Probabilistic Local Matrix Factorization based on User Reviews

Xu Chen[1], Yongfeng Zhang[2], Wayne Xin Zhao[3⋆], Wenwen Ye[1] and Zheng Qin[1]

[1] School of Software, Tsinghua University
[2] College of Information and Computer Sciences, University of Massachusetts Amherst
[3] School of Information, Renmin University of China

Abstract. Local matrix factorization (LMF) methods have been shown to yield competitive performance in rating prediction. The main idea is to leverage the ensemble of submatrices for better low-rank approximation. However, the generated submatrices and recommendation results in the existing methods are usually hard to interpret. To address this issue, we adopt a probabilistic approach to enhance model interpretability of LMF methods by leveraging user reviews. In specific, we incorporate item-topics to construct meaningful "local clusters", and further associate them with opinionated word-topics to explain the corresponding semantics and sentiments of users' ratings. The proposed approach is a joint model which characterizes both ratings and review text. Extensive experiments on real-world datasets demonstrate the effectiveness of our proposed model compared with several state-of-art methods. More importantly, the produced results provide meaningful explanations to understand users' ratings and sentiments.

1 Introduction

Recently, local matrix factorization (LMF) has attracted increasing attention [10, 4, 19] in recommender system community. LMF methods have been shown to give better performance than traditional matrix factorization (MF) techniques [9, 16] in rating prediction. Typically, LMF methods identify subgroups of users and items, and construct multiple submatrices based on the original user-item rating matrix. They apply traditional MF methods to each submatrix individually, and combine the results from submatrices as the final prediction. Such an approach aims to enhance the low-rank property of submatrices and improve parallel processing.

So far, existing LMF methods mainly focus on how to "look for" subgroups using the proximity criterion, including random dividing [13], kernel smoothing [10] and Bregman co-clustering [6]. An important aspect has been usually ignored, *i.e.,* model interpretability. Due to this, several queries cannot be well answered by previous studies, including why such subgroups are formed, what are the semantics of each subgroup, how a user in a subgroup rates and comments. In addition to performance, these problems are fundamental to understand users' rating behaviors and explain recommendation results. It is important to consider these factors in order to design an effective recommender.

To address this issue, we propose a novel explainable probabilistic LMF model by leveraging user reviews. The key to explainable LMF is how to derive explainable

⋆ Corresponding author

subgroups. Inspired by [19], we adopt topic modeling to characterize items clusters as *item-topics*, and further assign users to these item-topics "softly" (*i.e.,* in a probabilistic way). An item-topic is essentially a multinomial distribution over the set of items and tends to cluster items with similar functions or categories, which is relatively easy to interpret. For each item-topic, we set topic-specific latent factors for both users and items. In this way, we can better understand users' rating behaviors with the help of topical contexts. Such a formulation partially provides the semantics of subgroups, however, it is still unable to explain why a user gives a high or low rating to an item. Our solution is to further leverage review information to further enhance the interpretability of prediction results. In specific, we incorporate opinionated word-topics like that in topic models [3] to characterize the semantics and sentiments reflected in review text. In our model, an item-topic is associated with a unique distribution over word-topics in each sentiment level, and the generation of review text is based on both item-topic and sentiment level. The incorporation of opinionated word-topics can improve the learning of item-topics, since review text is likely to contain relevant aspect words. In addition, we can identify the most associated words to explain the opinion polarity of user ratings.

To evaluate the performance of the proposed model, we conduct extensive experiments on real-world datasets. The experimental results indicate the effectiveness of our model. Especially, it has been shown to give better explainability for the learned models and prediction results. The main contribution of this work is to incorporate item-topics to construct meaningful subgroups, and associate them with opinionated word-topics mined from review text to explain the corresponding semantics and ,sentiments for users ratings. By using topic-specific latent factors, our model yields competitive performance while the learned item- and word-topics give good interpretability. To our knowledge, it is the first time that word-topics discovered in the review text have been utilized to explain LMF methods.

2 Related Work

Local matrix factorization. Recently, local matrix factorization (LMF) has received much attention [10, 26, 25, 4, 13], which aim to enhance the low-rank property and parallel processing. Typically, these methods split the original matrix into smaller submatrices, and then apply traditional MF technqiues [9, 16] on submatrices individually. The final predictions are generated by combing the predictions from submatrices. In specific, the DFC model [13] randomly divided the original matrix into small subgroups; the kernel smoothing method was used to find nearest neighbors [10]; the Bregman co-clustering method was exploited to split the original matrix [4]. The most recent study [19] adopted a probabilistic approach to generating "soft" clusters, however, it cannot model review text.

Review-based recommendation. Review information has been shown to improve the performance of rating prediction [14, 8, 12, 7, 1, 23, 22, 11]. The major benefits gained from review information can be summarized in two aspects. First, the user-item rating matrix is sparse, and the auxiliary textual information is able to alleviate this issue to some extent [21]. More importantly, textual contents in user reviews can provide explainable information for users' ratings [14, 20, 24, 5]. The HFT model [14] directly transformed the latent factors in MF side into the topic distributions. By aligning latent factors with topics, the produced results help understand users' ratings. The PACO

model [20] designed a poisson additive co-clustering model to build interpretable recommendation system. The EFM model [24] extracted product features and users' corresponding sentiments on them, and further used them to explain user ratings.

Our work is closely related to these studies, and makes a meaningful connection between LMF and explainable MF methods. Although the superiority of LMF methods in rating prediction has been shown, the explainability has seldom been well addressed. Hence, the semantics of each subgroup were not clear and users' rating behaviors cannot be well understood. As a comparison, we propose to use item-topics to explain subgroups and opinionated word-topics from review text to explain the ratings and sentiments.

Table 1. Notations and descriptions.

Notations	Descriptions
N, M	number of rows (users) and columns (items)
K_1, K_2	the number of item- and word-topics respectively
D	the number of dimensions for latent vectors
u, i, c, k	index variables respectively for users and items, item- and word-topics
$\boldsymbol{p}_u^{(c)}$, $\boldsymbol{q}_i^{(c)}$	the latent vector ($\in \mathbb{R}^D$) respectively for user u and item i *w.r.t.* the item-topic c
$r_{u,i}$/$\boldsymbol{w}_{u,i}$	the rating and review text of user u on item i
$y_{u,i}$	item-topic assignment associated with $r_{u,i}$
$z_{u,i,j}$	word-topic assignment for the j-th word in $\boldsymbol{w}_{i,j}$
ψ^k	word distribution in word-topic k
θ^c	word-topic distribution of item-topic c
φ^c	item distribution in item-topic c
ϕ^u	item-topic distribution of user u
$\boldsymbol{\lambda_0}, \boldsymbol{\lambda_P}, \boldsymbol{\lambda_Q}$	priors of latent factors
$\beta/\beta^{'}$, α/α'	priors of the word/item-topics and distributions over word/item-topics respectively

3 Probabilistic Local Matrix Factorization based on User Review

In this section, we present the proposed probabilistic local matrix factorization. To make more clear presentation, we first list the notations used throughout the paper in Table 1.

3.1 The Proposed Model

As indicated in [19], lack of interpretability has been one major issue for previous LMF methods [10, 13]. These models cannot answer two typical queries well: (1) what are the semantics for the generated submatrices, and (2) why a user likes or dislikes an item. In our model, we aim to solve the above issues by considering two aspects. First, we adopt a probabilistic topic modeling approach to learn "soft" subgroup of the items, and each item-topic together with the associated ratings can be considered as a local (*i.e.,* topic-specific) view of the entire user-item rating matrix. Second, we incorporate opinionated word-topics to enhance the learning of item-topics, and describe how a user

comment on an item with some sentiment. The final model integrates both aspects (*i.e.,* ratings and review).

Modeling local subgroups with probabilistic topic models. The key step for LMF methods lies in that how to generate subgroups of users and items, which will further form a corresponding submatrix. We first adopt item-topics to model item subgroups. Formally, an item-topic c is a multinomial distribution over the set of all the items, denoted by φ^c. Each entry φ_i^c denotes the probability of item i in item-topic c. Further, we model a user u's interests by a multinomial distribution over item-topics denoted by ϕ^u. Each entry ϕ_c^u denotes the probability that a user is likely to rate an item in item-topic c. The interest distribution can be considered as users' membership over item subgroups, which forms probabilistic user subgroups. Next, we study how to generate a rating triplet $\langle u, i, r_{u,i}\rangle$. In LMF methods, each user (or item) will be associated with a unique latent factor in different submatrices. In our case, an item-topic corresponds to a local submatrix. Following [19], we propose to use topic-specific latent factors. For each item-topic c, we set a corresponding latent factor $\boldsymbol{p}_u^c$ ($\boldsymbol{q}_i^c$) for user u (item i). When a user u starts to rate an item i, she first draw a topic assignment $y_{u,i}$ according to ϕ^u, and then generates item i using φ^c like that in LDA [3]. Once the item-topic assignment has been sampled, following PMF [16, 19], the rating $r_{u,i}$ is generated by using a topic-specific Gaussian distribution

$$\mathcal{N}(r_{ui}|(\boldsymbol{p}_u^{y_{u,i}})^\top \cdot (\boldsymbol{q}_i^{y_{u,i}}), \sigma^2_{y_{u,i}}), \tag{1}$$

where $\sigma^2_{y_{u,i}}$ is the variance corresponding to topic $y_{u,i}$.

Modeling user reviews to explain the item subgroups. Above, a subgroup of items is modeled as an item-topic, which is a soft clustering of the items. These item-topics provide important topical contexts to explain rating predictions using topic-specific latent factors. However, the item-topics mainly reflect co-occurrence patterns based on users' rating history, and they cannot capture the sentiment level of a user towards an item, *i.e.,* why a user gives a high rating or a low rating. Intuitively, the opinion polarity of a user review tends to be more positive if her rating is higher, and the generation of review text is closely related to the sentiment levels of a user on an item. Let $\mathcal{O} = \{1, ..., l, ..., L\}$ be a set of L sentiment labels, in which each label l denotes a sentiment level and a higher level indicates a more positive polarity. Our key idea lies in that the generation of a review text should be based on both item-topic and sentiment level. Formally, we assume that there are a set of K_2 word-topics. Each word-topic k is modeled as a multinomial distribution over the terms in the vocabulary, denoted by ψ^k. We assume that an item-topic c will correspond to an opinionated distribution over word-topics for each sentiment level l, denoted by $\theta^{c,l}$. Each entry $\theta_k^{c,l}$ denotes the probability of word-topic k for item-topic c with sentiment label l. Let $\boldsymbol{w}_{u,i}$ denote a vector of words in the review associated with the rating record $\langle u, i, r_{u,i}\rangle$. For each word token $w_{u,i,j} \in \boldsymbol{w}_{u,i}$, we first draw a word-topic assignment $z_{u,i,j}$ according to $\theta^{z_{u,i}, l_{u,i}}$, where $z_{u,i}$ and $l_{u,i}$ correspond to the item-topic and sentiment label respectively for $\langle u, i, r_{u,i}\rangle$. Then we generate word $w_{u,i,j}$ according to the word-topic $\psi^{z_{u,i,j}}$. Our generation process involves the sentiment label for a user review. There can be several ways to set the sentiment labels. Here we adopt a simple yet effective method: we consider two sentiment levels (*i.e.,* positive and negative) and set it based on the corresponding rating score. In

a five-star rating system, the sentiment label of a user review is set to positive, if the rating score is higher than three stars, otherwise it will be set to negative.

The final model. Our final model combines the above two parts: it characterizes user ratings by probabilistic LMF and models user reviews to explain the item subgroups learned in LMF. We implement a full Bayesian formulation for this model. In specific, for each item-topic c, we put priors on topic-specific latent factors $\boldsymbol{p}_u^c$ and $\boldsymbol{q}_i^c$, denoted by $\boldsymbol{\lambda}_{\boldsymbol{P}}^{\mathbf{c}}, \boldsymbol{\lambda}_{\boldsymbol{Q}}^{\mathbf{c}}, \boldsymbol{\lambda}_{\mathbf{0}}^{\mathbf{c}}$ as in BPMF [17]; we also put priors on variables φ^c, ϕ^u, ψ^k and θ^c, denoted by $\alpha, \alpha', \beta, \beta'$ respectively. We refer to the proposed model as **ELMF** (*Explainable LMF*). The generation process and complete plate notation for our model have been shown in Fig. 1 and 2 respectively. In our model, the item-topic $y_{u,i}$ is not only used to generate the items, but associated with distributions over word-topics to generate review text. The incorporation of review text is able to improve the coherence of item subgroups (*i.e.,* topics), and also enhance the semantic explainability of user ratings. When generating review text, the sentiment label of a review also plays an important role. As indicated [18], when a user is praising or criticizing an item, the word topics she selects are likely to be different. Based on this consideration, the generation of review text is based on both item-topic and sentiment level.

1. For each item-topic $c = 1, ..., K_1$, draw a multinomial distribution over all the items $\varphi^c \sim Dir(\beta^{'})$;
2. For each word-topic $k = 1, ..., K_2$, draw a multinomial distribution over all the words $\psi^k \sim Dir(\beta)$;
3. For item-topic $c = 1,...,K_1$,
   i. Draw the hyperparameters of the user and item latent vectors $P(\boldsymbol{\lambda}_{\boldsymbol{P}}^{\mathbf{c}}|\boldsymbol{\lambda}_{\mathbf{0}}^{\mathbf{c}})$ and $P(\boldsymbol{\lambda}_{\boldsymbol{Q}}^{\mathbf{c}}|\boldsymbol{\lambda}_{\mathbf{0}}^{\mathbf{c}})$
   ii. For each item $i = 1, ..., M$, draw the topic-specific item latent vector $\boldsymbol{q}_i^c \sim P(\boldsymbol{q}_i^c|\boldsymbol{\lambda}_{\boldsymbol{Q}}^{\mathbf{c}})$;
   iii. For each user $u = 1, ..., N$, draw the topic-specific item latent vector $\boldsymbol{p}_u^c \sim P(\boldsymbol{p}_u^c|\boldsymbol{\lambda}_{\boldsymbol{P}}^{\mathbf{c}})$;
   iv. For each sentiment label $l = 1, ..., L$, draw a multinomial distribution over all the word-topics $\theta^{c,l} \sim Dir(\alpha)$;
3. For each user $u = 1, ..., N$,
   i. Draw a multinomial distribution over all the item-topics $\phi^u \sim Dir(\alpha')$;
   ii. For each rated item i by u,
      (1) Draw an item-topic $y_{u,i} \sim Disc(\phi^u)$;
      (2) Draw the item $i \sim Disc(\varphi^{y_{u,i}})$;
      (3) Draw the rating $r_{u,i} \sim \mathcal{N}(r_{u,i}|(\boldsymbol{p}_u^{y_{u,i}})^\top \cdot (\boldsymbol{q}_i^{y_{u,i}}), \sigma^2_{y_{u,i}})$;
      (4) Set the sentiment label $l_{u,i}$ based on $r_{u,i}$;
      (5) For each word token $w_{u,i,j}$ in $\boldsymbol{w}_{u,i}$,
         · Draw a word-topic $z_{u,i,j} \sim Disc(\theta^{y_{u,i},l_{u,i}})$;
         · Draw the word $w_{u,i,j} \sim Disc(\psi^{z_{u,i,j}})$.

Fig. 1. The generative process of the ELMF model.

Insights into the model interpretability of ELMF. The model interpretability of ELMF has been reflected in two aspects. First, it tries to look for more meaningful subgroups of items and users in a probabilistic way. We achieve this by capturing item co-occurrence patterns in rating history. The derived item-topics can be meaningful with similar func-

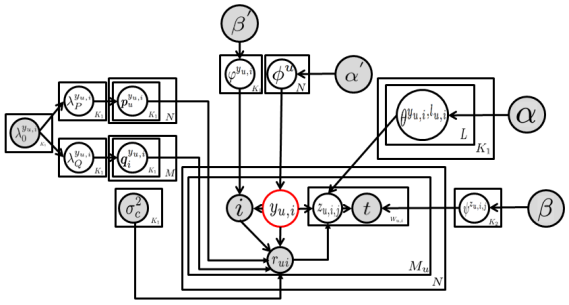

Fig. 2. The plate notation for our ELMF model.

tions or categories. For each item-topic, we set topic-specific latent factors for both users and items. In this way, we can better understand users' rating preference with the help of topical contexts. Second, an item-topic is associated with different word-topics in different sentiment labels. We incorporate textual contexts as opinionated word-topics to explain why a user likes or dislikes an item. The combination of these two aspects yields a better model explainability and meanwhile keeps the performance superiority of LMF methods.

3.2 Model Learning

We would like to learn the following parameters or variables: $\{\theta, \psi, \phi, \varphi, \boldsymbol{p}, \boldsymbol{q}\}$ by fixing the hyper-parameters. We aim to maximize the joint likelihood of observed ratings and review text. The problem is hard to directly optimize, and we adopt a collapsed Gibbs sampling method for solving it. Our learning tasks involve two major parts, inferring word- and item-topic assignments $\{\boldsymbol{y}, \boldsymbol{z}\}$ and optimizing the latent factors $\{\boldsymbol{p}, \boldsymbol{q}\}$. Once the topic assignments $\{\boldsymbol{y}, \boldsymbol{z}\}$ have been obtained, the distribution parameters $\{\theta, \psi, \phi, \varphi\}$ can be then estimated based on the word- and item-topic assignments. Let $\boldsymbol{r}, \boldsymbol{m}, \boldsymbol{w}, \boldsymbol{l}, \boldsymbol{y}$ and $\boldsymbol{z}$ be the vectors for ratings, items, words, sentiment labels, item-topic assignments and word-topic assignments respectively. Next we give the Gibbs sampling formula for topic assignments and the update formula for latent factors. For convenience, let Ψ denote all the hyper-parameters.

Sampling item-topics. Fixing all latent factors $\{\boldsymbol{p}, \boldsymbol{q}\}$, the item-topic assignment for the rating triplet $\langle u, i, r_{u,c} \rangle$ can be drawn according to:

$$
\begin{aligned}
& P(y_{u,i} = c | \boldsymbol{r}, \boldsymbol{m}, \boldsymbol{w}, \boldsymbol{l}, \boldsymbol{y}_{\neg(u,i)}, \boldsymbol{z}, \boldsymbol{p}, \boldsymbol{q}; \Psi) \\
\propto & \frac{n_c^u + \alpha'}{\sum_{c'=1}^{K_1} n_{c'}^u + \alpha'} \times \frac{n_i^c + \beta'}{\sum_{i'=1}^{M} n_{i'}^c + \beta'} \\
& \times \mathcal{N}(r_{u,i} | \boldsymbol{p}_u^c, \boldsymbol{q}_i^c, \sigma_c^2) \times \frac{\Delta(\boldsymbol{n}^{c,l_{u,i}} + \boldsymbol{n}^{u,i} + \boldsymbol{\alpha})}{\Delta(\boldsymbol{n}^{c,l_{u,i}} + \boldsymbol{\alpha})},
\end{aligned}
\tag{2}
$$

where n_c^u denotes the number of the rated items by user u assigned to item-topic c, n_i^c denotes the number that item i is assigned to item-topic c, $\boldsymbol{n}^{c,l_{u,i}}$ is a V-dimensional

(vocabulary size) count vector in which $n_w^{c,l_{u,i}}$ denotes the number of word w attached to items in item-topic c with sentiment label $l_{u,i}$, $\boldsymbol{n}^{u,i}$ is a V-dimensional count vector in which $n_w^{u,i}$ denotes the number of word w appearing the review associated with rating $r_{u,i}$, and $\boldsymbol{\alpha}$ is a V-dimensional vector of equal value α. All the count statistics are computed by excluding the information associated with $\langle u, i, r_{u,i}\rangle$. We define the $\Delta(\boldsymbol{x})$ function as $\Delta(\boldsymbol{x}) = \frac{\Pi_{w=1}^{V}\Gamma(x_w)}{\Gamma(\sum_{w'=1}^{V} x_{w'})}$.

Sampling word-topics. Given the item-topic assignment $y_{u,i}$, we sample the word-topic for the j-th word $n^k_{w_{u,i,j}}$ in the review associated with the rating $r_{u,i}$ according to:

$$\begin{aligned} & P(z_{u,i,j} = k|\boldsymbol{r}, \boldsymbol{m}, \boldsymbol{w}, \boldsymbol{y}, \boldsymbol{l}, \boldsymbol{z}_{\neg(u,i,j)}, \boldsymbol{p}, \boldsymbol{q}; \Psi) \\ \propto & \ \frac{n_k^{y_{u,i},l_{u,i}} + \alpha}{\sum_{k'=1}^{K_2} n_{k'}^{y_{u,i},l_{u,i}} + \alpha} \times \frac{n^k_{w_{u,i,j}} + \beta}{\sum_{w'} n^k_{w'} + \beta}, \end{aligned} \tag{3}$$

where $n_k^{y_{u,i},l_{u,i}}$ is the number that words assigned to word-topic k with the associated item-topic $y_{u,i}$ and sentiment label $l_{u,i}$, $n^k_{w_{u,i,j}}$ denotes the number that word $w_{u,i,j}$ is assigned to word-topic k.

Updating latent factors. Following BPMF [17], given an item-topic c, $\boldsymbol{\lambda_P^c} = \{\mu_p^c, \Lambda_P^c\}$ is first generated according to a Gaussian-Wishart distribution with parameter $\boldsymbol{\lambda_0^c} = \{\mu_0^c, \nu_0^c, \boldsymbol{W}_0^c\}$ by $P(\boldsymbol{\lambda_P^c}|\boldsymbol{\lambda_0^c}) = \mathcal{N}(\mu_p^c|\mu_0^c, (\Lambda_P^c)^{-1})\mathcal{W}(\Lambda_P^c|\nu_0^c, \boldsymbol{W}_0^c)$, then the conditional distribution over user u's latent factor $\boldsymbol{p}_u^c$ is:

$$\begin{aligned} & P(\boldsymbol{p}_u^c|\boldsymbol{r}, \boldsymbol{q}^c; \Phi) \\ = & \ \mathcal{N}(\boldsymbol{p}_u^c|\bar{\mu}_p^c, (\bar{\Lambda}_P^c)^{-1}) \\ \propto & \ P(\boldsymbol{p}_u^c|\mu_P^c, (\Lambda_P^c)^{-1}) \times \prod_{v=1}^{M} \mathcal{N}(r_{ui}|\boldsymbol{P}_u^c, \boldsymbol{q}_i^c, \sigma_c^2) \end{aligned} \tag{4}$$

where we have: $\bar{\Lambda}_P^c = \Lambda_P^c + \frac{1}{\sigma_c^2}\sum_{v=1}^{M}(\boldsymbol{q}_i^c(\boldsymbol{q}_i^c)^\top)^{\mathbb{I}(y_{u,i},c)}$ and $\bar{\mu}_P^c = (\bar{\Lambda}_P^c)^{-1}(\Lambda_P^c\mu_P^c + \frac{1}{\sigma_c^2}\sum_{v=1}^{M}(\boldsymbol{q}_i^c r_{u,i})^{\mathbb{I}(y_{u,i},c)})$, where $\mathbb{I}(y_{u,i}, c)$ is an indicator function which returns 1 only when $y_{u,i} = c$ otherwise 0. Following Eq. 5, $\boldsymbol{q}_i^c$ can be updated in a similar way, we omit it here.

Computational complexity. Let D be the dimension number of latent factors in $\boldsymbol{p}_u^c$ and $\boldsymbol{q}_v^c$, $S = \sum_{u,c,i}\mathbb{I}(y_{u,i}, c)$, K_1 be the number of item-topics, then according to [19], updating users' and items' latent factors in each iteration takes a cost of $\mathcal{O}(D^2S + D^3K_1N + D^3K_1M)$. Let A be the number of all the observed ratings, B be average number of words in a review, K_2 be the number of word-topics. The complexity for item-topic and word-topic assignment is $\mathcal{O}(A(K_1DB + BK_2))$. Hence, the overall cost in an iteration is $\mathcal{O}(D^2S + D^3K_1N + D^3K_1M + ABK_1D + ABK_2)$

Rating Prediction. Once the model parameters have been learned, we can predict the final ratings by: $\hat{r}_{u,i} = \sum_{c=1}^{K_1}\{\frac{Z_c}{Z_{(\cdot)}}(\boldsymbol{p}_u^c)^\top \cdot \boldsymbol{q}_i^c\}$, where $Z_c = \phi_c^u \times \varphi_i^c$ and $Z_{(\cdot)} = \sum_{c=1}^{K_1} Z_c$. Z_c can be understood as the weight coefficient for the predictions from the item-topic c. Z_c will be large if both u and i have close associations with item-topic c, *i.e.,* ϕ_c^u and φ_i^c are large.

4 Experiments

In this section, we conduct evaluation experiments to examine the performance of the proposed model.

4.1 Experimental Setup

Datasets. Six datasets [15] from diverse Amazon categories have been used as evaluation collection. We present the statistics of the datasets in Table 2. For each dataset, we randomly select 80% ratings to train our model, and the rest are held out for testing.

Table 2. Statistics of our datasets.

Datasets	#Users	#Items	#Ratings	$\frac{\#Reviews}{\#Users}$	Density
Music	1492	900	7931	5.55	0.59%
Auto	2928	1835	18308	6.25	0.34%
Office	4905	2420	39974	8.14	0.33%
Patio	1686	962	11740	6.96	0.72%
Video	5130	1685	33146	6.46	0.38%
DigiMu	5541	3568	48255	8.71	0.24%

Baseline methods. We compare our models with the following baselines:

- *PMF* [16]: It's a Bayesian probabilistic implementation of the traditional matrix factorization, shown to give accurate predictions in practice.
- *HFT* [14]: It's a competitive review-based rating prediction method by leveraging review information to enhance the prediction performance.
- *BPMTMF* [19]: It's a state-of-art local matrix factorization method which adopts a Bayesian formulation approach to model user ratings.

There can be more LMF baselines to compare here, including DFC [13], LLORMA [10] and WEMAREC [4]. As shown in [19], BPMTMF is better than these baselines. Our empirical results have also confirmed this. Hence, we only select BPMTMF as the only representative for LMF baselines. We use a five-fold cross validation to obtain the final performance for all the comparison methods. To set various parameters in both baselines and our models, we use a grid search method by finding the values which lead to the best performance in five-fold cross validation. For our model ELMF, we set the number of latent factors to 20, the number of item-topics K_1 to 20, the number of word-topics K_2 to 15, and the hyper-parameters are set as follows $\alpha' = 0.01, \beta' = 2.5, \alpha = 0.1, \beta = 2.5, \mu_0^c = 0, \nu_0^c = 20, \boldsymbol{W}_0^c$ is a identity matrix, $\sigma = \sqrt{2}$.

4.2 Evaluation on Rating Prediction

We first evaluate the performance of the proposed models in rating prediction, and adopt the commonly used Root Mean Square Error (RMSE) as the evaluation metrics,

defined as $\sqrt{\frac{\sum_{\langle u,i,r_{u,i}\rangle \in \mathcal{D}} (r_{u,i} - \hat{r}_{u,i})^2}{|\mathcal{D}|}}$, where $|\mathcal{D}|$ is the number of samples in the test set $\mathcal{D}$.

Results and Analysis. We present the comparison results in Table 3 and have the following observations. First, HFT gives better performance than PMF in four of six datasets. HFT leverages review information to enhance rating prediction, which indicates the usefulness of review information. Second, BPMTMF performs best among the three baselines. The improvement margins over PMF and HFT are substantial, especially it yields 14.1% reduction of RMSE on the *DigiMu* dataset. It indicates the effectiveness of LMF methods for rating prediction. Third, our proposed models achieve similar or better performance compared with BPMTMF. As shown in [19], so far, BPMTMF has been the most competitive LMF method in rating prediction by comparing with DFC [13], LLORMA [10] and WEMAREC [4]. The margins that our models improve over BPMTMF are not large, while small RMSE reductions are likely to yield significant system performance improvement in practice [2]. The above results and analysis have shown the effectiveness of our proposed models.

Table 3. Performance comparisons of RMSE results on six datasets. Smaller is better.

Datasets	PMF	HFT	BPMTMF	ELMF
Music	0.953	0.954	0.905	**0.902**
Auto	1.004	0.981	0.970	**0.969**
Office	0.955	0.965	0.950	**0.947**
Patio	1.110	1.091	1.070	**1.063**
Video	1.352	1.312	1.301	**1.281**
DigiMu	1.363	1.313	1.171	**1.165**

Parameter Tuning. In our models, two important parameters are the number of item-topics (*i.e.,* K_1) and the number of word-topics (*i.e.,* K_2). Here, we examine how they affect the model performance. We select the *Music* dataset to report the tuning results and the rest datasets give the similar findings. We first find the optimal values by using cross validation. Then we fix one and vary the other. To tune K_1 and K_2, we use a grid search method by varying the values from 5 to 40 with a gap of 5. We present the tuning results in Fig. 3. Overall, the model performance is relatively robust for both parameters, and a range between 5 and 20 usually give good performance.

4.3 Evaluation on Model Interpretability

Besides the performance, interpretability is also important to consider in rating prediction. Here, we conduct evaluation experiments to examine the model interpretability. We start with an illustrative example to show the explainable recommendation results for our model.

Qualitative analysis of the ELMF model. In specific, we are particularly interested in two queries: (1) what are the semantics for the subgroups of users and items, and (2)

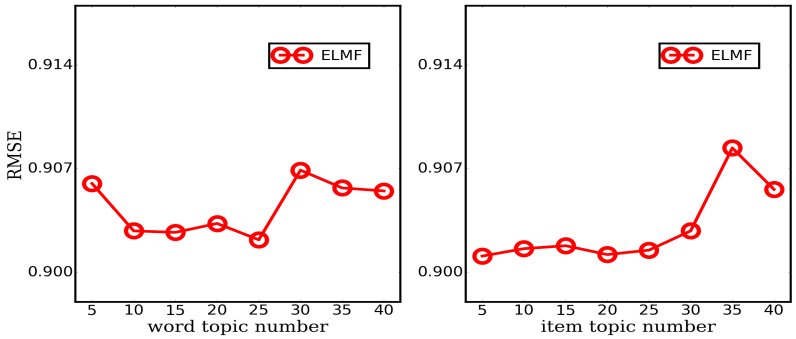

Fig. 3. Parameter tuning for the numbers of item- and word-topics. Smaller is better.

why a user likes or dislikes an item. To answer the first query, ELMF probabilistically clusters items using topic models, and each item-topic groups items with coherent semantics. We present three sample item-topics on *Music* dataset discovered by ELMF in Fig. 4. As we can see, the identified item-topics are clear and coherent. To answer the second query, we attach each topic with the most related words. Given an item-topic c, we select words by computing $\sum_{k=1}^{K_2} P(w|k)P(k|c,l)$ with a specific sentiment label l. Here, we consider two sentiment labels, *i.e.,* positive and negative. It is interesting to see different item-topics are related to different word-topics and different sentiment labels are related to different opinionated words.

Fig. 4. Three sample item-topics learned on *Music* dataset with associated words. Red and blue words come from positive and negative sentiment labels respectively. We use icons but not words to present each item for ease of understanding.

Quality evaluation of the identified item-topics. Above, we have shown the sample item-topics learned by our model. Now we quantitatively evaluate the quality of these item-topics. We select BPMTMF as a baseline since it is also able to generate item-topics. Intuitively, a good item-topic should group items from same categories. Our datasets provide the original Amazon category labels of these items in a three-layer hierarchy. We select the *purity* as the evaluation metrics. *Purity*[4] is a popular measure to evaluate the clustering quality. It compares the generated clusters against the gold results. We take the categorization of items on Amazon as the ground truth for evaluation. We present the comparison results in Table 4. Since each item is associated with three category labels, we can compute three different purity scores. Overall, our method has better purity performance compared with BPMTMF. The major reason is that BPMTMF only utilizes rating information while our model further leverages review information.

Table 4. Purity comparison with three-level category labels.

Category level	1^{st}	2^{nd}	3^{rd}
BPMTMF	0.659	0.622	0.246
ELMF	0.702	0.628	0.264

5 Conclusion

In this paper, we made an attempt to improve the interpretability of LMF methods by leveraging both item co-rated patterns and user reviews. We incorporated item-topics to construct meaningful subgroups, and associate them with opinionated word-topics to explain the semantics and sentiments for users ratings. By using topic-specific latent factors, our model yields competitive performance while the learned item- and word-topics give good interpretability to the recommendation results. Currently, our work only considers two sentiment labels, in the future, we will explore more levels of sentiments for more fine-grained explanations on the correlations between ratings and sentiments.

Acknowledgment

Xin Zhao was partially supported by the National Natural Science Foundation of China under grant 61502502 and the Beijing Natural Science Foundation under grant 4162032.

References

1. Ai, Q., Zhang, Y., Bi, K., Chen, X., Croft, W.B.: Learning a hierarchical embedding model for personalized product search. In: SIGIR. ACM (2017)
2. Amatriain, X., Mobasher, B.: The recommender problem revisited: morning tutorial. In: KDD (2014)
3. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. JMLR (2003)
4. Chen, C., Li, D., Zhao, Y., Lv, Q., Shang, L.: Wemarec: Accurate and scalable recommendation through weighted and ensemble matrix approximation. In: SIGIR. ACM (2015)

[4] http://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-clustering-1.html

5. Chen, X., Qin, Z., Zhang, Y., Xu, T.: Learning to rank features for recommendation over multiple categories. In: SIGIR. ACM (2016)
6. Dhillon, I.S., Mallela, S., Modha, D.S.: Information-theoretic co-clustering. In: KDD. ACM (2003)
7. Ganu, G., Elhadad, N., Marian, A.: Beyond the stars: Improving rating predictions using review text content. In: WebDB. Citeseer (2009)
8. He, X., Chen, T., Kan, M.Y., Chen, X.: Trirank: Review-aware explainable recommendation by modeling aspects. In: CIKM. ACM (2015)
9. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer (2009)
10. Lee, J., Kim, S., Lebanon, G., Singer, Y.: Local low-rank matrix approximation. In: ICML (2013)
11. Lin, X., Zhang, M., Zhang, Y.: Joint factorizational topic models for cross-city recommendation. In: APWeb-WAIM. Springer (2017)
12. Liu, H., He, J., Wang, T., Song, W., Du, X.: Combining user preferences and user opinions for accurate recommendation. Electronic Commerce Research and Applications (2013)
13. Mackey, L.W., Jordan, M.I., Talwalkar, A.: Divide-and-conquer matrix factorization. In: NIPS (2011)
14. McAuley, J., Leskovec, J.: Hidden factors and hidden topics: understanding rating dimensions with review text. In: Recsys. ACM (2013)
15. McAuley, J., Pandey, R., Leskovec, J.: Inferring networks of substitutable and complementary products. In: KDD. ACM (2015)
16. Mnih, A., Salakhutdinov, R.: Probabilistic matrix factorization. In: NIPS (2007)
17. Salakhutdinov, R., Mnih, A.: Bayesian probabilistic matrix factorization using markov chain monte carlo. In: ICML. ACM (2008)
18. Tan, Y., Zhang, M., Liu, Y., Ma, S.: Rating-boosted latent topics: Understanding users and items with ratings and reviews. In: IJCAI. pp. 2640–2646 (2016)
19. Wang, K., Zhao, W.X., Peng, H., Wang, X.: Bayesian probabilistic multi-topic matrix factorization for rating prediction. In: IJCAI (2016)
20. Wu, C.Y., Beutel, A., Ahmed, A., Smola, A.J.: Explaining reviews and ratings with paco: Poisson additive co-clustering. In: WWW (2016)
21. Wu, Y., Ester, M.: FLAME: A probabilistic model combining aspect based opinion mining and collaborative filtering. In: WSDM. ACM (2015)
22. Zhang, Y.: Explainable recommendation: Theory and applications. arXiv preprint arXiv:1708.06409 (2017)
23. Zhang, Y., Ai, Q., Chen, X., Croft, W.: Joint representation learning for top-n recommendation with heterogeneous information sources. In: CIKM. ACM (2017)
24. Zhang, Y., Lai, G., Zhang, M., Zhang, Y., Liu, Y., Ma, S.: Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In: SIGIR. ACM (2014)
25. Zhang, Y., Zhang, M., Liu, Y., Ma, S.: Improve collaborative filtering through bordered block diagonal form matrices. In: SIGIR. ACM (2013)
26. Zhang, Y., Zhang, M., Liu, Y., Ma, S., Feng, S.: Localized matrix factorization for recommendation based on matrix block diagonal forms. In: WWW. pp. 1511–1520. ACM (2013)