equation 5:

$$\begin{aligned}\mathcal{L} =& \mathcal{L}_{rating} + \lambda\mathcal{L}_{review} = \sum_i^I\sum_j^J c_{ij}(u_i^T v_j + \mu + \beta_u(i) + \beta_v(j) \\ & - r_{ij})^2 + \lambda\sum_i^I\sum_j^J\sum_n^N c_{ij}(\theta_{d_{ij}}\phi_n^T - W_{d_{ijn}})^2 + \lambda_u\sum_i^I u_i^T u_i \\ & + \lambda_v\sum_j^J v_j^T v_j + \lambda_B\Big(\sum_i^I\beta_u(i)^2 + \sum_j^J\beta_v(j)^2\Big)\end{aligned} \tag{8}$$

where λ is a parameter that balances the performance of rating prediction and topic modeling. Gradient descent is conducted to update U, V, Θ, Φ, and κ. The details of using Equation 6 (referred as **TopicMF-AT**) to compute the corresponding gradients given Equation 9 are listed as follows[2]:

$$\begin{aligned}&\frac{1}{2}\frac{\partial\mathcal{L}}{\partial u_i} = \sum_{j=1}^{J} c_{ij}v_j(u_i^T v_j + \mu + \beta_u(i) + \beta_v(j) - r_{ij}) + \lambda_u u_i \\ &+ \lambda\Big[\sum_{j=1}^{J} c_{ij}\sum_{n=1}^{N}\kappa_1(\theta_{d_{ij}}\phi_n^T - W_{d_{ijn}})\theta_{d_{ij}}^T .*(1-\theta_{d_{ij}}^T).*\phi_n^T.*(u_i./|u_i|)\Big] \\ &\frac{1}{2}\frac{\partial\mathcal{L}}{\partial v_j} = \sum_{i=1}^{I} c_{ij}u_i(u_i^T v_j + \mu + \beta_u(i) + \beta_v(j) - r_{ij}) + \lambda_v v_j \\ &+ \lambda\Big[\sum_{i=1}^{I} c_{ij}\sum_{n=1}^{N}\kappa_2(\theta_{d_{ij}}\phi_n^T - W_{d_{ijn}})\theta_{d_{ij}}^T .*(1-\theta_{d_{ij}}^T).*\phi_n^T.*(v_j./|v_j|)\Big] \\ &\frac{1}{2}\frac{\partial\mathcal{L}}{\partial\kappa_1} = \lambda\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{n=1}^{N} c_{ij}(\theta_{d_{ij}}\phi_n^T - W_{d_{ijn}})\theta_{d_{ij}}.*\phi_n(|u_i| - \theta_{d_{ij}}|u_i|) \\ &\frac{1}{2}\frac{\partial\mathcal{L}}{\partial\kappa_2} = \lambda\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{n=1}^{N} c_{ij}(\theta_{d_{ij}}\phi_n^T - W_{d_{ijn}}\theta_{d_{ij}}.*\phi_n(|v_j| - \theta_{d_{ij}}|v_j|)\end{aligned} \tag{9}$$

[2]Due to the space limitation, the gradients of the biased terms are not listed in the paper.

Accordingly, the details of gradient descent of using Equation 7 (referred as **TopicMF-MT**) is:

$$\begin{aligned}&\frac{1}{2}\frac{\partial\mathcal{L}}{\partial u_i} = \sum_{j=1}^{J} c_{ij}v_j(u_i^T v_j + \mu + \beta_u(i) + \beta_v(j) - r_{ij}) + \lambda_u u_i + \\ &\lambda\Big[\sum_{j=1}^{J} c_{ij}\sum_{n=1}^{N}\kappa(\theta_{d_{ij}}\phi_n^T - W_{d_{ijn}})|v_j|.*\theta_{d_{ij}}^T.*(1-\theta_{d_{ij}}^T).*\phi_n^T.*(u_i./|u_i|)\Big] \\ &\frac{1}{2}\frac{\partial\mathcal{L}}{\partial v_j} = \sum_{i=1}^{I} c_{ij}u_i(u_i^T v_j + \mu + \beta_u(i) + \beta_v(j) - r_{ij}) + \lambda_v v_j + \\ &\lambda\Big[\sum_{i=1}^{I} c_{ij}\sum_{n=1}^{N}\kappa(\theta_{d_{ij}}\phi_n^T - W_{d_{ijn}})|u_i|.*\theta_{d_{ij}}^T.*(1-\theta_{d_{ij}}^T).*\phi_n^T.*(v_j./|v_j|)\Big] \\ &\frac{1}{2}\frac{\partial\mathcal{L}}{\partial\kappa} = \lambda\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{n=1}^{N} c_{ij}\Big[(\theta_{d_{ij}}\phi_n^T - W_{d_{ijn}})\times \\ &\qquad\theta_{d_{ij}}.*\phi_n\Big(|u_i|.*|v_j| - \theta_{d_{ij}}(|u_i|.*|v_j|)\Big)\Big]\end{aligned} \tag{10}$$

Recall that our goal is to simultaneously optimize the parameters associated with ratings (i.e. U and V), and the parameters associated with topics (i.e. Θ and Φ). And, Θ are fit by U and V. As presented above, U and V are fit by gradient descent in Equation 9 or 10, while Φ is updated through Equation 5. Therefore, we design a procedure that alternates between the following two steps:

$$\begin{aligned}&\text{update } U, V, \Theta, \kappa = \underset{U,V,\Theta,\kappa}{\text{argmin}}\ \mathcal{L} \\ &\text{update } \Phi = \underset{\Phi}{\text{argmin}}\ \mathcal{L}_{review}\end{aligned} \tag{11}$$

For the first step of Equation 11, we update these paramters through the commonly used non-linear optimization package, L-BFGS[3]. In NMF, the update of Φ can be accomplished through the well-known projected gradients technique (Lin 2007). Since we fix the Θ in the second step, the update of Φ is reduced to a sub-problem of Algorithm 2 of the NMF as described in (Lin 2007). Thus, we particularly adopt the source code regarding to the sub-problem of updating Φ when fixing Θ.

Experiments

In this section, we empirically evaluate the effectiveness of our model of adopting reviews for recommendation on 22 Amazon datasets, and compare it with three state-of-the-art approaches.

Datasets

To make the fair evaluation of our model, we directly generate the 22 Amazon datasets from the datasets provided by (McAuley and Leskovec 2013). Each dataset contains a category of products sold on the Amazon. In particular, due to our hardware limitation, for some large datasets (over GB), we sample a portion of the whole dataset by limiting the number of items up to 5000, respectively. The statistic information is summarized in Table 1. Besides, we randomly subdivide each dataset into the training and testing sets, where 80% of each dataset is used for training, and the rest is for testing (see Table 1). Note that we only utilize the existing reviews in the training dataset for model learning.

[3]www.chokkan.org/software/liblbfgs/

Table 1: Data description

dataset	#users	#items	#ratings (training)	#ratings (testing)	avg. rating
arts	24,069	4,207	22,158	5,496	4.1280
automotive	89,246	5,000	94,261	23,567	4.1873
baby	13,928	1,651	12,840	3,129	3.9543
beauty	129,026	5,000	149,510	37,434	4.1581
cell accessories	66,185	5,000	59,518	14,556	3.5202
clothing	73,276	5,000	307,045	76,733	4.2252
pet supplies	144,725	5,000	152,251	37,641	4.0715
gourmet foods	88,651	5,000	91,660	22,888	4.2523
health	235,925	5,000	241,037	60,710	4.1031
home and kitchen	448,574	5,000	482,828	120,170	4.0088
industrial &scientific	21,716	5,000	74,482	18,650	4.4107
jewelry	31,597	5,000	136,623	34,285	3.9577
musical instrument	58,095	5,000	56,976	14,055	4.1820
office products	98,141	5,000	93,091	23,076	3.9756
patio	143,846	5,000	136,623	34,285	3.9577
shoes	48,310	5,000	221,371	55347	4.3639
software	62,764	5,000	59,239	14,690	3.3038
sports &outdoors	218,938	5,000	248,112	61,784	4.1826
tools&home	198,290	5,000	207,028	51,452	4.0685
toys	194,347	5,000	194,231	48,706	4.1199
video games	194,614	5,000	236,891	59,431	4.0065
watches	57,283	5,000	49,751	12,369	4.1624

Baselines and Evaluation Metric

For the competing methods, the HFT (item) (McAuley and Leskovec 2013) is so far the best among the methods that attempt to exploit unstructured textual reviews for recommendation. We therefore choose it as our benchmark method[4]. The HFT (item) refers to the one that topics in review text are associated with item parameters, which is proved to have better performance than HFT (user) in terms of rating predictive accuracy. We also conduct comparisons with other state-of-the-art methods, including PMF (Salakhutdinov and Mnih 2008a) which only exploits rating information, and SVD++ (Koren 2008) that exploits both rating information and other hidden feedback (e.g. click information). We adopt the source codes of these two methods provided by MyMediaLite Recommender System Library[5]. For all the methods, we set optimal parameters recommended in the literature, and set the number of latent factors (K) as 5.

To measure the performance of these methods, we adopt the commonly used metric MSE (the mean squared error), which is defined as the average of the squared error between the predictions and the ground-truth. Smaller MSE values normally indicate better performance for rating prediction.

[4] i.stanford.edu/~julian

[5] www.mymedialite.net/index.html

Performance Comparisons

Here, we show the performance comparison of our proposed TopicMF model with all baseline methods. The results are shown in Table 2, where the best performance is in bold font. "*" indicates the better performance between TopicMF-AT and TopicMF-MT on each dataset. For HFT (item) and our method, we report the MSE value after 20 iterations (see Equation 11). For our method, the balance parameter λ is set to 1, and $\lambda_u = \lambda_v = \lambda_B = 0.001$, while other parameters are fit using L-BFGS package and projected gradient technique[6]. In order to make fair comparison with HFT (item), our c_{ij} value does not take the review quality into consideration. Instead, we set $c_{ij} = 1$ if $r_{ij} \neq 0$.

As can be seen in Table 2, our method (TopicMF-AT or TopicMF-MT) performs much better than HFT (item) on most categories (e.g. baby, video games and gourmet foods), and can achieve comparable performance (i.e. the difference is around or less than 1%) on the rest categories (e.g. shoes and sports&outdoors). Under the categories like baby, video games and gourmet foods, users are more enthusiastic, and would like to reveal and express their subjective feelings and attitudes towards the products they have interest, such as baby products, video games or favorite foods. Therefore, more meaningful topics are expected to be derived from each single review regarding to the items of these categories. Not surprisingly, the methods that also exploit unstructured reviews, i.e. TopicMF and HFT (item), perform much better than PMF and SVD++.

On average, as presented in Table 3, across all the datasets, PMF achieves an MSE of 1.5585, SVD++ achieves an MSE of 1.4395, while HFT (item) achieves an MSE of 1.3836. TopicMF-MT performs slightly better than TopicMF-AT, and they achieve MSE of 1.3468 and 1.3511, respectively. Besides, TopicMF-MT gains improvements over SVD++ and HFT (item) up to 6.43% and 2.73%, respectively. We also conduct t-test for the performance difference across 22 datasets, and the results (see Table 3) show that the performance improvement of our method is statistically significant at the 5% level (i.e., p-value < 0.05).

Table 3: The average performance of different methods.

method		**average MSE**
PMF		1.5585
SVD++		1.4393
HFT (item)		1.3836
TopicMF-AT		1.3511
TopicMF-MT		**1.3468**
improvement	vs. SVD++	6.43%
	p-value	0.0000
	vs. HFT(item)	2.73%
	p-value	0.0001

Parameter Sensitivity

There are two important parameters in our TopicMF model: 1) the number of latent factors K; and 2) the parameter λ

[6] www.csie.ntu.edu.tw/~cjlin/nmf

Table 2: Performance comparisons of different methods ($K = 5$).

method		arts	automotive	baby	beauty	cell accessories	clothing	pet supplies	gourmet foods	health	home and kitchen	industrial &scientific
PMF		1.7604	1.5359	1.9473	1.4613	2.3142	0.2893	1.7559	1.5536	1.6732	1.7721	0.4414
SVD++		1.6422	1.4031	1.6699	1.5460	2.1646	0.4493	1.6123	1.4921	1.5842	1.5733	0.4261
HFT (item)		1.4931	1.3762	1.7180	**1.3059**	2.2224	0.2250	1.5824	1.5245	1.4910	1.5382	**0.3514**
TopicMF-AT		1.4820	**1.3364***	1.6676	1.3216	**2.1590***	0.2290	**1.5256***	**1.4297***	**1.4793***	1.5298	0.3540
TopicMF-MT		**1.4749***	1.3377	**1.6212***	1.3193*	2.1601	**0.2215***	1.5257	1.4319	1.4811	**1.5232***	0.3523*
improve	vs. SVD++	10.2%	4.75%	2.92%	14.7%	0.26%	50.7%	5.38%	4.18%	6.62%	3.18%	17.3%
	vs. HFT(item)	1.22%	2.89%	5.63%	-1.03%	2.85%	1.56%	3.59%	6.22%	0.78%	0.98%	-0.24%

method		jewelry	musical instrument	office products	patio	shoes	software	sports& outdoors	tool& home	toys	video games	watches
PMF		1.4239	1.5365	1.9026	1.9826	0.2335	2.7219	1.2304	1.7158	1.6596	1.8030	1.5733
SVD++		1.3670	1.4279	1.7342	1.7556	0.4161	2.3420	1.1912	1.5242	1.3699	1.4630	1.5098
HFT (item)		1.2587	1.4577	1.5898	1.7474	**0.1560**	2.3223	**1.0751**	1.4845	1.4008	1.5692	1.5493
TopicMF-AT		**1.2137***	1.3820	1.5945	1.7096	0.1569*	2.2932	1.0787*	1.4670	1.3680	**1.4453***	**1.5022***
TopicMF-MT		1.2370	**1.3806***	**1.5891***	**1.6941***	0.1599	**2.2816***	1.0804	**1.4553***	**1.3505***	1.4468	1.5056
improve	vs.SVD++	11.2%	3.31%	8.37%	3.50%	62.29%	2.58%	9.44%	4.52%	1.42%	1.21%	0.51%
	vs.HFT(item)	3.58%	5.29%	0.04%	3.05%	-0.59%	1.75%	-0.33%	1.97%	3.59%	7.89%	3.04%

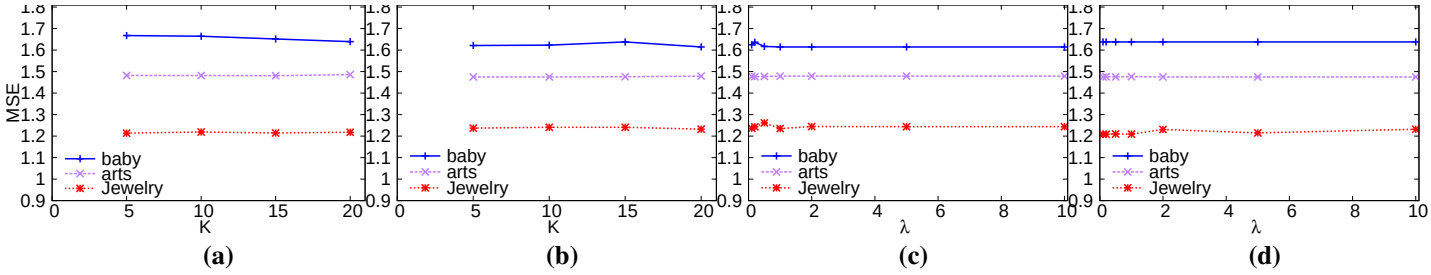

Figure 2: Performance by varying number of latent factors (K) (a) TopicMF-AT; and (b) TopicMF-MT; Performance by varying parameter λ (c) TopicMF-AT; and (d)TopicMF-MT.

that balances the performance of rating prediction and topic modeling. Their optimal values cannot be automatically fit by our algorithms. To investigate the effects of these two parameters, we show the performance of our models by varying one of them meanwhile fixing the other one on the following three datasets: "arts", "baby" and "jewelry".

Firstly, we fix λ to its default value 1, and vary the number of latent factors to be 5, 10, 15, and 20, respectively. As shown in Figures 2(a) and (b), we can see that the performance of our two models is relatively stable, indicating that our models are not sensitive to the K value. This is different from the traditional latent factor models which are inclined to use more factors (Koren, Bell, and Volinsky 2009). This finding on the whole the same as (McAuley and Leskovec 2013). This is mainly because in a review, a user might only mention a few of all the possible factors.

Next, we fix the parameter K to 5, and vary λ to be values of $\{0.1, 0.2, 0.5, 1, 2, 5, 10\}$. The results are shown in Figures 2(c) and (d). As can be seen, the performance is relatively stable, and only have slight fluctuations as λ varies, especially when $\lambda \geq 1$. This demonstrates that our methods are not very sensitive to parameter λ.

To summarize, given the insensitivity of parameters λ and K, and automatical optimization of other parameters, our TopicMF model owns good flexibility.

Conclusion and Future Work

In this paper, we propose a latent factor model, called TopicMF, for recommendation by jointly considering user ratings and unstructured reviews. Specifically, we use a biased matrix factorization model that factorizes user-item rating matrix into latent user and item factors for rating prediction. Simultaneously, we employ the non-negative matrix factorization technique to derive topics from user unstructured review text. We relate these two tasks by designing the *A-T* or *M-T* transform function to align the topic distribution parameters with the corresponding latent user and item factors. In this case, we can further improve the accuracy of rating prediction. We conduct experiments on 22 real-world datasets, each of which contains the items of a category sold on Amazon. Experimental results demonstrate that our model outperforms the three state-of-the-art methods (i.e. PMF, SVD++, and HFT), and can achieve better performance for rating prediction.

In the future, we intend to quantitatively analyze the learned topics for each review, and explore how well these topics interpret users' rating behavior. Given these topics, we will construct user preference models, and then empirically validate them.

Acknowledgements

This work is supported by the MoE AcRF Tier 2 Grant M4020110.020 and the ACI Seed Funding M4080962.C90 awarded to Dr. Jie Zhang.

References

Agarwal, D., and Chen, B.-C. 2010. fLDA: matrix factorization through latent dirichlet allocation. In *Proceedings of the 3rd ACM International Conference on Web Search and Data Mining (WSDM)*, 91–100.

Bell, R. M., and Koren, Y. 2007. Lessons from the netflix prize challenge. *ACM SIGKDD Explorations Newsletter* 9(2):75–79.

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research* 3:993–1022.

Cai, D.; Mei, Q.; Han, J.; and Zhai, C. 2008. Modeling hidden topics on document manifold. In *Proceedings of the 17th ACM conference on Information and Knowledge Management*, 911–920. ACM.

Ganu, G.; Elhadad, N.; and Marian, A. 2009. Beyond the stars: Improving rating predictions using review text content. In *WebDB*.

Ghose, A., and Ipeirotis, P. G. 2011. Estimating the helpfulness and economic impact of product reviews: Mining text and reviewer characteristics. *IEEE Transactions on Knowledge and Data Engineering* 23(10):1498–1512.

Hu, Y.; Koren, Y.; and Volinsky, C. 2008. Collaborative filtering for implicit feedback datasets. In *Proceedings of the Eighth IEEE International Conference on Data Mining (ICDM)*, 263–272. IEEE.

Koren, Y.; Bell, R. M.; and Volinsky, C. 2009. Matrix factorization techniques for recommender systems. *IEEE Computer* 42(8):30–37.

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*, 426–434. ACM.

Lin, C.-J. 2007. Projected gradient methods for nonnegative matrix factorization. *Neural Computation* 19(10):2756–2779.

McAuley, J., and Leskovec, J. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM cConference on Recommender Systems (RecSys)*, 165–172. ACM.

Salakhutdinov, R., and Mnih, A. 2008a. Probabilistic matrix factorization. *Advances in Neural Information Processing Systems (NIPS)* 20:1257–1264.

Salakhutdinov, R., and Mnih, A. 2008b. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th International Conference on Machine Learning (ICML)*, 880–887. ACM.

Wang, C., and Blei, D. M. 2011. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 448–456.

Wang, S.; Li, F.; and Zhang, M. 2013. Supervised topic model with consideration of user and item. In *Workshops at the Twenty-Seventh AAAI Conference on Artificial Intelligence*.