12-2015

A Bayesian Recommender Model for User Rating and Review Profiling

Mingming JIANG

Dandan SONG

Lejian LIAO

Feida ZHU
Singapore Management University, fdzhu@smu.edu.sg

TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 11/13 pp634-643
Volume 20, Number 6, December 2015

A Bayesian Recommender Model for User Rating and Review Profiling

Mingming Jiang, Dandan Song*, Lejian Liao, and Feida Zhu

Abstract: Intuitively, not only do ratings include abundant information for learning user preferences, but also reviews accompanied by ratings. However, most existing recommender systems take rating scores for granted and discard the wealth of information in accompanying reviews. In this paper, in order to exploit user profiles' information embedded in both ratings and reviews exhaustively, we propose a Bayesian model that links a traditional Collaborative Filtering (CF) technique with a topic model seamlessly. By employing a topic model with the review text and aligning user review topics with "user attitudes" (i.e., abstract rating patterns) over the same distribution, our method achieves greater accuracy than the traditional approach on the rating prediction task. Moreover, with review text information involved, latent user rating attitudes are interpretable and "cold-start" problem can be alleviated. This property qualifies our method for serving as a "recommender" task with very sparse datasets. Furthermore, unlike most related works, we treat each review as a document, not all reviews of each user or item together as one document, to fully exploit the reviews' information. Experimental results on 25 real-world datasets demonstrate the superiority of our model over state-of-the-art methods.

Key words: collaborative filtering; topic model; recommender system; matrix factorization

1 Introduction

Recommender systems are rapidly becoming one of the most crucial functions for e-commerce platforms (e.g., Amazon and Netflix), enabling them to make accurate personalized recommendations to individual customers. They have achieved great success in business. The strategic importance of highly accurate recommender algorithms has inspired fruitful research in the past decade. The most successful such work yielded the Collaborative Filtering (CF) technique, which uncovers users' preferences by analyzing the underlying relationship between users or items they like through users' past behavior. Among all existing CF-based approaches, the most successful ones are latent factor models, which comprise alternative approaches to collaborative filtering, with the more holistic goal of uncovering latent features that explain observed ratings[1]. Latent factor models are usually implemented by applying matrix factorization techniques[2] to the rating matrix; then users and items are represented by corresponding latent-factor vectors. Finally, those unobserved ratings are predicted by the inner product of the corresponding user and item latent-factor vectors. Therefore learning these latent factors is the key to improving recommender systems' prediction accuracy.

Most CF-based recommender systems suffer from the sparseness problem, for even the most active users purchase only a limited number of items, which yields a rating matrix that is extremely sparse. This means that there is no enough user preference information that can be learned through the rating matrix to provide good results. In order to deal

• Mingming Jiang, Dandan Song, and Lejian Liao are with the School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China. E-mail: jmm@bit.edu.cn; sdd@bit.edu.cn; liaolj@bit.edu.cn.
• Feida Zhu is with the School of Information Systems, Singapore Management University, Singapore 178902. E-mail: fdzhu@smu.edu.sg.
∗ To whom correspondence should be addressed.
Manuscript received: 2015-10-07; accepted: 2015-11-05

with this issue, some probabilistic algorithms[3–7] are studied, which scale linearly with the number of observations and perform well on very sparse and unbalanced datasets. On the other hand, when dealing with new users or items, most recommender approaches cannot make any recommendations. This issue is known as the "cold-start" problem[8]. More generally, the solution is learning a hybrid model that combines collaborative filtering and content-based filtering. By considering both a rating preference matrix and additional information about users or items (e.g., items' content, users' age, etc.), hybrid methods[9, 10] can make better recommendations for new users or items.

Moreover, most recommender systems learn users' preferences through previous explicit feedback (e.g., ratings), and predict missing ratings. However, a rating indicates only whether a user liked or disliked an item, but does not express why. On most e-commerce platforms (e.g., Amazon), customers usually give a rating score accompanied by review texts to purchased items. There is a wealth of information about why a user likes or dislikes an item in those reviews, which is very helpful for discovering user preferences and expressed ratings. Furthermore, with review text information, recommender systems can alleviate the sparseness and "cold-start" problems discussed above. However, most existing recommender algorithms only focus on explicit ratings, and discard the valuable information embedded in reviews.

In this paper, we aim to combine collaborative filtering and content-based filtering to learn user rating and review preferences more accurately. To this end, we proposed a Bayesian model, called User Rating and Review Profile (URRP), which links User Rating Profile (URP)[3, 7] with Latent Dirichlet Allocation (LDA)[11] seamlessly. There are several excellences of our URRP model. First, URRP can handle the sparseness and "cold-start" problems through considering both ratings and reviews. Moreover, URRP is able to express ratings by introducing review topics into learning user rating behaviors. And unlike most related works, URRP treats each review as a document, but in the context of all reviews of each user or item, to fully exploit review information. Finally, experimental results on 25 real-world datasets demonstrate the superiority of URRP over those state-of-the-art methods.

2 Related Work

In this section, we review some methods related to our work from (1) CF-based recommender systems; and (2) hybrid approaches combining collaborative filtering and content-based filtering.

Recently, CF-based recommender system implementations using latent factor models and matrix factorization have become very popular, due to their good scalability and predictive accuracy. For example, Refs. [1, 12] employed Singular Value Decomposition (SVD) to provide a low-rank approximation of the original rating matrix, and add additional biases to the regularized SVD model in order to avoid overfitting. Some recent works[4, 5, 7] have also studied probabilistic methods in recommender scenarios, which scale linearly with the number of observations and perform well on large, sparse, and unbalanced datasets. However, these approaches only focus on learning user preferences by explicit ratings and discard valuable information embedded in review text, which makes them suffer from the "cold-start" problem. Furthermore, these CF-based methods can only predict users' rating scores on items, but are unable to express reasons for the ratings, which is typically a big drawback.

Several recent works combined collaborative filtering and content-based filtering. In Ref. [9], Collaborative Topic Regression (CTR) was proposed to recommend scientific articles to readers. CTR combines the merits of traditional collaborative filtering and LDA, and provides an interpretable latent structure for users and items. But as McAuley and Leskovec[13] pointed out, the dimensions learned by CTR are not necessarily correlated with ratings. They then proposed the Hidden Factors and Hidden Topics (HFT) model, which combines latent rating dimensions (learned by a latent factor model) with latent review topics (learned by LDA) by defining a transformation function to link the two. Their experimental results demonstrate HFT's superiority over recommender models that discard review text information. Reference [14] extended HFT by defining a different transformation function to link not only item latent factors but also user latent factors with review topics, which better reflects the real world scenario. However, as pointed out in Ref. [15], the transformation function in Ref. [13] fixes the relationship between latent vector and the topic distribution, which makes it difficult to ensure their scalability. In Ref. [15], Ratings Meet Reviews (RMR) applies LDA to the review text, and aligns the topics with latent item factors to improve recommendation accuracy. In RMR, ratings and review text are connected by the same item topic distribution $\boldsymbol{\theta}$. However, RMR operates on items — all reviews of each item are treated as a "document" of this item. This does not reflect the real world scenario, since each rating is accompanied with a review that has different topic distributions. In spite of the fact that the model proposed in Ref. [14] also learns the topic distribution for each review, the transformation function in this model fixes the relationship between latent vector and the topic distribution as mentioned above.

In contrast, we learn user topics for each review and user attitudes for each rating based on the same multinomial distribution, which links user rating behaviors and review patterns seamlessly, and thus increases the accuracy and interpretability of our rating predictions.

3 Preliminaries

Before defining our own model, we first formulate the recommendation task we study, then briefly review URP as a recommender system and LDA for topic modeling of review text. The main notations we use in the following sections are listed in Table 1.

3.1 Problem formulation

In general, the problem that traditional recommender systems investigate is how to predict missing values in rating matrices $\boldsymbol{R}$ through observed ratings. What we study in this paper is different; we take the review text that accompanies each observed rating for granted. Specifically, suppose there are M users and N items. Each observation is a 4-tuple – $(u, v, r_{u,v}, d_{u,v})$, where $u \in \boldsymbol{U}$ is the user index, $v \in \boldsymbol{V}$ is the item index, $r_{u,v}$ is the rating score for item v by user u and $d_{u,v} \in \boldsymbol{D}$ is the review text written by user u for item v accompanied by rating $r_{u,v}$. The problem we investigate in this paper is how to make more accurate predictions of missing values in rating matrix $\boldsymbol{R}$ by modeling user ratings and review profiles simultaneously.

3.2 Latent Dirichlet allocation

In Ref. [11], Blei et al. proposed LDA, which is a generative probabilistic model of a text corpus. The basic idea is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words. In our scenario, we treat each review as a document and assume there are S reviews and K topics, the generative process of LDA is as follows.

Table 1 Summary of notations.

Symbol	Description
M	Number of users
N	Number of items
S	Number of ratings or reviews
K	Number of latent user topics or attitudes
W	Number of words
$\boldsymbol{\theta}$	$M \times K$ matrix: user topics' or attitudes' distribution
$\boldsymbol{\phi}$	$K \times W$ matrix: topics distribution over words
$\boldsymbol{\xi}$	$K \times N \times S$ matrix: attitude and item's distribution over ratings
$r_{u,v}$	Rating score for item v by user u
$d_{u,v}$	Review text written by user u for item v
$L_{u,v}$	Number of words in review text $d_{u,v}$
$\boldsymbol{Z}$	$M \times N \times W$ matrix: topic-assignments for each word
$\boldsymbol{Z}_{\neg i}$	$M \times N \times W$ matrix: topic-assignments for each word exclude observation $i = \{u, v, w, d_{u,v}^{w}\}$
$\boldsymbol{X}$	$M \times N$ matrix: attitude-assignments for each rating
$\boldsymbol{X}_{\neg j}$	$M \times N$ matrix: attitude-assignments for each rating exclude observation $j = \{u, v, r_{u,v}\}$
$n_{k,w}$	Times that topic k has been assigned to word w
n_u^k	Times that topic k has been assigned to words in reviews written by user u
m_u^k	Times that attitude k assigned to ratings by user u
$c_{k,v}^s$	Times that rating s assigned to item v for attitude k

(1) For each latent topic dimension $k \in [1, K]$:
  (a) Sample word distribution $\boldsymbol{\phi}_k \sim$ Dirichlet($\boldsymbol{\beta}$).
(2) For each review $d \in \{d_1, d_2, \cdots, d_S\}$:
  (a) Sample topic distribution $\boldsymbol{\theta} \sim$ Dirichlet($\boldsymbol{\alpha}$).
  (b)For each word w in d,
    (i) Sample topic assignment $z \sim$ Multinomial($\boldsymbol{\theta}$).
    (ii) Sample word $w \sim$ Multinomial($\boldsymbol{\phi}_z$).

The corresponding graphic model is shown in Fig. 1. Based on the LDA assumption, each document has its own topic proportions on all topics, which conforms to the real scenario. LDA associates each review d with a K-dimensional topic distribution $\boldsymbol{\theta}$, which encodes the fraction of words in d that discuss each of the K topics[13]. LDA is not only a topic model for text

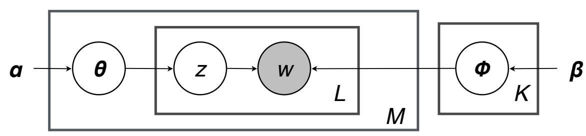

Fig. 1 Graphical model of LDA.

analysis, but also can be easily applied on other research tasks, such as human behavior recognition[16]. In our task, users discuss several topics of an item in a review, which makes LDA appropriate to uncover latent topics in review text.

3.3 User rating profile

The URP model proposed in Ref. [3] is an extension of LDA for collaborative filtering. In URP, each user is represented as a mixture of so-called *user attitudes*; the rating for each item is generated by selecting a user attitude for the item, and then sampling a rating according to the preference pattern associated with that attitude. Barbieri et al.[7] extended URP by employing a Dirichlet prior over the rating distribution on items and attitude. In this paper, we focus on this refined version of URP. A graphic model of URP is shown in Fig. 2. And URP is characterized by the following generative process:

(1) For each user attitude $x \in [x_1, x_2, \cdots, x_K]$,
(a) For each item $v \in \boldsymbol{V}$,
(i) Sample rating probabilities $\boldsymbol{\xi}_{x,v} \sim \text{Dirichlet}(\boldsymbol{\lambda})$.
(2) For each user $u \in \boldsymbol{U}$,
(a) Sample user attitude distribution $\boldsymbol{\theta}_u \sim \text{Dirichlet}(\boldsymbol{\alpha})$.
(b) For each item $v \in \boldsymbol{V}(u)$,
(i) Sample a user attitude $x \sim \text{Multinomial}(\boldsymbol{\theta}_u)$.
(ii) Sample rating value $r_{u,v}$ for chosen item v according to $r \sim \text{Multinomial}(\boldsymbol{\xi}_{x,v})$.

4 The URRP Model

In this section, we define the URRP model, which links the URP model for a recommender system with LDA for topic modeling seamlessly.

Suppose there are M users $\boldsymbol{U} = \{u_1, u_2, \cdots, u_M\}$, N items $\boldsymbol{V} = \{v_1, v_2, \cdots, v_N\}$, a set of observed ratings $\boldsymbol{R} = [r_{u,v}]_{M \times N}$, each accompanied by a review $\boldsymbol{D} = [d_{u,v}]_{M \times N}$, where 0 denotes unobserved values in $\boldsymbol{R}$ and $\boldsymbol{D}$, and each review text $d_{u,v}$ has a bag of words with length $L_{u,v}$. Let K denote the number of user topics and attitudes. With the motivation of modeling user rating patterns and review topics simultaneously, for each user u, we employ the same multinomial distribution $\boldsymbol{\theta}_u$ to sample an attitude x for each observed rating $r_{u,v}$, and a set of topics $\{z_1, z_2, \cdots, z_{L_{u,v}}\}$ for each word w in associated review text $d_{u,v}$, where $L_{u,v}$ is the number of words in review text $d_{u,v}$. The corresponding graphical model of URRP is shown in Fig. 3 and is characterized by the following generative process:

(1) For each user topic $z \in [z_1, z_2, \cdots, z_K]$,
(a) Sample topic distribution $\boldsymbol{\phi}_z \sim \text{Dirichlet}(\boldsymbol{\beta})$.
(2) For each user attitude $x \in [x_1, x_2, \cdots, x_K]$,
(a) For each item $v \in \boldsymbol{V}$,
(i) Sample rating probabilities $\boldsymbol{\xi}_{x,v} \sim \text{Dirichlet}(\boldsymbol{\lambda})$.
(3) For each user $u \in \boldsymbol{U}$,
(a) Sample user review topic and rating attitude distribution $\boldsymbol{\theta}_u \sim \text{Dirichlet}(\boldsymbol{\alpha})$.
(b) For each item $v \in \boldsymbol{V}(u)$,
(i) For each word w in $d_{u,v}$,
(A) Sample topic assignment $z \sim \text{Multinomial}(\boldsymbol{\theta}_u)$.
(B) Sample word $w \sim \text{Multinomial}(\boldsymbol{\phi}_z)$.
(ii) Sample a user attitude $x \sim \text{Multinomial}(\boldsymbol{\theta}_u)$.
(iii) Sample rating value $r_{u,v}$ for chosen item v according to $r \sim \text{Multinomial}(\boldsymbol{\xi}_{x,v})$.

Obviously, there are three latent variables $\boldsymbol{\theta}$, $\boldsymbol{\phi}$, and $\boldsymbol{\xi}$, which are sampled from Dirichlet priors with parameters $\boldsymbol{\alpha}, \boldsymbol{\beta}$, and $\boldsymbol{\lambda}$ respectively. $\boldsymbol{\theta}$ encodes each user's review topics, as well as his rating attitudes; $\boldsymbol{\phi}$ represents review topics' distribution over words in vocabulary; and $\boldsymbol{\xi}$ interprets the probabilities over rating values with given attitudes and items. In the following, we provide a Gibbs sampling procedure to estimate these latent variables.

5 Parameter Estimation

In this section, we design a collapsed Gibbs sampler to learn all parameters in URRP.

From the generative process of URRP, and given hyperparameters $\Theta = \{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\lambda}\}$, the joint distribution of observed evidence $\boldsymbol{D}$ and $\boldsymbol{R}$, the user topic and

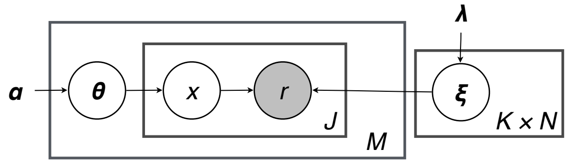

Fig. 2 Graphical model of URP.

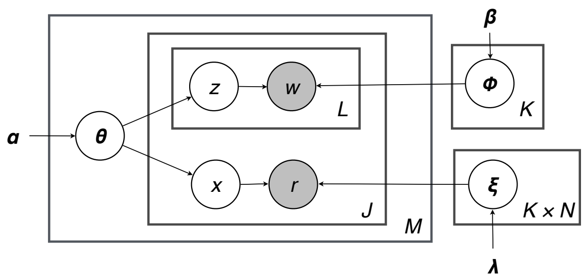

Fig. 3 Graphical model of URRP.

attitude distribution $\boldsymbol{\theta}$, topic assignments $\boldsymbol{Z}$, attitude assignments $\boldsymbol{X}$, rating probabilities $\boldsymbol{\xi}$, and word distributions $\boldsymbol{\phi}$, can computed as

$$P(\boldsymbol{D},\boldsymbol{R},\boldsymbol{\theta},\boldsymbol{Z},\boldsymbol{X},\boldsymbol{\xi},\boldsymbol{\phi}|\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\lambda}) = P(\boldsymbol{\theta}|\boldsymbol{\alpha})P(\boldsymbol{\phi}|\boldsymbol{\beta})P(\boldsymbol{\xi}|\boldsymbol{\lambda})\cdot P(\boldsymbol{Z}|\boldsymbol{\theta})P(\boldsymbol{D}|\boldsymbol{Z},\boldsymbol{\phi})\cdot P(\boldsymbol{X}|\boldsymbol{\theta})P(\boldsymbol{R}|\boldsymbol{X},\boldsymbol{\xi}) \quad (1)$$

Then integrating over $\boldsymbol{\theta},\boldsymbol{\xi}$, and $\boldsymbol{\phi}$, the likelihood of $\boldsymbol{R},\boldsymbol{D},\boldsymbol{Z}$, and $\boldsymbol{X}$ can be derived as

$$P(\boldsymbol{R},\boldsymbol{D},\boldsymbol{Z},\boldsymbol{X}|\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\lambda}) = \int P(\boldsymbol{X}|\boldsymbol{\theta})P(\boldsymbol{Z}|\boldsymbol{\theta})P(\boldsymbol{\theta}|\boldsymbol{\alpha})\mathrm{d}\boldsymbol{\theta}\cdot \int P(\boldsymbol{D}|\boldsymbol{Z},\boldsymbol{\phi})P(\boldsymbol{\phi}|\boldsymbol{\beta})\mathrm{d}\boldsymbol{\phi}\cdot\int P(\boldsymbol{R}|\boldsymbol{X},\boldsymbol{\xi})P(\boldsymbol{\xi}|\boldsymbol{\lambda})\mathrm{d}\boldsymbol{\xi} \quad (2)$$

Now we derive each integration on the right side of Eq. (2). First, since $\boldsymbol{\theta},\boldsymbol{\phi}$, and $\boldsymbol{\xi}$ are sampled from the Dirichlet distribution with parameters $\boldsymbol{\alpha},\boldsymbol{\beta}$, and $\boldsymbol{\lambda}$, respectively, we can derive $P(\boldsymbol{\theta}|\boldsymbol{\alpha})$, $P(\boldsymbol{\phi}|\boldsymbol{\beta})$, and $P(\boldsymbol{\xi}|\boldsymbol{\lambda})$ in the same way:

$$P(\boldsymbol{\theta}|\boldsymbol{\alpha}) = \prod_{u\in\boldsymbol{U}}\frac{1}{\Delta(\boldsymbol{\alpha})}\prod_{k=1}^{K}\theta_{u,k}^{\alpha_k-1},$$

$$P(\boldsymbol{\phi}|\boldsymbol{\beta}) = \prod_{k=1}^{K}\frac{1}{\Delta(\boldsymbol{\beta})}\prod_{w=1}^{W}\phi_{k,w}^{\beta_w-1},$$

$$P(\boldsymbol{\xi}|\boldsymbol{\lambda}) = \prod_{k=1}^{K}\prod_{v\in\boldsymbol{V}}\frac{1}{\Delta(\boldsymbol{\lambda})}\prod_{s=1}^{S}\xi_{k,v,s}^{\lambda_s-1},$$

where $\Delta(\boldsymbol{\alpha}) = \dfrac{\prod_{k=1}^{K}\Gamma(\alpha_k)}{\Gamma\left(\sum_{k=1}^{K}\alpha_k\right)}$, $\Delta(\boldsymbol{\beta}) = \dfrac{\prod_{w=1}^{W}\Gamma(\beta_w)}{\Gamma\left(\sum_{w=1}^{W}\beta_w\right)}$, and $\Delta(\boldsymbol{\lambda}) = \dfrac{\prod_{s=1}^{S}\Gamma(\lambda_s)}{\Gamma\left(\sum_{s=1}^{S}\lambda_s\right)}$.

Then $P(\boldsymbol{Z}|\boldsymbol{\theta})$ represents the probabilities of observing the topic assignments given multinomial distribution, which can be computed as

$$P(\boldsymbol{Z}|\boldsymbol{\theta}) = \prod_{u\in\boldsymbol{U}}\prod_{v\in\boldsymbol{V}}\prod_{w=1}^{W}p(z_{u,v,w}|\boldsymbol{\theta}_u) = \prod_{u\in\boldsymbol{U}}\prod_{k=1}^{K}\theta_{u,k}^{n_u^k},$$

where n_u^k denotes the number of times that topic k has been assigned to observed words in reviews which user u wrote and $\boldsymbol{\theta}_{u,k} = p(z_k|\boldsymbol{\theta}_u)$.

Analogously, we compute $P(\boldsymbol{X}|\boldsymbol{\theta})$ as

$$P(\boldsymbol{X}|\boldsymbol{\theta}) = \prod_{u\in\boldsymbol{U}}\prod_{v\in\boldsymbol{V}}p(x_{u,v}|\boldsymbol{\theta}_u) = \prod_{u\in\boldsymbol{U}}\prod_{k=1}^{K}\theta_{u,k}^{m_u^k},$$

where m_u^k denotes the number of times that attitude k has been assigned to observed ratings corresponding to user u.

Given attitude assignments $\boldsymbol{X}$ and the distribution over rating values $\boldsymbol{\xi}$, the likelihood of rating matrix $\boldsymbol{R}$ can be computed as

$$P(\boldsymbol{R}|\boldsymbol{X},\boldsymbol{\xi}) = \prod_{u\in\boldsymbol{U}}\prod_{v\in\boldsymbol{V}}\prod_{s=1}^{S}\xi_{x_{u,v},v,s} = \prod_{k=1}^{K}\prod_{v\in\boldsymbol{V}}\prod_{s=1}^{S}\xi_{k,v,s}^{c_{k,v}^s},$$

where $c_{k,v}^s$ denotes the number of times that rating s has been assigned to item v when attitude is k and $\boldsymbol{\xi}_{k,v,s} = P(s|x_k,v)$.

Similarly, we can derive $P(\boldsymbol{D}|\boldsymbol{Z},\boldsymbol{\phi})$ as

$$P(\boldsymbol{D}|\boldsymbol{Z},\boldsymbol{\phi}) = \prod_{u\in\boldsymbol{U}}\prod_{v\in\boldsymbol{V}}\prod_{w=1}^{W}\phi_{z_{u,v,w},w} = \prod_{k=1}^{K}\prod_{w=1}^{W}\phi_{k,w}^{n_{k,w}},$$

where $n_{k,w}$ denotes the number of times that topic k has been assigned to word w and $\boldsymbol{\phi}_{k,w} = P(w|z_k)$.

Rearranging corresponding components in Eq. (2), we obtain the full likelihood as below:

$$P(\boldsymbol{R},\boldsymbol{D},\boldsymbol{Z},\boldsymbol{X}|\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\lambda}) = \prod_{u\in\boldsymbol{U}}\frac{1}{\Delta(\boldsymbol{\alpha})}\int\prod_{k=1}^{K}\theta_{u,k}^{m_u^k+n_u^k+\alpha_k-1}\mathrm{d}\boldsymbol{\theta}_u\cdot \prod_{k=1}^{K}\frac{1}{\Delta(\boldsymbol{\beta})}\int\prod_{w=1}^{W}\phi_{k,w}^{n_{k,w}+\beta_w-1}\mathrm{d}\boldsymbol{\phi}_k\cdot \prod_{k=1}^{K}\prod_{v\in\boldsymbol{V}}\frac{1}{\Delta(\boldsymbol{\lambda})}\int\prod_{s=1}^{S}\xi_{k,v,s}^{c_{k,v}^s+\lambda_s-1}\mathrm{d}\boldsymbol{\xi}_{k,v} \quad (3)$$

In order to develop a Gibbs sampler for URRP, we need to specify the conditional probability of the hidden variables $\boldsymbol{Z}$ and $\boldsymbol{X}$, i.e., $P(\boldsymbol{Z},\boldsymbol{X}|\boldsymbol{D},\boldsymbol{R})$, which is intractable to compute. According to the collapsed Gibbs Sampling procedure, we can use a full conditional to simulate it. In our case, we define the following two conditional probabilities:

$$P(\boldsymbol{Z}_i = k|\boldsymbol{Z}_{\neg i},\boldsymbol{D},\boldsymbol{X},\boldsymbol{R}),\quad P(\boldsymbol{X}_j = k|\boldsymbol{X}_{\neg j},\boldsymbol{R},\boldsymbol{Z},\boldsymbol{D}) \quad (4)$$

where i denotes a single word observation $\{u,v,w,d_{u,v}^w\}$, $\boldsymbol{Z}_i$ is the cell of matrix $\boldsymbol{Z}$ that corresponds to this observation, while $\boldsymbol{Z}_{\neg i}$ denotes the remaining topic assignments, and j denotes a single rating $\{u,v,r_{u,v}\}$, $\boldsymbol{X}_j$ is the cell of matrix $\boldsymbol{X}$ that corresponds to this rating, and $\boldsymbol{X}_{\neg j}$ represents the remaining attitude assignments.

Thus according to the Bayes theorem and Eq. (3), we derive the two conditional probabilities in Eq. (4) as

$$P(\boldsymbol{Z}_i = k|\boldsymbol{Z}_{\neg i},\boldsymbol{D},\boldsymbol{X},\boldsymbol{R}) \propto \frac{P(\boldsymbol{Z},\boldsymbol{D},\boldsymbol{X},\boldsymbol{R})}{P(\boldsymbol{Z}_{\neg i},\boldsymbol{D}_{\neg i},\boldsymbol{X},\boldsymbol{R})} =$$

$$\frac{n_{u,\neg i}^{k}+m_{u}^{k}+\boldsymbol{\alpha}_{k}}{\sum_{k'=1}^{K}(n_{u}^{k'}+m_{u}^{k'}+\boldsymbol{\alpha}_{k'})-1}\cdot \frac{n_{k,w,\neg i}+\boldsymbol{\beta}_{w}}{\sum_{w'=1}^{W}(n_{k,w',\neg i}+\boldsymbol{\beta}_{w})-1} \tag{5}$$

$$P(X_j=k|\boldsymbol{X}_{\neg j},\boldsymbol{R},\boldsymbol{Z},\boldsymbol{D})\propto\frac{P(\boldsymbol{X},\boldsymbol{R},\boldsymbol{Z},\boldsymbol{D})}{P(\boldsymbol{X}_{\neg j},\boldsymbol{R}_{\neg j},\boldsymbol{Z},\boldsymbol{D})}= \frac{n_{u}^{k}+m_{u,\neg j}^{k}+\boldsymbol{\alpha}_{k}}{\sum_{k'=1}^{K}(n_{u}^{k'}+m_{u}^{k'}+\boldsymbol{\alpha}_{k'})-1}\cdot \frac{c_{k,v,\neg j}^{s}+\boldsymbol{\lambda}_{s}}{\sum_{s'=1}^{S}(c_{k,v,\neg j}^{s'}+\boldsymbol{\lambda}_{s})-1} \tag{6}$$

Then we can use these two equations to sample $\boldsymbol{Z}$ and $\boldsymbol{X}$, and after that we readout parameters of URRP as below:

$$\boldsymbol{\theta}_{u,k}=\frac{n_{u}^{k}+m_{u}^{k}+\boldsymbol{\alpha}_{k}}{\sum_{k'=1}^{K}(n_{u}^{k'}+m_{u}^{k'}+\boldsymbol{\alpha}_{k'})} \tag{7}$$

$$\boldsymbol{\phi}_{k,w}=\frac{n_{k,w}+\boldsymbol{\beta}_{w}}{\sum_{w'=1}^{W}(n_{k,w'}+\boldsymbol{\beta}_{w'})} \tag{8}$$

$$\boldsymbol{\xi}_{k,v,s}=\frac{c_{k,v}^{s}+\boldsymbol{\lambda}_{s}}{\sum_{s'=1}^{S}(c_{k,v}^{s'}+\boldsymbol{\lambda}_{s'})} \tag{9}$$

Now we can define the Gibbs Sampling algorithm for URRP, which is shown in Algorithm 1.

We focus on the recommender task that predicts the rating scores for items that users have not rated yet. Given a user-item pair $\langle u,v\rangle$, according to Eqs. (7) and (9), we obtain the corresponding rating distribution as

$$P(r_{u,v}{=}s)=\sum_{k=1}^{K}P(x_k|u)P(s|x_k,v)=\sum_{k=1}^{K}\boldsymbol{\theta}_{u,k}\cdot\boldsymbol{\xi}_{k,v,s} \tag{10}$$

Thus, we compute the expected rating score as the final prediction:

$$r_{u,v}=\sum_{s=1}^{S}P(r_{u,v})\cdot s=\sum_{s=1}^{S}\sum_{k=1}^{K}\boldsymbol{\theta}_{u,k}\cdot\boldsymbol{\xi}_{k,v,s}\cdot s \tag{11}$$

Algorithm 1 Gibbs Sampling for URRP

```
Input: M users, N items, all ratings R and reviews D, the
       number of latent topics K, initial hyperparameters
       α, β, λ.
Output: θ, φ, ξ

Random initialize attitudes and topics for each rating and
word;
iteration ← 0; converged ← false;
while iteration < MaxIterations && !converged do
    for each (u, v, r) ∈ R do
        x_{u,v} ← current attitude for r_{u,v};
        Exclude (u, v, r) and its attitude from current
        attitude related counts;
        x'_{u,v} ← sample new attitude according to Eq. (6);
        Set new attitude with value x'_{u,v};
        Update attitude related counts;
        for each word w ∈ accompanied review d_{u,v} do
            z_w ← current topic for word w;
            Exclude w and its topic from current topic
            related counts;
            z'_w ← sample new topic according to Eq. (5);
            Set new topic with value z'_w;
            Update topic related counts;
        end
    end
    Update hyperparameters α, β, λ;
    if iteration > burnin && iteration%sampleLag == 0
    then
        Readout parameters θ, φ, ξ according to Eq. (7),
        (8), (9) respectively;
        Calculate current MSE;
        if Current MSE > Previous MSE then
            converged ← true;
        end
    end
    iteration ← iteration + 1;
end
return θ, φ, ξ;
```

6 Empirical Study

In this section, we evaluate the effectiveness of our URRP model by applying it to 25 Amazon datasets, and compare its performance with three state-of-the-art methods.

6.1 Datasets

The Amazon datasets are collected from Ref. [13], and each of them contains mass ratings and reviews associated with a category of items. Due to the limitation of our hardware, we sample up to 50 000 users and up to 5000 items for very large datasets. For future work, we can study some optimization algorithms[17], large-scale parallel machines, and predict its performance by approach proposed in Ref. [18]. The statistics of the datasets after sampling are shown in Table 2. We can see all datasets are extremely sparse, with a 99% average sparsity. There are 1.3 million ratings (reviews) in total, and on average, each item has 24.56 ratings (reviews). The number of words in each review is 28.10.

6.2 Baselines and evaluation

We compare our approach with HFT(item)[13], the state-of-the-art method that introduces review information into recommender systems. We also compare URRP with SVD++[1], the state-of-the-art method that takes only ratings into consideration. We also compare URRP with URP[7], one of most popular probabilistic methods.

Mean Squared Error (MSE) is the most common metric used to evaluate recommender methods' performance in terms of the rating prediction task. Therefore we evaluate the performance of all comparative approaches based on MSE.

For URP and URRP, we set the hyperparameters $\boldsymbol{\alpha} = \boldsymbol{\beta} = \boldsymbol{\lambda} = 0.1$. For SVD++ and HFT (item), the bias parameters are set to 0.001. The others are set to optimal values recommended in the literature; the number of latent factors (K) is set to 5.

We randomly divide each dataset into training, validation, and test sets. Specifically, we use 80% of each dataset for training, and the remaining data is evenly split into validation and test sets. Note that we employ review texts in both the training process and the test process.

6.3 Experimental results

The experimental results from 25 datasets are shown in Table 3. Not surprisingly, URRP and HFT(item) perform much better than URP and SVD++, due to exploiting the rich information in the reviews. And URP performs slightly better than SVD++ on 22 out of the 25 datasets, since probabilistic methods are more robust than traditional matrix factorization — especially on sparse datasets. Compared with HFT(item), which is a strong baseline method that also exploits reviews, URRP still performs better on 20 out of 25 datasets.

Moreover, we conduct t-tests for the performance

Table 2 Statistics of the datasets.

Dataset	M	N	S	W	S/N	W/S
Amazon Instant Video	50 000	2812	56 357	1 236 305	20.04	21.94
Apps for Android	50 000	365	53 342	737 423	146.14	13.82
Arts	24 071	4211	27 980	677 230	6.64	24.20
Automotive	48 085	5000	52 496	1 137 671	10.50	21.67
Baby	50 000	1026	59 051	1 824 757	57.55	30.90
Beauty	50 000	2702	53 565	1 170 469	19.82	21.85
Books	50 000	3363	55 526	2 059 300	16.51	37.09
CDs & Vinyl	49 738	5000	59 023	2 066 507	11.80	35.01
Cell Phones & Accessories	50 000	4816	53 192	1 568 823	11.04	29.49
Clothing Shoes & Jewelry	50 000	3049	51 655	996 992	16.94	19.30
Digital Music	50 000	2066	76 408	3 454 727	36.98	45.21
Electronics	50 000	3880	54 307	1 792 568	14.00	33.01
Grocery & Gourmet Food	34 977	5000	38 291	790 679	7.66	20.65
Health & Personal Care	50 000	2071	52 572	1 146 945	25.38	21.82
Home & Kitchen	50 000	1589	54 214	1 400 038	34.12	25.82
Kindle Store	21 450	5000	24 647	626 018	4.93	25.40
Movies & TV	50 000	1512	62 015	2 424 778	41.02	39.10
Musical Instruments	20 396	5000	23 357	911 114	4.67	39.01
Office Products	50 000	4401	54 012	1 392 064	12.27	25.77
Patio Lawn & Garden	50 000	1623	53 241	1 463 622	32.80	27.49
Pet Supplies	50 000	2172	55 954	1 347 845	25.76	24.09
Sports & Outdoors	50 000	3459	51 852	1 174 824	14.99	22.66
Tools & Home Improvement	50 000	4307	59 507	1 601 561	13.82	26.91
Toys & Games	50 000	3526	55 412	1 504 493	15.72	27.15
Video Games	44 003	5000	64 594	2 777 999	12.92	43.01
Total	1 142 720	82 950	1 302 570	37 284 752	24.56	28.10

Table 3 MSE results of all comparative methods for $K = 5$ (standard error is shown in parentheses, and boldfaces are the best).

Dataset	Comparative methods				URRP's improvement (%)		
	URP	SVD++	HFT(item)	URRP	vs. URP	vs. SVD++	vs. HFT(item)
Amazon Instant Video	0.9924 (0.03)	1.0288 (0.05)	**0.9752 (0.03)**	0.9875 (0.03)	0.49	4.02	−1.26
Apps for Android	1.4806 (0.04)	1.5638 (0.04)	1.4908 (0.03)	**1.4641 (0.03)**	1.11	6.38	1.79
Arts	1.5185 (0.05)	1.4657 (0.03)	**1.4189 (0.05)**	1.4232 (0.05)	6.27	2.90	−0.30
Automotive	1.4652 (0.03)	1.4825 (0.05)	1.4416 (0.04)	**1.4252 (0.03)**	2.73	3.87	1.14
Baby	1.6315 (0.03)	1.7941 (0.01)	1.6158 (0.03)	**1.6038 (0.03)**	1.70	10.61	0.74
Beauty	1.5874 (0.03)	1.6297 (0.01)	1.5751 (0.03)	**1.5571 (0.03)**	1.91	4.45	1.14
Books	1.2535 (0.02)	1.2822 (0.02)	1.2488 (0.03)	**1.2204 (0.03)**	2.64	4.82	2.27
CDs & Vinyl	1.0448 (0.05)	1.0386 (0.02)	1.0343 (0.03)	**1.0129 (0.03)**	3.05	2.47	2.07
Cell Phones & Accessories	2.2254 (0.03)	2.2736 (0.03)	2.2174 (0.03)	**2.0191 (0.03)**	9.27	11.20	8.94
Clothing Shoes & Jewelry	1.4230 (0.04)	1.4484 (0.05)	**1.4098 (0.04)**	1.4126 (0.03)	0.73	2.47	−0.20
Digital Music	0.6987 (0.01)	0.6708 (0.03)	0.6688 (0.02)	**0.6480 (0.02)**	7.26	3.40	3.11
Electronics	1.7036 (0.04)	1.7353 (0.03)	1.6938 (0.04)	**1.6771 (0.03)**	1.55	3.35	0.99
Grocery & Gourmet Food	1.4828 (0.02)	1.4839 (0.04)	1.4761 (0.04)	**1.4571 (0.04)**	1.73	1.81	1.29
Health & Personal Care	1.6797 (0.01)	1.7293 (0.03)	1.7084 (0.04)	**1.6578 (0.04)**	1.30	4.14	2.96
Home & Kitchen	1.4164 (0.04)	1.4627 (0.03)	1.4035 (0.03)	**1.3885 (0.03)**	1.97	5.07	1.07
Kindle Store	1.6544 (0.05)	1.7031 (0.04)	1.6440 (0.05)	**1.6436 (0.04)**	0.65	3.50	0.02
Movies & TV	1.1904 (0.05)	1.1929 (0.02)	**1.1780 (0.03)**	1.1787 (0.03)	0.98	1.19	−0.06
Musical Instruments	1.2595 (0.05)	1.3071 (0.05)	1.2437 (0.05)	**1.2271 (0.05)**	2.58	6.12	1.34
Office Products	1.7640 (0.01)	1.7907 (0.01)	**1.7417 (0.03)**	1.7470 (0.03)	0.96	2.44	−0.30
Patio Lawn & Garden	1.6213 (0.02)	1.6957 (0.05)	1.6164 (0.04)	**1.5946 (0.03)**	1.65	5.96	1.35
Pet Supplies	1.5588 (0.03)	1.6599 (0.03)	1.5534 (0.03)	**1.5200 (0.03)**	2.49	8.43	2.15
Sports & Outdoors	1.3831 (0.04)	1.4207 (0.04)	1.3728 (0.03)	**1.3538 (0.03)**	2.12	4.71	1.38
Tools & Home Improvement	1.4074 (0.01)	1.4229 (0.04)	1.3903 (0.03)	**1.3692 (0.03)**	2.72	3.77	1.52
Toys & Games	1.1095 (0.03)	1.1420 (0.05)	1.1378 (0.03)	**1.1066 (0.03)**	0.26	3.10	2.74
Video Games	1.4139 (0.03)	1.4392 (0.03)	1.4006 (0.03)	**1.3747 (0.03)**	2.77	4.48	1.85

of URRP compared with the baselines; the results are shown in Table 4. We see that our URRP model shows average improvements in terms of MSE of 2.49%, 4.87%, and 1.65% when compared with URP, SVD++, and HFT (item), respectively. Furthermore, the performance improvement of our method is statistically significant, at the 1% level (i.e., p-value < 0.01).

6.4 Interpretable topics

In addition to better performance, there is another promising ability of the URRP model: It can learn interpretable user review topics. In Tables 5 and 6, we show the top five words of five topics learned by URRP on the Apps for Android and Home & Kitchen datasets. More specifically, the five topics learned on Apps for Android are radio, game, read, health, and music, and for Home & Kitchen they are snack,

Table 4 Average MSE results of comparative methods and t-test p-values of URRP vs. baselines (Best result in boldface).

Method	Average MSE	Improvement (%) (p-value)
URP	1.4386	2.49 (0.0001)
SVD++	1.4745	4.87 (0.0000)
HFT(item)	1.4263	1.65 (0.0030)
URRP	**1.4028**	

Table 5 Top five words for topics in Apps for Android dataset.

Topics	Radio	Game	Read	Health	Music
Top 1	radio	game	kindle	love	amazon
Top 2	listen	play	phone	day	music
Top 3	time	fun	book	track	version
Top 4	word	enjoy	read	calories	player
Top 5	station	free	tablet	weight	stars

Table 6 Top five words for topics in Home & Kitchen dataset.

Topics	Snack	Kitchenware	Bread	Beverage	Rice
Top 1	ice	great	pan	coffee	cooker
Top 2	cream	knife	great	hot	pressure
Top 3	make	easy	product	water	rice
Top 4	popcorn	works	bread	tea	time
Top 5	easy	kitchen	made	cup	cups

kitchenware, bread, beverage, and rice.

Through these topics, we can express ratings by analyzing their accompanying reviews' distribution on these topics. This will help us understand users' rating behavior, why they give a high or low rating to an item, what's the most popular property of an item, and why. In a word, these interpretable topics can help us understand users' preferences and items' properties better, allowing us to make better recommendations to users.

7 Conclusions

We have presented URRP, a Bayesian model that exploits user preference information embedded in ratings and reviews in order to make accurate recommendations to customers. URRP links traditional collaborative filtering with topic modeling seamlessly, by applying the same multinomial distribution to each user's latent rating factors and review topics, since his rating behaviors and review topics are essentially the same. By introducing the wealth of information in reviews, URRP can learn users' rating behaviors more accurately, even with few ratings. For the purpose of evaluating URRP's effectiveness with other state-of-the-art recommender methods, we conducted experiments on 25 large and real-world datasets, which consist of over one million ratings and reviews. The experimental results demonstrate that URRP can make more accurate rating predictions than those of other state-of-the-art approaches, and express these ratings by review topics.

Acknowledgements

This work was supported by the National Key Basic Research and Development (973) Program of China (No. 2013CB329600), the National Natural Science Foundation of China (Nos. 61472040 and 60873237), and Beijing Higher Education Young Elite Teacher Project (No. YETP1198).

References

[1] Y. Koren, Factorization meets the neighborhood: A multifaceted collaborative filtering model, in *Proc. 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Las Vegas, NV, USA, 2008, pp. 426–434.

[2] Y. Koren, R. Bell, and C. Volinsky, Matrix factorization techniques for recommender systems, *Computer*, no. 8, pp. 30–37, 2009.

[3] M. Benjamin, Modeling user rating profiles for collaborative filtering, in *Proc. 18th Advances in Neural Information Processing Systems*, Vancouver, Canada, 2004, pp. 627–634.

[4] A. Mnih and R. Salakhutdinov, Probabilistic matrix factorization, in *Proc. 21th Advances in Neural Information Processing Systems*, Vancouver, Canada, 2007, pp. 1257–1264.

[5] R. Salakhutdinov and A. Mnih, Bayesian probabilistic matrix factorization using Markov chain Monte Carlo, in *Proc. 25th ACM International Conference on Machine Learning*, Helsinki, Finland, 2008, pp. 880–887.

[6] H. Shan and A. Banerjee, Residual Bayesian co-clustering for matrix approximation, in *Proc. SIAM International Conference on Data Mining*, Columbus, OI, USA, 2010, pp. 223–234.

[7] N. Barbieri, Regularized Gibbs sampling for user profiling with soft constraints, in *Proc. International Conference on Advances in Social Networks Analysis and Mining*, Kaohsiung, Taiwan, China, 2011, pp. 129–136.

[8] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, Methods and metrics for cold-start recommendations, in *Proc. 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Tampere, Finland, 2002, pp. 253–260.

[9] C. Wang and D. M. Blei, Collaborative topic modeling for recommending scientific articles, in *Proc. 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Diego, USA, 2011, pp. 448–456.

[10] D. Agarwal and B. C. Chen, fLDA: Matrix factorization through latent dirichlet allocation, in *Proc. 3th ACM International Conference on Web Search and Data Mining*, New York, USA, 2010, pp. 91–100.

[11] D. M. Blei, A. Y. Ng, and M. I. Jordan, Latent Dirichlet allocation, *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[12] A. Paterek, Improving regularized singular value decomposition for collaborative filtering, in *Proc. KDD Cup and Workshop*, San Jose, CA, USA, 2007, pp. 5–8.

[13] J. McAuley and J. Leskovec, Hidden factors and hidden topics: Understanding rating dimensions with review text, in *Proc. 7th ACM Conference on Recommender Systems*, Hong Kong, China, 2013, pp. 165–172.

[14] Y. Bao, H. Fang, and J. Zhang, Topicmf: Simultaneously exploiting ratings and reviews for recommendation, in *Proc. 28th AAAI Conference on Artificial Intelligence*, Quebec, Canada, 2014, pp. 2–8.

[15] G. Ling, M. R. Lyu, and I. King, Ratings meet reviews, a combined approach to recommend, in *Proc. 8th ACM Conference on Recommender Systems*, Foster, Silicon Valley, CA, USA, 2014, pp. 105–112.

[16] H. Li, F. Zhang, and S. Zhang, Multi-feature hierarchical topic models for human behavior recognition, *Science China Information Sciences*, vol. 57, no. 9, p. 092107, 2014.

[17] M. Kaedi, N. Ghasem-Aghaee, and C. W. Ahn, Holographic memory-based Bayesian optimization algorithm (HM-BOA) in dynamic environments, *Science China Information Sciences*, vol. 56, no. 9, p. 092111, 2013.

[18] J. Zhai, W. Chen, and W. Zheng, Phantom: Predicting performance of parallel applications on large-scale parallel machines using a single node, *ACM Sigplan Notices*, vol. 45, no. 5, pp. 305–314, 2010.

Mingming Jiang received his BEng degree from Beijing Institute of Technology, China, in 2013. He was a research assistant at Singapore Management University from February to July in 2015. He is currently a master student in the School of Computer Science and Technology, Beijing Institute of Technology, China. His research interests include recommender system, topic model, and social network.

Lejian Liao received the PhD degree from the Institute of Computing Technology, Chinese Academy of Sciences in 1994. He is currently a professor in the School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China. He also served as the Vice Dean of the school. With main research interest in machine learning, natural language processing, and intelligent network, Professor Lejian Liao has published numerous papers in several areas of computer science.

Dandan Song received her BEng degree and PhD degree from Tsinghua University, China, in 2004 and 2009, respectively. She is currently an associate professor in the School of Computer Science and Technology, Beijing Institute of Technology. Her research interests include information retrieval, data mining, and bioinformatics.

Feida Zhu received the BSc degree from Fudan University, China in 2001, and the PhD degree in computer science from the University of Illinois at Urbana-Champaign in 2009. He is an assistant professor in the School of Information Systems at Singapore Management University. His research interests include large-scale data mining, graph/network mining, and social network analysis.