*

Adaptive Local Low-rank Matrix Approximation for Recommendation

HUAFENG LIU, Beijing Jiaotong University, Beijing Key Lab of Traffic Data Analysis and Mining, China
LIPING JING, Beijing Jiaotong University, Beijing Key Lab of Traffic Data Analysis and Mining, China
YUHUA QIAN, Shanxi University, School of Computer and Information Technology, China
JIAN YU, Beijing Jiaotong University, Beijing Key Lab of Traffic Data Analysis and Mining, China

Low-rank matrix approximation (LRMA) has attracted more and more attention in the community of recommendation. Even though LRMA-based recommendation methods (including Global LRMA and Local LRMA) obtain promising results, they suffer from the complicated structure of the large-scale and sparse rating matrix especially when the underlying system includes a large set of items with various types and a huge amount of users with diverse interests. Thus they have to predefine the important parameters, such as the rank of the rating matrix, the number of submatrices. Moreover, most existing Local LRMA methods are usually designed in a two-phase separated framework and do not consider the missing mechanisms of rating matrix. In this paper, a non-parametric unified Bayesian graphical model is proposed for **A**daptive **Lo**cal low-rank **M**atrix **A**pproximation (**ALoMA**). **ALoMA** has ability to simultaneously identify rating submatrices, determine the optimal rank for each submatrix, and learn the submatrix-specific user/item latent factors. Meanwhile, the missing mechanism is adopted to characterize the whole rating matrix. These four parts are seamlessly integrated and enhance each other in a unified framework . Specifically, the user-item rating matrix is adaptively divided into proper number of submatrices in **ALoMA** by exploiting Chinese Restaurant Process. For each submatrix, by considering both global/local structure information and missing mechanisms, the latent user/item factors are identified in an optimal latent space by adopting automatic relevance determination technique. We theoretically analyze the model's generalization error bounds and give an approximation guarantee. Furthermore, an efficient Gibbs sampling-based algorithm is designed to infer the proposed model. A series of experiments have been conducted on six real-world datasets (*Epinions*, *Douban*, *Dianping*, *Yelp*, *Movielens (10M)* and *Netflix*). The results demonstrate that **ALoMA** outperforms the state-of-the-art LRMA-based methods and can friendly provide interpretable recommendation results.

CCS Concepts: • **Information systems** → **Recommender systems**; **Clustering and classification**;

Additional Key Words and Phrases: Recommendation System, Clustering, Probabilistic Graphical Model

ACM Reference Format:
Huafeng Liu, Liping Jing, Yuhua Qian, and Jian Yu. 2018. Adaptive Local Low-rank Matrix Approximation for Recommendation. *ACM Transactions on Information Systems* *, *, Article * (August 2018), 32 pages. https://doi.org/0000001.0000001

This work was supported in part by the National Natural Science Foundation of China under Grant 61370129, Grant 61375062, Grant 61632004, and Grant 61773050.
Authors' addresses: Huafeng Liu, Beijing Jiaotong University, Beijing Key Lab of Traffic Data Analysis and Mining, No.3 Shangyuancun, Haidian District, Beijing, Beijing, 100044, China, huafeng@bjtu.edu.cn; Liping Jing, Beijing Jiaotong University, Beijing Key Lab of Traffic Data Analysis and Mining, Beijing, China, lpjing@bjtu.edu.cn; Yuhua Qian, Shanxi University, School of Computer and Information Technology, Taiyuan, China, jinchengqyh@126.com; Jian Yu, Beijing Jiaotong University, Beijing Key Lab of Traffic Data Analysis and Mining, Beijing, China, jianyu@bjtu.edu.cn.

1046-8188/2018/8-ART* $15.00
https://doi.org/0000001.0000001

1 INTRODUCTION

Matrix approximation (MA) is one of the most effective collaborative filtering (CF) methods for recommendation systems. It formulates the recommendation problem as an unobserved entries prediction task on sparse user-item rating matrix. The popular MA-based recommendation methods are designed under the assumption that the whole sparse rating matrix is low-rank. These methods, denoted as global low-rank matrix approximation (Global LRMA), are generally effective at estimating the global structure which simultaneously relates to most or all items. However, as stated in [17], they do not work well especially when there are a large set of items with various types and a huge amount of users with diverse interests. For example, an Amazon user shares similar tastes on books with a certain group of users, while having similar preferences with another group of users on movies.

To capture the user's diverse interests, more and more researchers consider the local associations among users/items. They firstly divided the original rating matrix into several submatrices [17, 31, 31, 35, 36], where each submatrix contains a set of like-minded users and the items that they are interested in. In each submatrix, the LRMA technique is adopted to model the submatrix-specific latent user/item factors. Finally, the missing entries are estimated with the weighted sum of the predictions in the submatrices [11]. This kind of methods is called as two-phase separated Local LRMA, i.e., they require a separate rating matrix partitioning phase that is decoupled from the low-rank submatrix approximation. To this end, these two phases are combined together via a unified probabilistic graphical model [4, 33]. Even though these studies have improved over Global LRMA methods to some extent, they may compromise recommendation quality because each submatrix only covers partial rating information for a particular user or item, which makes LRMA overemphasize the local structure but ignore the global information.

Table 1. The root mean square errors (RMSEs) of PMF for users/items (Yelp dataset) with different numbers of ratings when rank is 10 and 100. When the rank is 10, the users/items with less than 5 ratings achieve lower RMSEs than the cases when the rank is 100. This indicates that the PMF model overfits the users/items with less than 5 ratings when rank is 100. Moreover, PMF with rank 100 achieves higher accuracy than PMF with rank 10, but the improvement comes with sacrificed accuracy for the users and items with a small number of ratings, e.g., less than 5.

	rank = 10	rank = 100
♯ user ratings < 5	1.0573	1.0783
♯ user ratings > 50	0.8533	0.8423
♯ item ratings < 5	1.1531	1.1862
♯ item ratings > 50	0.8641	0.8539
All	0.8733	0.8661

For effective exploiting the global and local structure among rating matrix, researchers incorporate global and local latent factor identification [9] or embed a previously-learnt global structure into the local structure training process [10, 19]. However, these methods suffer from two main issues. One is how to adaptively determine the number of submatrices for capturing the local information rather than manual tuning [5]. The other is how to adaptively set the proper rank for each submatrix rather than fixing the rank for all submatrices. In real-world rating matrices, e.g., Movielens, Netflix and Yelp, users/items have a varying number of ratings. A case study is conducted on the Yelp dataset with the aid of global latent factor identification method (PMF) as shown in Table 1. It can be seen that lower rank is helpful for the users/items with less ratings, while higher rank benefits the users/items with more ratings. This study confirms that, in the large-scale rating matrix, users/items with a significantly varying number of ratings should be

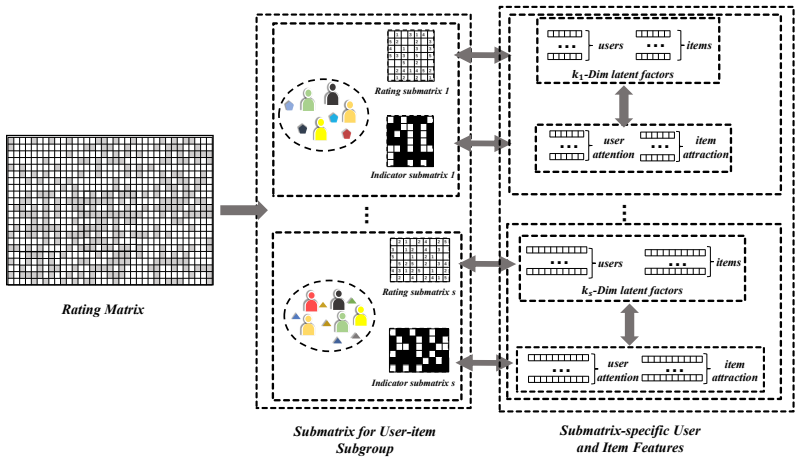


Fig. 1. The unified framework of the proposed **ALoMA** model.

treated differently. Mathematically speaking, internal submatrices with different ranks indeed coexist in the rating matrix with complex structure. Thus, it is necessary to adaptively find the user/item submatrices and adaptivley determine submatrix-specific rank. In this case, a submatrix with few ratings should be of low rank, while a submatrix with many ratings may be of a relatively higher rank. Meanwhile, in order to effective handle the recommendation data with *missing not at random* (MNAR) [24], the missing data should be considered when learning the latent user/item factors.

Motivated by these issues, we present an **A**daptive **Lo**cal low-rank **M**atrix **A**pproximation (**ALoMA**) model for recommendation. **ALoMA** is a non-parametric unified Bayesian graphical model to simultaneously identify submatrices, determine the optimal rank for each submatrix, learn the submatrix-specific user/item latent factors, and estimate the importance of latent feature with missing mechanisms, so that these four parts could be seamlessly integrated and enhance each other as shown in Fig.1. The main contributions of this paper are summarized as follows.

- **ALoMA** captures the local associations among a subset of users/items with the aid of Chinese Restaurant Process [13], which allows dynamic allocation of statistical capacity among clusters instead of predefining the number of clusters.
- **ALoMA** has ability to mine the latent user/item factors from each submatrix, where the submatrix-specific optimal rank is adaptively determined by automatic relevance determination technique [27, 34].
- **ALoMA** builds a unified Bayesian matrix factorization model for recommendation to leverage the local latent user/item factors, importance of latent features and global user/item bias.
- The generalizagtion error bound of **ALoMA** model is theoretically proved, which guarantees that the rating matrix can be approximated .
- An efficient Gibbs sampling-based algorithm is developed to infer **ALoMA** model, which can deal with large-scale recommendation data.
- A series of experiments are conducted on six real-world data sets (*Epinions*, *Douban*, *Dianping*, *Yelp*, *Movielens (10M)* and *Netflix*). By comparing with the existing state-of-the-art recommendation methods, **ALoMA** significantly improves the recommendation performance on both rating and ranking prediction [14], as well as friendly provide interpretable results.

The rest of the paper is organized as follows. The related work is discussed in Section 2. Section 3 gives the proposed **ALoMA** model and its theoretical analysis. The Gibbs sampling-based model

inference is given in Section 4. In Section 5, a series of experiments on six large-scale datasets (*Epinions*, *Douban*, *Dianping*, *Yelp*, *Movielens (10M)* and *Netflix*) are listed to demonstrate the performance of **ALoMA** by comparing with the state-of-the-art methods. Finally, a brief conclusion is given in Section 6.

2 RELATED WORK

Suppose there are n users and m items, $\mathbf{R} = \left[\mathbf{R}_{ij}\right]_{n\times m}$ indicates the user-item rating matrix, where the observed element $\mathbf{R}_{ij}$ records the rate of j-th item given by the i-th user. Usually, $\mathbf{R}$ is very sparse (less than 1% values are known), thus, our goal is to predict the rating value for unknown cell set $\Pi = \{(i,j) : \mathbf{R}_{ij}\ is\ missing\}$, i.e., $\mathbf{R}_\Pi$. Let $\mathbf{X} = \left[\mathbf{X}_{ij} \in \{0,1\}\right]_{n\times m}$ be an indicator matrix where $\mathbf{X}_{ij} = 1$ if $\mathbf{R}_{ij}$ is observed, and $\mathbf{X}_{ij} = 0$ if $\mathbf{R}_{ij}$ is missing. In this section, we review the existing work on recommendation with global low-rank matrix approximation (LRMA), local LRMA, global-local LRMA techniques and CF with non-random missing assumption.

2.1 LRMA with Global Perspective

Matrix Approximation (MA) is widely used in recommender systems to fill in the missing values of $\mathbf{R}$. To predict the missing values in original partially observed rating matrix, a popular way is to design an optimization problem minimizing the predicted rating error. In this case, the general MA-based recommendation methods can be formulated as

$$\min_{\theta} \sum_{\mathbf{R}_{ij}\neq 0} l(\mathbf{R}_{ij}, F(\theta)_{ij}) + \alpha_r \varphi(\theta), \tag{1}$$

Among them, $l(x)$ denotes the loss function, e.g., square loss. θ is the parameter set, and $F(\theta)$ is a model to fit the ground-truth rating matrix $\mathbf{R}$. $\varphi(\theta)$ is the regularization term to control the model complexity. The parameter α_r is to trade-off the loss term and regularization term. In literatures, the fitting model $F(\theta)$ is usually designed under the assumption that rating matrix is low-rank (e.g., matrix factorization-based collaborative filtering) [1, 12, 16, 29]. More specifically, the observed rating $\mathbf{R}$ is approximated by the inner product of two low-rank latent factors $\mathbf{U}$ and $\mathbf{V}$ with $\mathbf{R}_{ij} = \mathbf{U}_i{}^\top \mathbf{V}_j$, where $\mathbf{U}_i$ and $\mathbf{V}_j$ indicate the latent representation for user i and item j respectively.

These methods mentioned above are founded on a common assumption that rating data is *missing at random* (MAR). Thus those recommendation methods are modeled solely depend on observed rating data, as illustrated in (1). However, according to recent research on statistical theory of missing data and the reported evidences in collaborative filtering [21, 24], the data in recommendation system is *missing not at random* (MNAR). Compared with recommendation methods with MAR data, methods with MNAR data model not only the observed rating matrix $\mathbf{R}$, but the indicator matrix $\mathbf{X}$. Similarly, the recommendation model can be formulated as

$$\min_{\theta} \sum_{\mathbf{R}_{ij}\neq 0} l(\mathbf{R}_{ij}, F(\theta)_{ij}) + \sum_{\mathbf{X}_{ij}} l(\mathbf{X}_{ij}, P(\theta)_{ij}) + \alpha_r \varphi(\theta), \tag{2}$$

where the first term indicates the loss of rating prediction, same with (1). The second term is usually modeled by the likelihood of generating $\mathbf{X}$. Common variables θ can model the dependency between missing data and observed rating data. In literature, Marlin et al. [25] considered non-random missing data mechanism to multinomial mixture model. Hernandez-lobato et al. [15] presented the first practical implementation of a probabilistic matrix factorization model for MNAR data. Bence et al. [6] proposed an extended Bayesian probabilistic matrix factorization method, which integrates a flexible MNAR model with proper prior for the missing data. However, those methods construct missing mechanisms in a sophisticated and highly complex strategy, which leads to time-consuming parameter estimation. Recently, Ohsawa et al. [28] extended probabilistic matrix

factorization to take into account the dependency between why a user consumes an item and how that affects the rating behavior. These methods are pretty good to estimate the global structure where each user is simultaneously related to most or all items. However, at stated in [35], they can not work well especially when the items have various types and users have diverse interests.

2.2 LRMA with Local Perspective

To capture the local structure of the large-scale rating matrix, researchers attempted to introduce Local Low-Rank Matrix Approximation (Local LRMA) method. Its main idea is to divide the original large-scale rating matrix into several submatrices, so that the strong local associations can be exploited. To sufficiently reduce the approximation error, the original matrix is partitioned several times to get a set of approximated matrices $\{\mathbf{R}^{(1)}, \mathbf{R}^{(2)}, \cdots, \mathbf{R}^{(K)}\}$, and reconstructed with them and the corresponding weights in an ensemble manner as follows.

$$\min_{\theta} \sum_{\mathbf{R}_{ij} \neq 0} l(\mathbf{R}_{ij}, \frac{1}{\sum_{\tau=1}^{K} \mathbf{W}_{ij}^{(\tau)}} \sum_{\tau=1}^{K} \mathbf{W}_{ij}^{(\tau)} F^{(\tau)}(\theta)_{ij}) \tag{3}$$

where $F^{(\tau)}(\theta)$ is the τ-th model to fit the rating information, $\mathbf{W}_{ij}^{(\tau)}$ indicates the weight for the i-th user to the j-th item in the τ-th approximation.

In literature, a simple way is random selecting users/items to form submatrices [23], but it can not guarantee that the users in the same submatrix share the common interests and the items have the similar categories. To address this problem, several works [4, 11, 17, 33, 36] focused on how to partition matrix well. Lee et al. [17] proposed a local low-rank matrix approximation (LLORMA) method using a kernel smoothing nearest neighbors method to acquire local structure and represent rating matrix as a weighted sum of several local low-rank matrices. WEMAREC [11] employs Bregman co-clustering techniques to obtain several submatrices and adopts a rating distribution based weighting strategy to approximate original rating. bACCAMS [4] uses Bayesian co-clustering for local structure detection and build concise model for matrix approximation in a additive strategy. Wang et al. [33] presents a Bayesian formulation of local matrix factorization which integrates probabilistic matrix factorization with clustering (topic) detection in a joint model and acquire high recommendation accuracy. Zhang et al. [36] proposed a heuristic anchor-point selecting method to enhance local low-rank matrix approximation. However, these methods assign each user/item to only one single cluster, which make them can not well handle the users with multiple interests. Thus, researchers introduced an affiliation score to characterize the strength between user/item and the corresponding submatrices [35, 37].

2.3 LRMA with Local and Global Perspectives

Although the Local LRMA obtained promising results on rating prediction, it may suffer from insufficient data in each submatrix because each submatrix only covers partial rating information for a particular user or item, which makes LRMA overemphasize the local structure but ignore the global information. Actually, before Local LRMA, Global LRMA is one of the most popular methods in collaborative filtering, which can effectively mine overall structures by exploiting the whole ratings set.

Recently, researchers integrated Global LRMA and Local LRMA [9, 10, 19], called as Global-local LRMA. Chen et al. [9] exploit global information to unify global latent factors and local latent factors of users and items by a Gaussian mixture model to improve recommendation accuracy. Chen et al. [10] demonstrates an extension of clustering-based matrix approximation method, where a previous-trained standard MA model is introduced to capture global information, and local models are combined with global information for further prediction. It is a good way to unify local

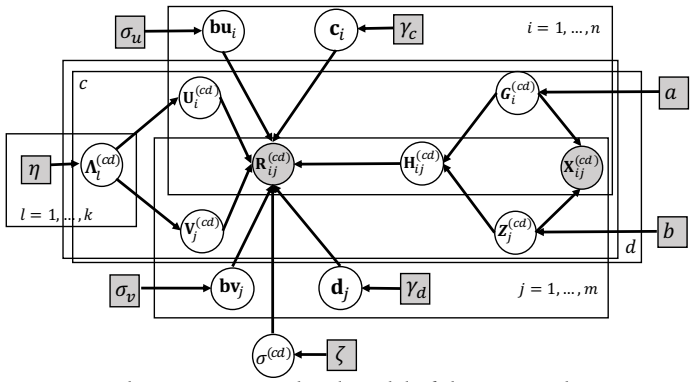


Fig. 2. The Bayesian graphical model of the proposed **ALoMA**.

associations in user-item submatrices and common associations among all users/items to improve the recommendation performance. However, like Local LRMA methods, these Global-local LRMA methods have to fix the latent space size when determining the user/item factors.

As mentioned in [18], adopting a fixed latent space size for all users and items can not perfectly model the internal structures of whole rating matrix, which may result in imperfect approximation as well as degraded prediction accuracy. Consequently, Li et al. [18] approximated the whole rating matrix with a mixture of global low-rank matrix approximation models (MRMA) with different ranks. However, MRMA has higher computational complexity because it has to try different ranks in a large range.

3 THE PROPOSED ALOMA METHOD

Given a rating matrix with n users and m items, our goal has three facets. The first one is to identify submatrices from rating matrix, where the number of submatrices can be adaptively determined rather than predefined. The second is to adaptively learn the user/item factors in an optimal latent space from each submatrix by integrating the global users/items information, and the third one is that the missing mechanism with missing not at random assumption is adopted to build a unified Bayesian matrix factorization model for the indicator values and observed rating scores in each submatrix. To reach this goal, we propose an adaptive local low-rank matrix approximation method (**ALoMA**) with the aid of non-parametric Bayesian graphical model, as shown in Fig. 2.

In **ALoMA**, $\mathbf{c}_i$ and $\mathbf{d}_j$ denote the i-th user's cluster assignment and j-th item's cluster assignment respectively. The ij-th rating entry ($\mathbf{R}_{ij}$) will be assigned to the (cd)-th submatrix $\mathbf{R}_{ij}^{(cd)}$ if the i-th user belongs to c-th cluster (i.e. $c = \mathbf{c}_i$) and the j-th item belongs to d-th cluster (i.e. $d = \mathbf{d}_j$). For each rating submatrix $\mathbf{R}^{(cd)} \in \mathbb{R}^{n_c \times n_d}$, we introduce an indicator matrix $\mathbf{X}^{(cd)} \in \{0,1\}^{n_c \times n_d}$ to denote whether the rating is missing, where n_c is the number of users in user cluster c and n_d is the number of items in item cluster d. $\mathbf{U}_i^{(cd)}$ and $\mathbf{V}_j^{(cd)}$ are the corresponding latent user/item factors in the (cd)-th submatrix. $\mathbf{\Lambda}_l^{(cd)}$ controls the generation procedure for the l-th component of the submatrix-specific latent factors. $\mathbf{H}_{ij}^{(cd)}$ captures the importance of each latent feature on every rating entry, which is modeled by $\mathbf{G}_i^{(cd)}$ and $\mathbf{Z}_j^{(cd)}$. Among them, $\mathbf{G}_i^{(cd)}$ denotes what features of items a user seeks to enjoy, and $\mathbf{Z}_j^{(cd)}$ denotes what features an item emphasizes, which are learnt from the indicator matrix $\mathbf{X}^{(cd)}$. Besides, $\mathbf{bu}_i$ and $\mathbf{bv}_j$ indicate the user/item bias to capture the global user/item information. All variables can be generated according to their distributions and more detail will be given in next subsections.

3.1 Generating Rating Submatrix

In order to capture local structure from large-scale rating matrix $\mathbf{R}$, our primary goal is to divide users/items into an appropriate number of subgroups, so that a user-cluster can be taken as a collection of users that express similar preferences over the items, and an item-cluster as a collection of items that have similar properties attracting users. To accomplish this, we apply *Chinese Restaurant Process (CRP)* [13] on rating matrix to co-cluster the rows and columns, which simultaneously allow dynamic allocation of statistical capacity among clusters. More specifically, the cluster assignments $\mathbf{c} = \{\mathbf{c}_i\}_{i=1}^{n}$ with $\mathbf{c}_i \in \{1, ..., k_n\}$ for user i and $\mathbf{d} = \{\mathbf{d}_j\}_{j=1}^{m}$ with $\mathbf{d}_j \in \{1, ..., k_m\}$ for item j can be generated as follows.

$$\mathbf{c}_i \sim \mathcal{CRP}(\gamma_c) \text{ and } \mathbf{d}_j \sim \mathcal{CRP}(\gamma_d) \tag{4}$$

CRP is adopted because it can be taken as a *Dirichlet* process based nonparametric Bayesian model, which allows for an unrestricted number of clusters, i.e., the number of user clusters k_n and the number of item clusters k_m can be adaptively allocated rather than being predefined. Specifically, the first user is assigned to the first cluster. The i-th user will be handled in two cases. In one hand, it may be assigned to an existing cluster c with probability proportional to the number of users already in cluster c. On the other hand, it may create a new cluster with probability proportional to the scaling parameter γ_c:

$$p(\mathbf{c}_i = c|\gamma_c) = \begin{cases} n_c/(n-1+\gamma_c) & \text{if } n_c > 0 \\ \gamma_c/(n-1+\gamma_c) & \text{if } n_c = 0 \end{cases} \tag{5}$$

An analogous expression is available for items with scaling parameter γ_d. Note that γ_c and γ_d control the probability of creating new cluster. Larger value will encourage larger number of clusters. In this case, the users belonging to the c-th cluster and items belonging to the d-th cluster form a subgroup (cd). Then, the whole rating matrix will be divided into $k_n \times k_m$ submatrices and each submatrix is denoted as $\mathbf{R}^{(cd)}$.

3.2 Generating Local Latent Factor in Optimal Space

As the existing local LRMA methods, once having the submatrices, we can determine the local latent user/item factors. Different from the existing models, **ALoMA** can adaptively determine the rank of each submatrix. This ability is necessary because different submatrices may have different ranks so that the local latent user/item factors can be mined in an optimal latent space. For instance, users or items in more sparse submatrix should be of a lower rank and be of a higher rank in a more dense submatrix.

For each submatrix $\mathbf{R}^{(cd)}$, let $\mathbf{U}^{(cd)}$ and $\mathbf{V}^{(cd)}$ indicate the latent user/item factors and can be characterized as

$$\begin{aligned} \mathbf{U}^{(cd)} &= [\mathbf{U}_1^{(cd)}, \ldots, \mathbf{U}_i^{(cd)}, \ldots, \mathbf{U}_{n_{(cd)}}^{(cd)}] \in \mathbb{R}^{r^{(cd)} \times n^{(cd)}}, \\ \mathbf{V}^{(cd)} &= [\mathbf{V}_1^{(cd)}, \ldots, \mathbf{V}_j^{(cd)}, \ldots, \mathbf{V}_{m_{(cd)}}^{(cd)}] \in \mathbb{R}^{r^{(cd)} \times m^{(cd)}}. \end{aligned}$$

where $n^{(cd)}$ and $m^{(cd)}$ be the number of users and items that the (cd)-th submatrix has, $r^{(cd)}$ be the optimal rank of this submatrix. Similar to the rating model of PMF, the l-th component of latent vector $\mathbf{U}_i^{(cd)}$ and $\mathbf{V}_j^{(cd)}$ is modeled via a *Gaussian* distribution with mean 0 and variance $\boldsymbol{\lambda}_l^{(cd)}$ as follows.

$$\begin{aligned} &\mathbf{U}_{li}^{(cd)} \sim \mathcal{N}(0, \boldsymbol{\lambda}_l^{(cd)}) \qquad \mathbf{V}_{lj}^{(cd)} \sim \mathcal{N}(0, \boldsymbol{\lambda}_l^{(cd)}), \\ &\boldsymbol{\lambda}_l^{(cd)} \sim \mathcal{IG}(\eta_a, \eta_b). \end{aligned} \tag{6}$$

The latent features are assumed independent to each other. In this case, $\mathbf{U}_i^{(cd)}$ and $\mathbf{V}_j^{(cd)}$ can be modeled with the following *Gaussian* distribution

$$\begin{aligned}\mathbf{U}_i^{(cd)} &\sim \prod_{l=1}^{r_{(cd)}} \mathcal{N}(0, \boldsymbol{\lambda}_l^{(cd)}) = \mathcal{N}(0, \Lambda^{(cd)}) \\ \mathbf{V}_j^{(cd)} &\sim \prod_{l=1}^{r_{(cd)}} \mathcal{N}(0, \boldsymbol{\lambda}_l^{(cd)}) = \mathcal{N}(0, \Lambda^{(cd)})\end{aligned} \tag{7}$$

where $\Lambda^{(cd)} = diag(\boldsymbol{\lambda}^{(cd)}) \in \mathbb{R}^{r^{(cd)} \times r^{(cd)}}$ is the covariance matrix of latent factors, which is a diagonal matrix whose l-th diagonal element is $\boldsymbol{\lambda}_l^{(cd)}$. To determine the optimal rank $r^{(cd)}$, we take advantage of the automatic relevance determination (*ARD*) techniques [27, 34]. Recall that each latent feature (l) is assumed having zero mean, thus, the feature with small variance ($\boldsymbol{\lambda}_l^{(cd)}$) will approach to zero. In this case, the latent features with small variance will not be used to characterize users and items. In other words, only the latent features with larger variance (empirically greater than 0.05) are useful to form the latent space. The rationale behind this is that, with a zero mean in the prior for this column, a very small variance indicates that this column will shrink to zero and hence will not contribute to explaining the data.

3.3 Estimating the Importance of Latent Feature with Missing Mechanisms

In the existing Local LRMA models, the latent features are determined only from the observed data because they assume that the ratings are missing at random. However, this assumption is violated in recommendation systems [6, 15, 24, 25, 28], i.e., the ratings are missing not at random.

In order to model the missing mechanism of the rating submatrix, we introduce the indicator $\mathbf{X}^{(cd)}$ for each rating submatrix $\mathbf{R}^{(cd)}$, where $\mathbf{X}_{ij}^{(cd)} = 1$ indicates that the rating $\mathbf{R}_{ij}^{(cd)}$ is observed, otherwise $\mathbf{X}_{ij}^{(cd)} = 0$. As demonstrated in [28], the indicator entry $\mathbf{X}_{ij}^{(cd)}$ can be modeled with the aid of user's attention and item's attraction. This is based on the underlying fact, a user consumes an item with some probability that depends on what factors the user seeks to enjoy (i.e., the attention of the user) and what factors the item emphasizes (i.e., attraction of the item).

Let $\mathbf{G}_i^{(cd)} \in \mathbb{R}^{r^{(cd)}}$ $(1 \le i \le n^{(cd)})$ and $\mathbf{Z}_j^{(cd)} \in \mathbb{R}^{r^{(cd)} \times m^{(cd)}}$ $(1 \le j \le m^{(cd)})$ be the i-th user's attention vector and j-th item's attraction vector in the cd-th submatrix. Since the indicator entry $\mathbf{X}_{ij}^{(cd)}$ denotes whether $\mathbf{R}_{ij}^{(cd)}$ is observed or not, we adopt *Bernoulli* distribution to characterize it, i.e.,

$$\mathbf{X}_{ij}^{(cd)} \sim \mathcal{B}(g(\mathbf{G}_i^{(cd)\top} \mathbf{Z}_j^{(cd)})) \tag{8}$$

where $\mathcal{B}(p)$ is the probability density function of the *Bernoulli* distribution with mean p. The function $g(x)$ is the logistic function $g(x) = \frac{1}{1+\exp(-x)}$ and bounds the rang of $\mathbf{G}_i^{(cd)\top} \mathbf{Z}_j^{(cd)}$ with $[0, 1]$. Meanwhile, the elements of $\mathbf{G}_i^{(cd)}$ and $\mathbf{Z}_j^{(cd)}$ independently follow *Beta* distribution

$$\mathbf{G}_i^{(cd)} \sim \prod_{l=1}^{r^{(cd)}} \mathcal{B}e(\mathbf{G}_{li}^{(cd)} | a_g, b_g) \qquad \mathbf{Z}_j^{(cd)} \sim \prod_{l=1}^{r^{(cd)}} \mathcal{B}e(\mathbf{Z}_{lj}^{(cd)} | a_z, b_z) \tag{9}$$

Here *Beta* distribution is adopted due to that *Beta* distribution is the conjugate prior of *Bernoulli* distribution, which enables the sampling-based algorithm more efficient. Following this distribution, the element $\mathbf{G}_{li}^{(cd)}$ has the support on $[0, 1]$, and all elements in $\mathbf{G}_i^{(cd)}$ are independent to each other, so that each user can simultaneously focus on multiple aspects. Similarly, the element $\mathbf{Z}_{lj}^{(cd)}$ has

the support on $[0, 1]$, and all elements in $\mathbf{Z}_j^{(cd)}$ are independent to each other, so that each item can simultaneously attracts users from several aspects.

Then for each pair of user-item, we can estimate the importance of each latent feature with user's attention and item's attraction as follows.

$$\mathbf{H}_{ij}^{(cd)} = \frac{1}{\mathbf{G}_i^{(cd)\top}\mathbf{Z}_j^{(cd)}} diag(\mathbf{G}_{1i}^{(cd)}\mathbf{Z}_{1j}^{(cd)}, \cdots, \mathbf{G}_{r^{(cd)}i}^{(cd)}\mathbf{Z}_{r^{(cd)}j}^{(cd)}) \in \mathbb{R}^{r^{(cd)}\times r^{(cd)}} \tag{10}$$

The element $\mathbf{G}_{li}^{(cd)}\mathbf{Z}_{lj}^{(cd)}$ takes a large value when both user's attention and items' attraction on the l-th latent feature have large values. The coefficient $\mathbf{G}_i^{(cd)\top}\mathbf{Z}_j^{(cd)}$ guarantees that the trace of $\mathbf{H}_{ij}^{(cd)}$ is 1.

3.4 Modeling Observed Ratings

Except for focusing on the local structure of rating matrix, we take into account the global information among all users and items. Specifically, we introduce the user/item bias to characterize the global user preference and item popularity. Let $\mathbf{bu}_i$ and $\mathbf{bv}_j$ denote the i-th user bias and j-th item bias respectively, and they are assumed following the *Gaussian* distribution with zero mean, i.e.,

$$\begin{aligned} \mathbf{bu}_i \sim \mathcal{N}(\mathbf{bu}_i|0, \sigma_u) \text{ and } \mathbf{bv}_j \sim \mathcal{N}(\mathbf{bv}_j|0, \sigma_v) \\ \sigma_u \sim \mathcal{IG}(u_a, u_b) \text{ and } \sigma_v \sim \mathcal{IG}(v_a, v_b) \end{aligned} \tag{11}$$

where σ_u and σ_v are user variance and item variance respectively. The *Inverse Gamma* distribution for variance is used here for generating a full Bayesian approach.

Following probabilistic matrix factorization, each observed rating belonging to (cd)-th submatrix, $\mathbf{R}_{ij}^{(cd)}$ is assumed following *Gaussian* distribution. Instead of mean $\mathbf{U}_i^{(cd)\top}\mathbf{V}_j^{(cd)}$, we model $\mathbf{R}_{ij}^{(cd)}$ with mean $\mathbf{U}_i^{(cd)\top}\mathbf{H}_{ij}^{(cd)}\mathbf{V}_j^{(cd)} + \mathbf{bu}_i + \mathbf{bv}_j$ and variance $\sigma^{(cd)}$as follows.

$$\begin{aligned} \mathbf{R}_{ij}^{(cd)} &\sim \mathcal{N}(\mathbf{U}_i^{(cd)\top}\mathbf{H}_{ij}^{(cd)}\mathbf{V}_j^{(cd)} + \mathbf{bu}_i + \mathbf{bv}_j, \sigma^{(cd)}) \\ \sigma^{(cd)} &\sim \mathcal{IG}(\zeta_a, \zeta_b). \end{aligned} \tag{12}$$

where $\mathbf{H}_{ij}^{(cd)}$ is a diagonal matrix, and each diagonal entry indicates the importance of the corresponding latent feature. It will back to traditional PMF for each submatrix when $\mathbf{H}_{ij}^{(cd)} = \frac{1}{r^{(cd)}}\mathbf{I}_{r^{(cd)}\times r^{(cd)}}$. Similar to user/item bias, the *Inverse Gamma* distribution is enforced on variance $\sigma^{(cd)}$ for full Bayesian approach.

3.5 ALoMA Model

The proposed **ALoMA** model, as shown in Fig. 2, integrates the above four components, i.e., generating the optimal number of submatrices, determining the local user/item factors in optimal latent space, estimating the importance of latent features and modeling the observed ratings. To form a full Bayesian approach, all parameters $\Theta = \{\eta_a, \eta_b, \zeta_a, \zeta_b, \sigma_u, \sigma_v, \gamma_c, \gamma_d, \sigma^{(cd)}, a_g, b_g, a_z, b_z\}$ are generated via the corresponding distributions which are conjugate to the likelihood terms. The whole generative process of **ALoMA** is summarized in Algorithm 1.

Our goal is to maximize the posterior of all latent variables, according to Bayesian rule, which can be implemented by maximizing the joint probability of observed data and all latent variables.

ALGORITHM 1: ALoMA Generative Process

1. For each hyperparameter:
   a) Draw hyperparameters $\{\eta_a, \eta_b, \zeta_a, \zeta_b, u_a, u_b, v_a, v_b, \zeta_a, \zeta_b\}$
2. For each variance:
   a). Draw $\sigma_u \sim \mathcal{IG}(u_a, u_b)$, $\sigma_v \sim \mathcal{IG}(v_a, v_b)$
3. For each user i:
   a) Draw user cluster $\mathbf{c}_i \sim \mathcal{CRP}(\gamma_c)$
   b) Draw user bais $\mathbf{bu}_i \sim \mathcal{N}(0, \sigma_u)$
3. For each item j:
   a) Draw item cluster $\mathbf{d}_j \sim \mathcal{CRP}(\gamma_d)$
   b) Draw item bais $\mathbf{bv}_j \sim \mathcal{N}(0, \sigma_v)$
4. For each submatrix $\mathbf{R}^{(cd)}$:
   a). Draw $\sigma^{(cd)} \sim \mathcal{IG}(\zeta_a, \zeta_b)$
   b) For each latent dimension l:
      i). Draw $\boldsymbol{\lambda}_l^{(cd)} \sim \mathcal{IG}(\eta_a, \eta_b)$
   c) For each user i in subgroup (cd)
      i). Draw latent user factors $\mathbf{U}_i^{(cd)} \sim \mathcal{N}(0, \boldsymbol{\Lambda}^{(cd)})$
      ii). Draw user attention $\mathbf{G}_i^{(cd)} \sim \prod_{l=1}^{r_{(cd)}} \mathcal{B}e(\mathbf{G}_{li}^{(cd)} | a_g, b_g)$
   d) For each item j in subgroup (cd)
      i). Draw latent item factors $\mathbf{V}_j^{(cd)} \sim \mathcal{N}(0, \boldsymbol{\Lambda}^{(cd)})$
      ii). Draw item attraction $\mathbf{Z}_j^{(cd)} \sim \prod_{l=1}^{r_{(cd)}} \mathcal{B}e(\mathbf{Z}_{lj}^{(cd)} | a_z, b_z)$
   e) For each element in indicator submatrix $\mathbf{X}^{(cd)}$
      i). Draw boolean variable $\mathbf{X}_{ij}^{(cd)} \sim \mathcal{B}(g(\mathbf{G}_i^{(cd)\top} \mathbf{Z}_j^{(cd)}))$
   f) For each observed rating assigned by user i to item j in submatrix $\mathbf{R}^{(cd)}$:
      i). Draw the rating $\mathbf{R}_{ij}^{(cd)} \sim \mathcal{N}(\mathbf{U}_i^{(cd)\top} \mathbf{H}_{ij}^{(cd)} \mathbf{V}_j^{(cd)} + \mathbf{bu}_i + \mathbf{bv}_j, \sigma^{(cd)})$

Following the generative process of **ALoMA** model, the joint probability can be expressed as:

$$
\begin{aligned}
& p(\mathbf{U}^{(cd)}, \mathbf{V}^{(cd)}, \mathbf{G}^{(cd)}, \mathbf{Z}^{(cd)}, \mathbf{c}, \mathbf{d}, \mathbf{bu}, \mathbf{bv}, \boldsymbol{\Lambda}^{(cd)}, \mathbf{R}, \mathbf{X} | \Theta) \\
&= \prod_{cd} \Big[\prod_{\mathbf{R}_{ij}^{(cd)} \neq 0} \mathcal{N}(\mathbf{R}_{ij}^{(cd)} | \mathbf{U}_i^{(cd)\top} \mathbf{H}_{ij}^{(cd)} \mathbf{V}_j^{(cd)} + \mathbf{bu}_i + \mathbf{bv}_j, \sigma^{(cd)}) \prod_{\mathbf{X}_{ij}^{(cd)}} \mathcal{B}(\mathbf{X}_{ij}^{(cd)} | g(\mathbf{G}_i^{(cd)\top} \mathbf{Z}_j^{(cd)})) \Big] \\
&\times \prod_{cd} \Big[\prod_{i=1}^{n^{(cd)}} \mathcal{N}(\mathbf{U}_i^{(cd)} | 0, \boldsymbol{\Lambda}^{(cd)}) \prod_{j=1}^{m^{(cd)}} \mathcal{N}(\mathbf{V}_j^{(cd)} | 0, \boldsymbol{\Lambda}^{(cd)}) \prod_{l=1}^{r^{(cd)}} \mathcal{IG}(\boldsymbol{\Lambda}_l^{(cd)} | \eta_a, \eta_b)] \Big] \\
&\times \prod_{cd} \Big[\prod_{i=1}^{n^{(cd)}} \prod_{l=1}^{r^{(cd)}} \mathcal{B}e(\mathbf{G}_{li}^{(cd)} | a_g, b_g) \prod_{j=1}^{m^{(cd)}} \prod_{l=1}^{r^{(cd)}} \mathcal{B}e(\mathbf{Z}_{lj}^{(cd)} | a_z, b_z) \Big] \\
&\times \prod_{i=1}^{n} \mathcal{CRP}(\mathbf{c}_i | \gamma_c) \prod_{j=1}^{m} \mathcal{CRP}(\mathbf{d}_j | \gamma_d) \\
&\times \prod_{i=1}^{n} \mathcal{N}(\mathbf{bu}_i | 0, \sigma_u^2) \prod_{j=1}^{m} \mathcal{N}(\mathbf{bv}_j | 0, \sigma_v^2).
\end{aligned}
\tag{13}
$$

where the first term is to characterize the observed rating and indicator matrix via (12) and (8), the second term aims to determine the submatrix-specific latent user/item factors and the corresponding latent feature variance and adaptively identify the optimal number of user/item clusters, the third

term is to capture user attention and item attraction for each factors, the forth term is to assign users and items to clusters, and the last term is to mine the global user and item biases so that the observed rating could be approximated more accurately.

3.6 Theoretical Analysis

In this subsection, we will theoretically analyze the generalization error bounds of the proposed **ALoMA** model. Here, the mean squared error (MSE), equivalent to Frobenius norm, between the ground truth ratings ($\mathbf{R}$) and the predicated ratings ($\hat{\mathbf{R}}$) with n users and m items, is used as the metric to establish the error bound of the proposed method, i.e.

$$E(\hat{\mathbf{R}}) = \frac{1}{nm}\sum_{\hat{\mathbf{R}}_{ij}}(\mathbf{R}_{ij} - \hat{\mathbf{R}}_{ij})^2 = \frac{1}{nm}||\mathbf{R} - \hat{\mathbf{R}}||_F$$

According to the generative process of **ALoMA**, it can be seen that the whole rating matrix is approximated by several submatrices, thus, we firstly prove the error bound of each submatrix. To make the theorems more convincing, we make several standard assumptions: (1) each submatrix $\mathbf{R}^{(cd)}$ is incoherent[8, 32], (2) each submatrix is well-conditioned[8], and (3) the number of users is larger than items in each submatrix ($n^{(cd)} \geq m^{(cd)}$). As shown in Theorem 7 in [7], a theoretical bound to matrix completion problem is existing as follows.

Theorem 3.1. *If submatrix $\mathbf{R}^{(cd)}$ is well-conditioned and incoherent such that $|\Omega| \geq C\mu^2 m^{(cd)} r^{(cd)} log^6 m^{(cd)}$, then with high probability $1 - \sqrt[3]{m^{(cd)}}$, $\mathbf{R}^{(cd)}$ satisfies*

$$||\mathbf{R}^{(cd)} - \hat{\mathbf{R}}^{(cd)}||_F \leq 4\varepsilon\sqrt{\frac{(2+\rho)m^{(cd)}}{\rho}} + 2\varepsilon$$

where $\rho = \frac{|\mathbf{R}^{(cd)}|}{n^{(cd)}m^{(cd)}}$, $\varepsilon = \max(\hat{\mathbf{R}}) - \min(\hat{\mathbf{R}})$ and Ω is the set of observed ratings.

This theorem indicates that the prediction error of each submatrix $\hat{\mathbf{R}}^{(cd)}$ is bounded. Then, we can analyze the generalization error bound of the proposed **ALoMA** method on whole rating matrix via the following theorem.

Theorem 3.2. *If each submatrix satisfied theorem 3.1, then with high probability $1 - \delta$, $\hat{\mathbf{R}}$ is divided into $k_n \times k_m$ submatrices satisfies*

$$P\left(||\mathbf{R} - \hat{\mathbf{R}}||_F \leq 4\varepsilon k_m\sqrt{\frac{(2+\rho)nk_n}{\rho}} + 2\varepsilon k_n k_m\right) \geq 1 - \delta$$

where $\delta = \sqrt[3]{2nk_n k_m^2}$.

Proof. According to the triangle inequality of Frobenius norm, we have

$$||\mathbf{R} - \hat{\mathbf{R}}||_F \leq \sum_{(cd)}||\mathbf{R}^{(cd)} - \hat{\mathbf{R}}^{(cd)}||_F.$$

Applying theorem 3.1 on each submatrix, and using Cauchy Inequality

$$\sum_{(cd)}\sqrt{m^{(cd)}} \leq \sqrt{k_n k_m \sum_{(cd)} m^{(cd)}} \leq k_n\sqrt{mk_m}$$

we can obtain the error bound of $\hat{\mathbf{R}}$ as stated above. The adjustments confidence level $\sqrt[3]{2nk_n k_m^2}$ is obtained using the union bound. □

Meanwhile, we show that the approximation guarantee of the whole rating matrix based on partial observed values via the following theorem.

Theorem 3.3. *With high probability* $1-\delta$ *over the set of observed entities* Ω*, for predictive rating matrix* $\boldsymbol{R} \in \mathbb{R}^{n\times m}$ *and* $\delta > 0$*, we have that* $\hat{\mathbf{R}}$ *satisfies*

$$P\left(E(\hat{\mathbf{R}}) - E_{\Omega}(\hat{\mathbf{R}}) \leq \sqrt{\frac{-\log\delta}{2|\Omega|}\varepsilon^2}\right) \geq 1-\delta$$

Proof. Based on Hoeffding Inequality

$$P(E(\hat{\mathbf{R}}) - E_{\Omega}(\hat{\mathbf{R}}) \geq \epsilon) \leq \exp(-2|\Omega|^2\epsilon^2/\sum_{\hat{\mathbf{R}}_{ij}}\varepsilon^2) = \exp(-2|\Omega|\epsilon^2/\varepsilon^2),$$

and setting $\epsilon = \sqrt{\frac{-\log\delta}{2|\Omega|}\varepsilon^2}$, we obtain the approximate guarantee of **ALoMA** as stated above. □

4 ALOMA MODEL INFERENCE

In **ALoMA** model, we have to learn the variables including user/item cluster assignments $\mathbf{c}_i$ and $\mathbf{d}_j$ for submatrix $\mathbf{R}^{(cd)}$ determination, latent user factors $\mathbf{U}^{(cd)}$, item factors $\mathbf{V}^{(cd)}$, the attention of users $\mathbf{G}^{(cd)}$ and the attraction of items $\mathbf{Z}^{(cd)}$ in each submatrix, global user/item bias $\mathbf{bu}$ and $\mathbf{bv}$ in the whole rating matrix, and several corresponding variances $\sigma_u, \sigma_v, \sigma^{(cd)}, \mathbf{\Lambda}^{(cd)}$. It is difficult to directly maximize the joint probability due to the complex coupling of variables and hyper-parameters. In this section, we make use of Gibbs sampling algorithm [3] for both inference and variable updating. We alternatively infer user/item cluster assignment, update the latent user/item factors, user attention, item attraction and global user/item bias and infer the corresponding parameters until the algorithm converges.

4.1 Inferring the User/item Cluster Assignment

For sampling the user cluster membership, we need to specify the posterior probability of the user cluster assignment. We introduce a sampling for the discrete cluster membership. Specifically, the conditional probability for each user cluster c consists of three parts, the *CRP* prior, the likelihood of the indicator values and the likelihood of the observed ratings.

Fixing all user/item local latent factors, user attention, item attraction, global bias and variances (all these variables are denoted as *rest*), we can derive the conditional distribution $p(\mathbf{c}_i = c|rest)$ for each user (i.e., the conditional probability that i-th user belongs to the existing cluster c) as below

$$\begin{aligned}
p(\mathbf{c}_i = c|rest) &= p(\mathbf{c}_i = c|\mathbf{R}, \mathbf{X}, \mathbf{U}, \mathbf{V}, \mathbf{G}, \mathbf{Z}, \mathbf{bu_i}, \mathbf{bv}, \mathbf{d}, \gamma_c, \gamma_d, \sigma^{(cd)}) \\
&\propto \mathcal{CRP}(\mathbf{c}_i = c|\gamma_c) \times \prod_d \prod_{\mathbf{X}_{ij}^{(cd)}} \mathcal{B}(\mathbf{X}_{ij}^{(cd)}|g({\mathbf{G}_i^{(cd)}}^\top \mathbf{Z}_j^{(cd)})) \\
&\quad \times \prod_d \prod_{\mathbf{R}_{ij}^{(cd)}\neq 0} \mathcal{N}(\mathbf{R}_{ij}^{(cd)}|{\mathbf{U}_i^{(cd)}}^\top \mathbf{H}_{ij}^{(cd)} \mathbf{V}_j^{(cd)} + \mathbf{bu}_i + \mathbf{bv}_j, \sigma^{(cd)}) \\
&\propto \frac{n_c}{\gamma_c + n - 1} \times \prod_d \prod_{\mathbf{X}_{ij}^{(cd)}} [g({\mathbf{G}_i^{(cd)}}^\top \mathbf{Z}_j^{(cd)})]^{\mathbf{X}_{ij}^{(cd)}} [1 - g({\mathbf{G}_i^{(cd)}}^\top \mathbf{Z}_j^{(cd)})]^{1-\mathbf{X}_{ij}^{(cd)}} \\
&\quad \times \prod_d \prod_{\mathbf{R}_{ij}^{(cd)}\neq 0} \frac{1}{\sqrt{2\pi\sigma^{(cd)}}} \exp\big(-\frac{1}{2\sigma^{(cd)}}(\mathbf{e}_{ij}^{(cd)})^2\big)
\end{aligned} \tag{14}$$

Meanwhile, the probability that the i-th user assigned to a new user cluster c_n is

$$\begin{aligned} p(\mathbf{c}_i = c_n)|rest) \propto & \frac{\gamma_c}{\gamma_c + n - 1} \times \prod_d \prod_{\mathbf{X}_{ij}^{(c_n d)}} [g(\mathbf{G}_i^{(c_n d)\top} \mathbf{Z}_j^{(c_n d)})]^{\mathbf{X}_{ij}^{(c_n d)}} [1 - g(\mathbf{G}_i^{(c_n d)\top} \mathbf{Z}_j^{(c_n d)})]^{1-\mathbf{X}_{ij}^{(c_n d)}} \\ & \times \prod_d \prod_{\mathbf{R}_{ij}^{(c_n d)} \neq 0} \frac{1}{\sqrt{2\pi\sigma^{(c_n d)}}} \exp\big(-\frac{1}{2\sigma^{(c_n d)}} (\mathbf{e}_{ij}^{(c_n d)})^2\big) \end{aligned} \tag{15}$$

where $\mathbf{e}_{ij}^{(cd)} = \mathbf{R}_{ij}^{(cd)} - \mathbf{U}_i^{(cd)\top} \mathbf{H}_{ij}^{(cd)} \mathbf{V}_j^{(cd)} - \mathbf{bu}_i - \mathbf{bv}_j$ indicates the approximation error for each observed rating, γ_c is a parameter to control the probability that creates new cluster, n_c denotes the number of users assigned to the cluster c, n is the total number of users.

With the similar strategy in (14) and (15), the posterior distribution for an existing item cluster d and a new cluster d_n can be approximated by

$$\begin{aligned} p(\mathbf{d}_j = d|rest) \propto & \frac{n_d}{\gamma_d + m - 1} \times \prod_c \prod_{\mathbf{R}_{ij}^{(cd)} \neq 0} \frac{1}{\sqrt{2\pi\sigma^{(cd)}}} \exp\big(-\frac{1}{2\sigma^{(cd)}} (\mathbf{e}_{ij}^{(cd)})^2\big) \\ & \times \prod_c \prod_{\mathbf{X}_{ij}^{(cd)}} [g(\mathbf{G}_i^{(cd)\top} \mathbf{Z}_j^{(cd)})]^{\mathbf{X}_{ij}^{(cd)}} [1 - g(\mathbf{G}_i^{(cd)\top} \mathbf{Z}_j^{(cd)})]^{1-\mathbf{X}_{ij}^{(cd)}} \end{aligned} \tag{16}$$

and

$$\begin{aligned} p(\mathbf{d}_j = d_n)|rest) \propto & \frac{\gamma_d}{\gamma_d + m - 1} \times \prod_c \prod_{\mathbf{R}_{ij}^{(cd_n)} \neq 0} \frac{1}{\sqrt{2\pi\sigma^{(cd_n)}}} \exp\big(-\frac{1}{2\sigma^{(cd_n)}} (\mathbf{e}_{ij}^{(cd_n)})^2\big) \\ & \times \prod_c \prod_{\mathbf{X}_{ij}^{(cd_n)}} [g(\mathbf{G}_i^{(cd_n)\top} \mathbf{Z}_j^{(cd_n)})]^{\mathbf{X}_{ij}^{(cd_n)}} [1 - g(\mathbf{G}_i^{(cd_n)\top} \mathbf{Z}_j^{(cd_n)})]^{1-\mathbf{X}_{ij}^{(cd_n)}} \end{aligned} \tag{17}$$

4.2 Updating Local User/Item Latent Factor

For each submatrix, fixing the global bias, user attention, item attraction and variances (all these variables are denoted as *rest*), we can update the local user/item latent factors.

Due to the use of conjugate priors for the parameters and hyperparameters in the **ALoMA**, the conditional distribution over the local user latent vector $\mathbf{U}_i^{(cd)}$ is a *Gaussian* distribution,

$$\begin{aligned} p(\mathbf{U}_i^{(cd)}|rest) &= p(\mathbf{U}_i^{(c)}|\mathbf{R}^{(cd)}, \mathbf{V}^{(cd)}, \mathbf{G}^{(cd)}, \mathbf{Z}^{(cd)}, \mathbf{bu}_i, \mathbf{bv}, \mathbf{d}, \mathbf{\Lambda}^{(cd)}, \sigma^{(cd)}) \\ &\propto \prod_{\mathbf{R}_{ij}^{(cd)} \neq 0} \mathcal{N}(\mathbf{R}_{ij}^{(cd)}|\mathbf{U}_i^{(cd)\top} \mathbf{H}_{ij}^{(cd)} \mathbf{V}_j^{(cd)} + \mathbf{bu}_i + \mathbf{bv}_j, \sigma^{(cd)}) \times \mathcal{N}(\mathbf{U}_i^{(cd)}|0, \mathbf{\Lambda}^{(cd)}) \\ &\propto \mathcal{N}(\mathbf{U}_i^{(cd)}|\mu_u^*, [\Lambda_u^*]^{-1}) \end{aligned} \tag{18}$$

where

$$\begin{aligned} \Lambda_u^* &= \frac{1}{\sigma^{(cd)}} \sum_{\mathbf{R}_{ij}^{(cd)} \neq 0} \mathbf{H}_{ij}^{(cd)} \mathbf{V}_j^{(cd)} \mathbf{V}_j^{(cd)\top} \mathbf{H}_{ij}^{(cd)\top} + \mathbf{\Lambda}^{(cd)} \\ \mu_u^* &= [\Lambda_u^*]^{-1} \frac{1}{\sigma^{(cd)}} \sum_{\mathbf{R}_{ij}^{(cd)} \neq 0} (\mathbf{R}_{ij}^{(cd)} - \mathbf{bu}_i - \mathbf{bv}_j) \mathbf{H}_{ij}^{(cd)} \mathbf{V}_j^{(cd)} \end{aligned} \tag{19}$$

Note that the conditional distribution over the user latent feature matrix $\mathbf{U}^{(cd)}$ factorizes into the product of conditional distributions over the individual user feature vectors:

$$p(\mathbf{U}^{(cd)}) = \prod_{i=1}^{n^{(cd)}} \mathcal{N}(\mathbf{U}_i^{(cd)}|0, \mathbf{\Lambda}^{(cd)}).$$

Therefore we can easily speed up the sampler by sampling from these conditional distributions in parallel. The speedup could be substantial, particularly when the number of users is large in submatrix. Meanwhile, the sampler in different submatrix can be processed in parallel.

The local item latent factor $\{\mathbf{V}_j^{(cd)}\}$ can be learned similarly, which are omitted here due to page limitation.

4.3 Updating User Attention and Item Attraction

For each indicator submatrix $\mathbf{X}^{(cd)}$, fixing the local user latent factors $\mathbf{U}^{(cd)}$, item factors $\mathbf{V}^{(cd)}$, the attraction of item $\mathbf{Z}^{(\mathbf{cd})}$ and global biases, we can derive the conditional probability of the l-th dim attention of user $\mathbf{G}_{li}^{(cd)}$ via the user attention prior, the likelihood of the indicator values and the likelihood of the observed ratings.

$$\begin{aligned}
p(\mathbf{G}_{li}^{(cd)}|rest) &= p(\mathbf{G}_{li}^{(c)}|\mathbf{R}^{(cd)}, \mathbf{V}^{(cd)}, \mathbf{X}^{(cd)}, \mathbf{Z}^{(cd)}, \mathbf{bu}_i, \mathbf{bv}, \mathbf{d}, \sigma^{(cd)}, a_g, b_g) \\
&\propto \prod_{\mathbf{R}_{ij}^{(cd)} \neq 0} \mathcal{N}(\mathbf{R}_{ij}^{(cd)}|{\mathbf{U}_i^{(cd)}}^\top \mathbf{H}_{ij}^{(cd)} \mathbf{V}_j^{(cd)} + \mathbf{bu}_i + \mathbf{bv}_j, \sigma^{(cd)}) \\
&\quad \times \prod_{\mathbf{X}_{ij}^{(cd)}} \mathcal{B}(\mathbf{X}_{ij}^{(cd)}|g({\mathbf{G}_i^{(cd)}}^\top \mathbf{Z}_j^{(cd)})) \times \mathcal{B}e(\mathbf{G}_{li}^{(cd)}|a_g, b_g) \\
&= \prod_{\mathbf{R}_{ij}^{(cd)} \neq 0} \mathcal{N}(\mathbf{R}_{ij}^{(cd)}|{\mathbf{U}_i^{(cd)}}^\top \mathbf{H}_{ij}^{(cd)} \mathbf{V}_j^{(cd)} + \mathbf{bu}_i + \mathbf{bv}_j, \sigma^{(cd)}) \\
&\quad \times \mathcal{B}e(\mathbf{G}_{li}^{(cd)}|a_g + \sum_{\mathbf{X}_{ij}^{(cd)}} \left[\mathbf{X}_{ij}^{(cd)} \mathbf{Z}_{lj}^{(cd)}\right], b_g + n_c n_d - \sum_{\mathbf{X}_{ij}^{(cd)}} \left[\mathbf{X}_{ij}^{(cd)} \mathbf{Z}_{lj}^{(cd)}\right])
\end{aligned} \tag{20}$$

To efficiently approximate the conditional probability $p(\mathbf{G}_{li}^{(cd)}|rest)$, we replace Beta distribution $\mathcal{B}e(\mathbf{G}_{li}^{(cd)}|\bar{a}_g, \bar{b}_g)$ with an optimal Gaussian distribution $\mathcal{N}(\mathbf{G}_{li}^{(cd)}|\hat{\mu}_g, \hat{\sigma}_g)$. Note that $\bar{a}_g = a_g + \sum_{\mathbf{X}_{ij}^{(cd)}} \left[\mathbf{X}_{ij}^{(cd)} \mathbf{Z}_{lj}^{(cd)}\right]$ and $\bar{b}_g = b_g + n_c n_d - \sum_{\mathbf{X}_{ij}^{(cd)}} \left[\mathbf{X}_{ij}^{(cd)} \mathbf{Z}_{lj}^{(cd)}\right]$. Here the variational inference is adopted to find the optimal parameters $(\hat{\mu}_g, \hat{\sigma}_g)$, which can be implemented by minimizing the Kullback-Leibler (KL) divergence between the true distribution $(\mathcal{B}e(\mathbf{G}_{li}^{(cd)}|\bar{a}_g, \bar{b}_g))$ and the variational distribution ($\mathcal{N}(\mathbf{G}_{li}^{(cd)}|\mu_g, \sigma_g)$) as follows.

$$\arg\min_{\mu_g, \sigma_g} KL_{[0,1]}\big(\mathcal{N}(\mathbf{G}_{li}^{(cd)}|\mu_g, \sigma_g)||\mathcal{B}e(\mathbf{G}_{li}^{(cd)}|\bar{a}_g, \bar{b}_g)\big) \triangleq \min_{\mu_g, \sigma_g} \int_0^1 \mathcal{N}(\mathbf{G}_{li}^{(cd)}|\mu_g, \sigma_g) \log \frac{\mathcal{N}(\mathbf{G}_{li}^{(cd)}|\mu_g, \sigma_g)}{\mathcal{B}e(\mathbf{G}_{li}^{(cd)}|\bar{a}_g, \bar{b}_g)} d\mathbf{G}_{li}^{(cd)}$$

which is equal to maximizing an *Evidence Lower BOund* ($\mathcal{L}$):

$$\arg\max_{\mu_g, \sigma_g} \mathcal{L}(\mathbf{G}_{li}^{(cd)}) = \max_{\mu_g, \sigma_g} \mathbb{E}_{\mathcal{N}}[\log \mathcal{N}(\mathbf{G}_{li}^{(cd)}|\mu_g, \sigma_g)] - \mathbb{E}_{\mathcal{B}e}[\log \mathcal{B}e(\mathbf{G}_{li}^{(cd)}|\bar{a}_g, \bar{b}_g)]. \tag{21}$$

In this case, the optimal variational parameters can be obgtained by iteratively updating μ_g and σ_g with the aid of stochastic gradient ascent technique. In each iteration, we calculate the gradient of

$\mathcal{L}$ over μ_g and σ_g via

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \mu_g} &= -\frac{\mu_g}{\sigma_g^2} + \mu_g\left(\frac{\Gamma(\bar{a}_g+\bar{b}_g)}{\Gamma(\bar{a}_g)\Gamma(\bar{b}_g)}\mathbf{G}_{li}^{(cd)^{\bar{a}_g-1}}(1-\mathbf{G}_{li}^{(cd)})^{\bar{b}_g-1}\right)\\ \frac{\partial \mathcal{L}}{\partial \sigma_g} &= \left[\mu_g\left(\frac{\Gamma(\bar{a}_g+\bar{b}_g)}{\Gamma(\bar{a}_g)\Gamma(\bar{b}_g)}\mathbf{G}_{li}^{(cd)^{\bar{a}_g-1}}(1-\mathbf{G}_{li}^{(cd)})^{\bar{b}_g-1}\right)-\sigma_g\right]^{-1}\end{aligned} \tag{22}$$

Then these variables can be updated as follows.

$$\begin{aligned}\mu_g^{(\tau+1)} &= \mu_g^{(\tau)} + \rho^{(\tau)}\partial\mathcal{L}/\partial\mu_g^{(\tau)}\\ \sigma_g^{(\tau+1)} &= \sigma_g^{(\tau)} + \rho^{(\tau)}\partial\mathcal{L}/\partial\sigma_g^{(\tau)}\end{aligned} \tag{23}$$

where τ is the iteration index and $\rho^{(\tau)}$ indicates the learning rate in τ-th iteration. Then the optimal variables ($\hat{\mu}_g$ and $\hat{\sigma}_g$) will be obtained when the iterative processing converges. Consequently, the conditional distribution over the l-th dim attention of user $\mathbf{G}_{li}^{(cd)}$ can be re-wirtten as

$$\begin{aligned}p(\mathbf{G}_{li}^{(cd)}|rest) &\propto \prod_{\mathbf{R}_{ij}^{(cd)}\neq 0}\mathcal{N}(\mathbf{R}_{ij}^{(cd)}|\mathbf{U}_i^{(cd)^\top}\mathbf{H}_{ij}^{(cd)}\mathbf{V}_j^{(cd)}+\mathbf{bu}_i+\mathbf{bv}_j,\sigma^{(cd)})\times\mathcal{B}e(\mathbf{G}_{li}^{(cd)}|\bar{a}_g,\bar{b}_g)\\ &\propto \prod_{\mathbf{R}_{ij}^{(cd)}\neq 0}\mathcal{N}(\mathbf{R}_{ij}^{(cd)}|\mathbf{U}_i^{(cd)^\top}\mathbf{H}_{ij}^{(cd)}\mathbf{V}_j^{(cd)}+\mathbf{bu}_i+\mathbf{bv}_j,\sigma^{(cd)})\times\mathcal{N}(\mathbf{G}_{li}^{(cd)}|\hat{\mu}_g,\hat{\sigma}_g)\\ &\propto \mathcal{N}(\mathbf{G}_{li}^{(cd)}|\mu_g^*,\sigma_g^*)\end{aligned} \tag{24}$$

where

$$\begin{aligned}\sigma_g^* &= \frac{1}{\sigma^{(cd)}}\sum_{\mathbf{R}_{ij}^{(cd)}\neq 0}\left(\mathbf{U}_{li}^{(cd)}\mathbf{Z}_{lj}^{(cd)}\mathbf{V}_{lj}^{(cd)}\right)^2+\hat{\sigma}_g\\ \mu_g^* &= [\sigma_g^*]^{-1}\Big[\frac{\hat{\mu}_g}{\hat{\sigma}_g}+\frac{1}{\sigma^{(cd)}}\sum_{\mathbf{R}_{ij}^{(cd)}\neq 0}(\mathbf{R}_{ij}^{(cd)}-\mathbf{bu}_i-\mathbf{bv}_j)\mathbf{U}_{li}^{(cd)}\mathbf{Z}_{lj}^{(cd)}\mathbf{V}_{lj}^{(cd)}\Big]\end{aligned} \tag{25}$$

Similar to local latent factors, we can easily speed up the sampler by sampling these conditional distributions for different users in parallel. Sampling the attraction of item $\mathbf{Z}^{(cd)}$ can be done by an analogous sampler, which are omitted here due to page limitation.

4.4 Updating Global User/Item Bias

The conditional distribution over the global user bias $\mathbf{bu}$ is a *Gaussian* distribution by giving other variables

$$\begin{aligned}p(\mathbf{bu}_i|rest) &= p(\mathbf{bu}_i|\mathbf{R},\mathbf{X},\mathbf{U},\mathbf{V},\mathbf{G},\mathbf{Z},\mathbf{bv},\sigma^{(cd)},\sigma_u)\\ &\propto \prod_{cd}\prod_{j=1}^{m}[\mathcal{N}(\mathbf{R}_{ij}^{(cd)}|\mathbf{U}_i^{(cd)^\top}\mathbf{H}_{ij}^{(cd)}\mathbf{V}_j^{(cd)}+\mathbf{bu}_i+\mathbf{bv}_j,\sigma^{(cd)})]^{\mathbf{X}_{ij}^{(cd)}}\times\mathcal{N}(\mathbf{bu}_i|0,\sigma_u)\\ &\propto \mathcal{N}(\mu_u^*,\sigma_u^*)\end{aligned} \tag{26}$$

with

$$\begin{aligned}\sigma_u^* &= \frac{1}{\sigma_u}+\sum_{cd}\mathbf{I}_i^{(cd)}[\frac{1}{\sigma^{(cd)}}\sum_{j=1}^{m}\mathbf{X}_{ij}^{(cd)}]\\ \mu_u^* &= \frac{1}{\sigma_u^*}\sum_{cd}\mathbf{I}_i^{(cd)}[\sigma^{(cd)}\sum_{j=1}^{m}\mathbf{X}_{ij}^{(cd)}(\mathbf{e}_{ij}^{(cd)}+\mathbf{bv}_j)]\end{aligned} \tag{27}$$

where $\mathbf{I}_i^{(cd)}$ indicates whether user i belongs to subgroup (cd). The updating process for global item bias $\mathbf{bv}$ is analogous to $\mathbf{bu}$.

4.5 Inferring Variance

To generate a full Bayesian approach, four variances $\{\mathbf{\Lambda}^{(cd)}, \sigma^{(cd)}, \sigma_u, \sigma_v\}$ (about local latent factor, rating in submatrix, global user/item bias respectively) follow the *Inverse Gamma* distribution. They can be generated by the following conditional probability:

$$\begin{aligned} p(\mathbf{\Lambda}_l^{(cd)}|rest) &\propto \mathcal{IG}(\mathbf{\Lambda}_l^{(cd)}|\eta_a^*, \eta_b^*) \\ p(\sigma^{(cd)}|rest) &\propto \mathcal{IG}(\sigma^{(cd)}|\zeta_a^*, \zeta_b^*) \\ p(\sigma_u|rest) &\propto \mathcal{IG}(\sigma_u|u_a^*, u_b^*) \\ p(\sigma_v|rest) &\propto \mathcal{IG}(\sigma_v|v_a^*, v_b^*) \end{aligned} \tag{28}$$

with

$$\begin{aligned} \eta_a^* &= \eta_a + \frac{n_c + n_d}{2}, & \eta_b^* &= \eta_b + \Big[\sum_{i\in c} \mathbf{I}_i^{(c)} (\mathbf{U}_{li}^{(cd)})^2 + \sum_{j\in d} \mathbf{I}_j^{(d)} (\mathbf{V}_{lj}^{(cd)})^2\Big]/2, \\ \zeta_a^* &= \zeta_a + \frac{n_{(cd)}}{2}, & \zeta_b^* &= \zeta_b + \Big[\sum_{<i,j>\in(cd)} \mathbf{I}_{ij}^{(cd)} (\mathbf{e}_{ij}^{(cd)})^2\Big]/2, \\ u_a^* &= u_a + \frac{n}{2}, & u_b^* &= u_b + \frac{\sum_{i=1}^n \mathbf{bu}_i{}^2}{2}, \\ v_a^* &= v_a + \frac{m}{2}, & v_b^* &= v_b + \frac{\sum_{j=1}^m \mathbf{bv}_j{}^2}{2}. \end{aligned}$$

where n_c is the number of users in cluster c, n_d is the number of items in cluster d and $n_{(cd)}$ is the number of observed ratings in submatrix $\mathbf{R}^{(cd)}$. $\mathbf{I}_i^{(c)}$ and $\mathbf{I}_j^{(d)}$ are indicator function, $\mathbf{I}_i^{(c)} = 1$ means user i belongs to user cluster c and $\mathbf{I}_i^{(c)} = 0$ otherwise. The overall Gibbs sampling learning algorithm for **ALoMA** model is summarized in Algorithm 2.

After obtaining the above variables, the unknown rating can be predicted via

$$p(\hat{\mathbf{R}}_{ij}^{(cd)}) \sim \frac{1}{T-S} \sum_{t=S}^{T} \mathcal{N}(\mu^{(t)}, (\sigma^{(cd)})^{(t)}), \tag{29}$$

where $\mu^{(t)} = ((\mathbf{U}_i^{(cd)})^{(t)})^\top (\mathbf{H}_{ij}^{(cd)})^{(t)} (\mathbf{V}_j^{(cd)})^{(t)} + \mathbf{bu}_i^{(t)} + \mathbf{bv}_j^{(t)}$. t is the iteration index and T is the total number of iterations. In experiments, the top S (e.g., $S = 100$) iterations are taken as bun-in period.

4.6 Computational Complexity Analysis

The overall iterative process will be performed until it converges. To efficiently implement the learning process, the main part can be done in parallel.

In each iteration, updating global bias $\mathbf{bu}_i$ and $\mathbf{bv}_j$ costs $O(\sum_d (r^{(cd)})^2 n_i^{(cd)})$ and $O(\sum_c (r^{(cd)})^2 n_j^{(cd)})$, where $n_i^{(cd)}$ is the number of items rated by the i-th user in (cd)-th submatrix, and $r^{(cd)}$ is the rank of the (cd)-th submatrix which contains n_c users and n_d items. The running time for updating user cluster assignment $\mathbf{c}_i$ in (14) and (15) is $O\big(\sum_d (r^{(cd)})^2 (n_i^{(cd)} + n_c n_d)\big)$.

ALGORITHM 2: Sampling Process for **ALoMA**

Input: Rating matrix $\mathbf{R}$ with n users and m items, parameter γ_c and γ_d in *CRP*, parameters $\{\eta, \zeta, u, v\}$ in *Inverse Gamma* distribution, parameters $\{a_g, b_g, a_z, b_z\}$ for *Beta* distribution, *ARD* variance threshold ϵ_λ

```
while sampler not converged do
  for each user i do
    Sample user bias bu_i by (26) and user assignment c_i by (14) or (15);
  end for
  for each item j do
    Sample item bias bv_j and item assignment d_j by (16) or (17);
  end for
  for each subgroup (cd) do
    for each user i in subgroup (cd) do
      Sample the attention of user G_li^(cd) by (20) for each dimension;
      Sample latent user factor U_i^(cd) by (18);
    end for
    for each item j in subgroup (cd) do
      Sample the attraction of item Z_lj^(cd) for each dimension;
      Sample latent item factor V_j^(cd);
    end for
  end for
  for all variances do
    Sample variances σ^(cd), Λ^(cd), σ_u and σ_v by (28);
  end for
  Reduce certain dimension if corresponding ARD variance is less than the predefined threshold ε_λ
end while
```

Output: stable value of $\mathbf{U}^{(cd)}, \mathbf{V}^{(cd)}, \mathbf{G}^{(cd)}, \mathbf{Z}^{(cd)}$ for each subgroup, $\mathbf{bu}$ and $\mathbf{bv}$.

The computational complexity for updating $\mathbf{U}_i^{(cd)}$ in (18) and $\mathbf{G}_i^{(cd)}$ in (20) are $O((r^{(cd)})^3 n_i^{(cd)} + (r^{(cd)})^3)$ and $O(r^{(cd)}(n_i^{(cd)} + n_c n_d))$ respectively. Then the worst computational complexity for updating variables related to one user is

$$\mathcal{CUI} = \max_{\{i,(cd)\}} \left((r^{(cd)})^3(n_i^{(cd)}+1) + r^{(cd)}(n_i^{(cd)} + n_c n_d)\right) + \max_i \left(\sum_d (r^{(cd)})^2(n_i^{(cd)} + n_c n_d) + \sum_d (r^{(cd)})^2 n_i^{(cd)}\right) \tag{30}$$

Similarity, the worst complexity for updating the variables related to one item is

$$\mathcal{CIJ} = \max_{\{j,(cd)\}} \left((r^{(cd)})^3(n_j^{(cd)}+1) + r^{(cd)}(n_j^{(cd)} + n_c n_d)\right) + \max_j \left(\sum_c (r^{(cd)})^2(n_j^{(cd)} + n_c n_d) + \sum_c (r^{(cd)})^2 n_j^{(cd)}\right) \tag{31}$$

Consequently, the overall cost of inferring process with T iterations is $O\Big(T(n\mathcal{CUI} + m\mathcal{CIJ})\Big)$. Obviously, variables related to each user or item can be updated in parallel on multicores since the samplers related to users and items are independent with each other, and the total computational complexity is linearly scalable to the number of users and items, thus, it is practical for handling large-scale dataset.

5 EXPERIMENTS

In this section, we evaluate the proposed **ALoMA** on six datasets by comparing with the state-of-the-art methods in both rating prediction and ranking estimation tasks.

5.1 Datasets

In experiments, six widely used datasets, *Epinions*, *Douban*, *Dianping*, *Yelp*, *Movielens* with 10M ratings, and *Netflix* with 100M ratings are used to validate the recommendation performance. *ML 10M* and *Netflix* come from movie domain, *Epinions* belongs to online products domain, *Dianping* includes various products (e.g., movies, books and music), and *Yelp* and *Dianping* are related to local business domain. The ratings are ordinal values on the scale 1 to 5, more information is summarized in Table 2 and described as follows.

Table 2. Summary of experimental datasets

	Epinions	*Douban*	*Dianping*	*Yelp*	*Movielens (10M)*	*Netflix*
♯ users (n)	49,290	129,490	147,918	9,489,338	71,567	480,189
♯ items (m)	139,738	58,541	11,123	156,638	10,681	17,770
♯ ratings (n_R)	664,824	16,830,839	2,149,675	4,731,265	10,000,054	100,000,000
RDensity	0.010%	0.222%	0.13%	0.00032%	1.31%	1.17%
$\overline{n}$	5	288	44	30	936	5,627
$\overline{m}$	14	130	23	0.5	143	208

$\overline{n}$: the average number of users interested in each item
$\overline{m}$: the average number of items rated by each user
RDensity: the percentage of non-zero entries in rating matrix

Epinions[1]: This dataset was collected in a 5-week crawl (November/December 2003) from the Epinions.com website and published in [26]. In Epinions, users can write reviews and give ratings to various products and also establish social relations with others.

Douban[2]: This dataset was crawled by [22] from Douban which provides user ratings, reviews and recommendation services for movies, books and music. Users can assign 5-scale integer ratings (from 1 to 5) to various products. It also provides Facebook-like social networking services, which allows users to find their friends through their email accounts.

Dianping[3]: Li[20] crawled it from the real social network-based recommender system Dianping, which is a leading local business search and review platform in China. The dataset contains business items in Shanghai, a social network of users, and the ratings from April 2003 to November 2013.

Yelp[4]: This dataset was provided by the tenth round of the Yelp Dataset Challenge. Yelp is a local business recommendation platform, where users can obtain reviews and recommendations of best restaurants, shopping, nightlife, food, entertainment, things to do, services and more.

Movielens (10M)[5]: This dataset was collected by GroupLens Research and made available rating data sets from the MovieLens web site, which is a web-based recommender system and virtual community that recommends movies for its users to watch, based on their film preferences using collaborative filtering of members' movie ratings and movie reviews.

Netflix[6]: This dataset was provided from Netflix Prize competition, which was held by Netflix, an online DVD-rental and video streaming service.

Among them, *Netflix 100M* and *Yelp* are two largest datasets in terms of rating size and user/item size. The number of users and items in *Yelp* are 9,489,338 and 156,638 respectively. *Netflix 100M* has the largest number of ratings (100,000,000). Moreover, in *Yelp* and Netflix, each item (i.e., business on *Yelp* and movie on *Netflix*) is marked by one or more categores, and all items belong to 1240

[1]http://www.trustlet.org/downloaded_epinions.html
[2]https://www.cse.cuhk.edu.hk/irwin.king.new/pub/data/douban
[3]https://i.cs.hku.hk/ hli2/data.html
[4]https://www.yelp.com/dataset/challenge
[5]https://grouplens.org/datasets/movielens/
[6]https://www.netflixprize.com

categories in *Yelp* and 18 categories in *Netflix*. For each dataset, we adopt 5-fold cross-validation for training and testing. Specifically, each data set is randomly split into five equal-sized subsets, four subsets are used as the training set and the left as testing set in each round. Five rounds are conducted to ensure all subsets are tested, and the average test performance is recorded as the final results.

5.2 Methodology

5.2.1 Baselines. We compare the performance of **ALoMA** with three categories of recommendation methods including Global LRMA models, Local LRMA models and Global-local LRMA models. The detailed settings about baselines are summarized as follows:

- Global LRMA: PMF [30] is a classical Global probabilistic matrix factorization-based collaborative filtering method, BPMF [30] is a global LRMA with a fully Bayesian probabilistic matrix factorization. MRMA [18] is a global method with mixture-rank LRMA model. MF-MNAR [15] is an extended PMF model by considering missing value mechanisms. GPMF [28] incorporates a consumption-rating model into PMF via modeling the attention of users and the attraction of items.
- Local LRMA: DFC [23], LLORMA [17] and WEMAREC [11] are ensemble two-phase Local low-rank matrix approximation methods, which vary in matrix partition method. bACCAMS [4] is a unified Bayesian local matrix approximation method.
- Global-local LRMA: SMA [19] finds out local entry sets globally that are harder to predict than average, and adopt standard LRMA on subset except hard-predicted entries. MPMA [9] exploits global information to unify global and local latent user/item factors by a *Gaussian* mixture model. GLOMA [10] combines a previous-trained standard MA model with local information for further prediction.

5.2.2 Metrics. In order to validate the prediction quality, two well-known evaluation metrics, Root Mean Square Error (RMSE) and Mean Absolute Error (MAE), are adopted in experiments, which are defined as:

$$RMSE = \sqrt{\frac{1}{|\Omega_t|}\sum_{(i,j)\in\Omega_t}(\mathbf{R}_{ij} - \hat{\mathbf{R}}_{ij})^2} \qquad MAE = \frac{1}{|\Omega_t|}\sum_{(i,j)\in\Omega_t}|\mathbf{R}_{ij} - \hat{\mathbf{R}}_{ij}|$$

where Ω_t is the set of testing entries. $\mathbf{R}_{ij}$ is the true rating value that the i-th user gave to the j-th item in testing set. $\hat{\mathbf{R}}_{ij}$ is the predicted value from different methods. The smaller RMSE and MAE values indicate better recommendation results.

Apart from measuring rating prediction performance, several ranking metrics are adopted to measure the item ranking accuracy of different algorithms. We use Recall ($Re@K(i)$ for top K on user i) and Precision ($Pre@K(i)$ for top K on user i) as

$$Re@K(i) = \frac{|R(i) \cap T(i)|}{|T(i)|} \qquad Pre@K(i) = \frac{|R(i) \cap T(i)|}{K}$$

where $R(i) = \{j \in \Omega(i) | \mathbf{R} \geq 4\}$ denotes the set of recommended items to user i. $\Omega(i)$ denotes the set of items that user i has rated in testing set. $T(i) = \{j \in \Omega(i) | \hat{\mathbf{R}}_{ij} \geq 4\}$ denotes the set of favorite items of user i. The larger Recall and Precision value, the better the ranking.

Meanwhile, the normalized discounted cumulative gain (NDCG) is adopted in our experiments, which is defined as:

$$NDCG@K(i) = \frac{DCG@K(i)}{IDCG@K(i)}$$

where DCG is defined as:

$$DCG@K(i) = \sum_{j\in\Omega(i)} \frac{2^{R_{ij}} - 1}{\log_2(j+1)}$$

and IDCG is the DCG value with perfect ranking. The larger NDCG value indicates better ranking performance.

5.2.3 Parameter setting. The optimal experimental settings for each method are determined either by experiments or suggested by the authors. For **ALoMA**, the initial rank is set as 300 for all datasets. The parameters (γ_c, γ_d) for automatically detecting the number of user/item clusters are $\{\gamma_c = 4, \gamma_d = 6\}$ for *Epinions*, $\{\gamma_c = 9, \gamma_d = 4\}$ for *Douban*, $\{\gamma_c = 8, \gamma_d = 4\}$ for *Dianping*, $\{\gamma_c = 9, \gamma_d = 7\}$ for *Yelp*, $\{\gamma_c = 8, \gamma_d = 5\}$ for *Movielens (10M)* and $\{\gamma_c = 8, \gamma_d = 5\}$ for *Netflix*, respectively. The hyperparameters are set with $\eta_a = 3, \eta_b = 0.5, \zeta_a = 5, \zeta_b = 0.4, u_a = v_a = 5, u_b = v_b = 0.3, a_g = a_z = 0.5, b_g = b_z = 1$ for six datasets. The predefined threshold for controlling the variance in *ARD* is set as $\epsilon_\lambda = 0.005$. The samples from the burn-in period (top 100 iterations) are discarded.

5.3 Results and Discussion

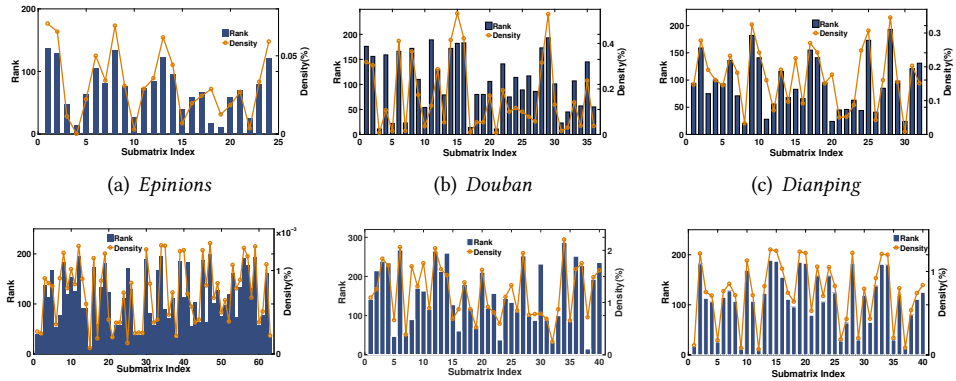
In this subsection, we investigate the proposed model **ALoMA** from five facets. Firstly, a series of experiments are conducted to demonstrate the effect of optimal rank in each submatrix. Secondly, **ALoMA** is compared with three kinds of matrix approximation-based recommendation methods (Global LRMA, Local LRMA and Global-local LRMA) from three views including *All Users*, *Near-cold-start Users* and *Long-tail Items* in terms of rating prediction and ranking estimation. Thirdly, we demonstrate the performance of **ALoMA** when dealing with sparsity problem. Next, we show that the recommendation results provided by **ALoMA** make sense with the aid of auxiliary information. Finally, the running times of **ALoMA** with different training data size are demonstrated to show its scalability.

Table 3. The number of submatrices and the minimal/maximal rank values determined by **ALoMA** on different datasets.

Datasets		*Epinions*	*Douban*	*Dianping*	*Yelp*	*Movielens (10M)*	*Netflix*
♯ Submatrices		24	36	32	63	40	40
Submatrix-specifit optimal rank	minima	11	11	12	11	15	10
	maxima	136	193	171	199	287	189

5.3.1 Effect of Optimal Submatrix-specific Rank Detection. **ALoMA** has the ability to adaptively identify submatrices and determine the optimal rank for each submatrix. Thus, the first experiment is conducted to investigate **ALoMA** on selecting the optimal rank for each submatrix. Table 3 lists the number of submatrices adaptively obtained by **ALoMA** in different datasets, which are obtained by 4×6, 9×4, 8×4, 9×7, 8×5 and 8×5 ($k_n \times k_m$, here k_n is the number of user clusters and k_m is the number of item clusters) user-item co-clustering on six datasets *Epinions*, *Douban*, *Dianping*, *Yelp*, *Movielens (10M)* and *Netflix* respectively. It is interesting to observe that the number of user or item clusters is different on six datasets, which is reasonable because that the items' types and users' interests are totally different in different datasets.

Unlike the existing methods which set the same rank on all submatrices, **ALoMA** determine the submatrix-specific optimal rank via automatic relevance determination technique with a threshold (0.0001 in all experiments). Thus each submatrix can be modeled in an optimal latent space. Table 3 also lists the minimal and maximal optimal rank value determined by **ALoMA** on different datasets. It can be seen that the rank values have a big variance. Moreover, Fig. 3 demonstrates the optimal rank in each submatrix determined by **ALoMA**, and the corresponding submatrices' densities are plotted in line chart for different datasets. Obviously, the rank values are almost proportional to the submatrices' density. This result confirms that submatrix with few ratings should be of low rank, otherwise be of high rank, which is consistent with the conclusion given by [18].



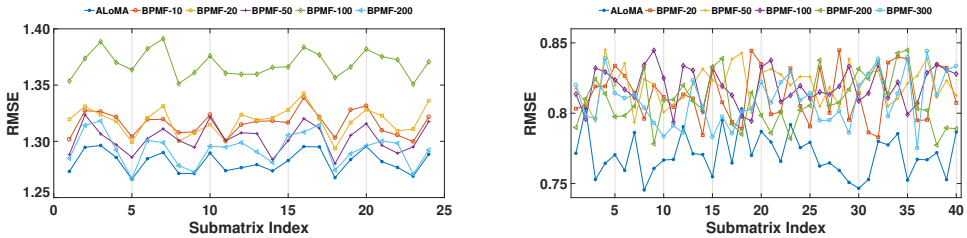
(a) *Epinions* (b) *Douban* (c) *Dianping*

(d) *Yelp* (e) *Movielens (10M)* (f) *Netflix*

Fig. 3. Demonstrating the optimal rank in each submatrix determined by **ALoMA** for (a) *Epinions*, (b) *Douban*, (c) *Dianping*, (d) *Yelp*, (e) *Movielens (10M)* and (f) *Netflix*.

Table 4. Comparing recommendation performance of **ALoMA** and BPMF with different ranks in terms of MAE and RMSE on six datasets.

Datasets	Metrics	BPMF-10	BPMF-20	BPMF-50	BPMF-100	BPMF-200	BPMF-300	**ALoMA**
Epinions	MAE	1.1068	1.1042	1.1121	1.1146	1.1137	1.1250	**1.0124**
	RMSE	1.3491	1.3471	1.3533	1.3562	1.3521	1.3674	**1.2613**
Douban	MAE	0.6153	0.6121	0.6196	0.6208	0.6211	0.6331	**0.5645**
	RMSE	0.7562	0.7543	0.7615	0.7621	0.7623	0.7746	**0.7123**
Dianping	MAE	0.8431	0.8452	0.8491	0.8450	0.8459	0.8486	**0.7346**
	RMSE	1.0378	1.0389	1.0436	1.0390	1.0396	1.0421	**0.9213**
Yelp	MAE	0.6682	0.6676	0.6681	0.6715	0.6746	0.6686	**0.6214**
	RMSE	0.8461	0.8431	0.8461	0.8496	0.8512	0.8474	**0.8137**
Movielens (10M)	MAE	0.6341	0.6322	0.6346	0.6362	0.6352	0.6338	**0.5842**
	RMSE	0.8248	0.8217	0.8245	0.8252	0.8248	0.8234	**0.7547**
Netflix	MAE	0.6236	0.6222	0.6231	0.6251	0.6246	0.6243	**0.6241**
	RMSE	0.8443	0.8435	0.8441	0.8463	0.8453	0.8448	**0.7873**



(a) *Epinions* (b) *Movielens (10M)*

Fig. 4. Demonstrating the RMSE of each submatrix obtained by **ALoMA** and BPMF with fixed rank on *Epinions* and *Movielens (10M)* dataset.

Additionally, we compare the recommendation accuracy of **ALoMA** against BPMF with different fixed ranks {10, 20, 50, 100, 200, 300} on the whole rating matrix. As shown in Table 4, when the rank increases from 10 to 300, the performance of BPMF is unstable (in terms of RMSE). Taking *Movielens (10M)* dataset as an example, BPMF with rank 50 achieves better performance than rank 100 but worse than rank 300. The reason is that matrix approximation with fixed rank cannot model all users and items well, so that some users and items are either under-fitted or over-fitted. Nevertheless,

BPMF with all ranks are inferior to **ALoMA**, because each user or item can be modeled in an optimal low-rank subspace, which can alleviate overfitting or underfitting problem to some extent.

In order to demonstrate the recommendation performance of each submatrix with optimal rank, we investigate the recommendation error (RMSE) on each submatrix applied via BPMF with different ranks on two datasets ({10, 20, 50, 100, 200} for *Epinions* and {20, 50, 100, 200, 300} for *Movielens (10M)*), as shown in Fig. 4. It can be seen that BPMF with different ranks are definitely inferior to **ALoMA** on each submatrix. This result further confirms that it is proper to determine the optimal rank for each submatrix rather than using the Global LRMA method with fixed rank on the whole matrix.

5.3.2 Comparison of Recommendation Performance. The second experiment is designed to evaluate **ALoMA** by comparing with twelve baselines. In addition to the **ALoMA**, we also study a simplified version of **ALoMA** (denoted as **ALoMA-s**), which only considers adaptive submatrix generation with *CRP* and adaptive latent factors determination with *ARD*. Three different views are adopted to evaluate the recommendation performance of the proposed method and the existing methods. Among them, *All Users* view indicates that all ratings are used as the testing set. *Near-cold-start Users* view means that the users who rate less than five items will be involved in the testing set. *Long-tail Items* view only considers the items which are in long tail, i.e., the items with total 20% ratings [2].

Table 5 shows the recommendation performance on testing *All Users*. The first, second and third best results are marked in bold, with star supscript and underlined respectively. As expected, **ALoMA-s** significantly improves the recommendation performance by comparing with all baselines. This result indicates that adaptive determining local structure and submatrix-specific rank benefits finding the optimal user/item latent factors. Meanwhile, **ALoMA** outperforms **ALoMA-s** because the former model sufficiently incorporates the missing mechanism and global information. In the baselines, the Global LRMA methods with non-random missing data (MF-MNAR, GPMF) outperform the methods with random missing data assumption except for MRMA. MRMA achieves the second best performance from *All Users* view on six datasets, the main reason is that MRMA adopts LRMA with multiple ranks. Moreover, Local LRMA methods (DFC, LLORMA, bACCAMS, WEMAREC) obtain competitive performance than Global LRMA in most case. It indicates that Local LRMA can capture the local structure of the large-scale rating matrix. Global-local LRMA (SMA, MPMA, GLOMA) is superior to Global LRMA and Local LRMA, which demonstrates considering both local and global information is more efficient.

The main reason that **ALoMA** achieves the best performance is that **ALoMA** considers both global and local rating information. Even though SMA, MPMA and GLOMA also take advantage of both global and local information, they set the same rank for all submtrices, which results in worse performance than **ALoMA**. For the recently published MRMA, it approximates the rating matrix with a mixture of global LRMA with different ranks, but the predefined rank set may not proper for the real data. Fortunately, the proposed **ALoMA** and **ALoMA-s** have ability to adaptively determine the optimal rank for each submatrix, that is why **ALoMA** and **ALoMA-s** outperform MRMA. Compared with **ALoMA-s**, **ALoMA** is superior to it since **ALoMA** considers not only adaptive submatrix generation and adaptive latent factors determination, but also non random missing mechanisms and global information.

Recommendation for *Near-cold-start Users* is a challenging problem. However, such kind of users are ubiquitous due to the sparsity of rating data. For example, the average number of *Near-cold-start Users* (with less than five ratings) in *Epinions*, *Douban*, *Dianping*, *Yelp*, *Movielens (10M)* and *Netflix* are 34310, 18943, 18194, 7439651, 1902, 2201 respectively, and they are about 69.6%, 14.6%, 12.3%, 78.4%, 2.7%, 0.46% of all users in the corresponding datasets. Since *Movielens (10M)* and *Netflix* are

Table 5. Comparing different recommendation methods on testing *All Users* for rating prediction task.

Datasets	*Epinions*		*Douban*		*Dianping*		*Yelp*		*Movielens (10M)*		*Netflix*	
Metrics	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
PMF	1.1206	1.3654	0.6230	0.7699	0.8641	1.0631	0.6684	0.8557	0.6431	0.8247	0.6385	0.8516
BPMF	1.1042	1.3471	0.6121	0.7543	0.8431	1.0378	0.6676	0.8431	0.6322	0.8217	0.6222	0.8435
MRMA	1.0241	1.2791	0.5723	0.7264	0.7415	0.9291	0.6297	0.8214	0.5937	0.7647	0.5863	0.7996
MF-MNAR	1.0743	1.3153	0.6073	0.7504	0.7921	0.9851	0.6631	0.8402	0.6123	0.7912	0.6141	0.8193
GPMF	1.0672	1.3116	0.6051	0.7481	0.7833	0.9769	0.6615	0.8389	0.6089	0.7983	0.6115	0.8162
DFC	1.0852	1.3239	0.6024	0.7472	0.7663	0.9462	0.6842	0.8731	0.6257	0.8064	0.6389	0.8451
LLORMA	1.0531	1.3033	0.5937	0.7396	0.7562	0.9351	0.6673	0.8453	0.6085	0.7862	0.6258	0.8296
bACCAMS	1.0432	1.2928	0.5841	0.7302	0.7435	0.9316	0.6649	0.8421	0.6242	0.8033	0.6301	0.8331
WEMAREC	1.0398	1.2872	0.5894	0.7356	0.7522	0.9328	0.6468	0.8356	0.6032	0.7772	0.6043	0.8143
SMA	1.0252	1.2799	0.5852	0.7331	0.7421	0.9302	0.6349	0.8278	0.5988	0.7695	0.5989	0.8058
MPMA	1.0211	1.2775	0.5831	0.7315	0.7443	0.9328	0.6322	0.8257	0.5971	0.7683	0.5953	0.8021
GLOMA	1.0181	1.2742	0.5826	0.7301	0.7406	0.9279	0.6315	0.8242	0.5964	0.7672	0.5931	0.8001
ALoMA-s	1.0154*	1.2693*	0.5684*	0.7213*	0.7372*	0.9258*	0.6254*	0.8182*	0.5885*	0.7594*	0.5796*	0.7944*
ALoMA	**1.0124**	**1.2613**	**0.5645**	**0.7123**	**0.7346**	**0.9213**	**0.6214**	**0.8137**	**0.5842**	**0.7547**	**0.5758**	**0.7873**

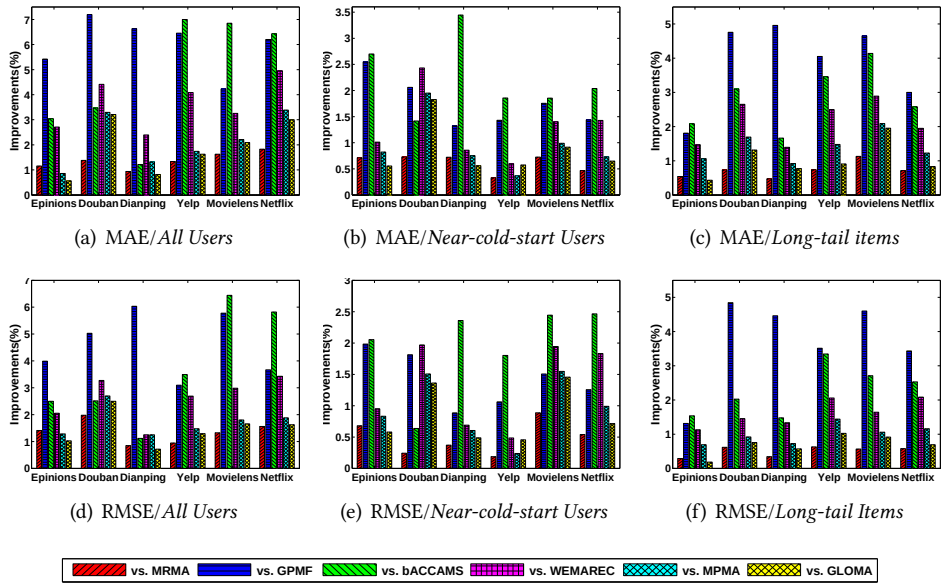
Table 6. Comparing different recommendation methods on testing *Near-cold-start Users* for rating prediction task.

Datasets	*Epinions*		*Douban*		*Dianping*		*Yelp*		*Movielens (10M)*		*Netflix*	
Metrics	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
PMF	1.3508	1.4477	0.8433	1.0239	1.1024	1.3742	0.8752	1.0722	0.8933	1.0946	0.9411	1.1941
BPMF	1.3412	1.4398	0.8246	1.0125	1.0942	1.3681	0.8623	1.0652	0.8852	1.0883	0.9341	1.1872
MRMA	1.2731	1.3915	0.7332	0.9126	1.0589	1.2974	0.8211	1.0124	0.8042	0.9911	0.8381	1.0973
MF-MNAR	1.2997	1.4113	0.7341	0.9132	1.0775	1.3161	0.8510	1.0519	0.8331	1.0292	0.8671	1.1225
GPMF	1.2963	1.4095	0.7429	0.9269	1.0652	1.3040	0.8301	1.0212	0.8124	0.9972	0.8462	1.1051
DFC	1.3225	1.4212	0.7561	0.9392	1.0952	1.3692	0.8531	1.0564	0.8412	1.0351	0.8952	1.1523
LLORMA	1.3041	1.4152	0.7492	0.9341	1.0912	1.3673	0.8421	1.0455	0.8212	1.0131	0.8712	1.1241
bACCAMS	1.2982	1.4105	0.7382	0.9162	1.0875	1.3231	0.8336	1.0287	0.8132	1.0064	0.8512	1.1183
WEMAREC	1.2769	1.3952	0.7456	0.9283	1.0603	1.3015	0.8233	1.0154	0.8096	1.0015	0.8461	1.1114
SMA	1.2757	1.3944	0.7441	0.9265	1.0644	1.3046	0.8224	1.0147	0.8088	0.9996	0.8415	1.1031
MPMA	1.2745	1.3936	0.7421	0.9241	1.0592	1.3004	0.8214	1.0129	0.8063	0.9976	0.8403	1.1022
GLOMA	1.2711	1.3901	0.7412	0.9228	1.0572	1.2989	0.8231	1.0151	0.8057	0.9967	0.8396	1.0992
ALoMA-s	1.2677*	1.2855*	0.7311*	0.9113*	1.0551*	1.2948*	0.8199*	1.0109*	0.8015*	0.9874*	0.8366*	1.0958*
ALoMA	**1.2641**	**1.3821**	**0.7279**	**0.9104**	**1.0513**	**1.2926**	**0.8184**	**1.0105**	**0.7984**	**0.9824**	**0.8342**	**1.0914**

Table 7. Comparing different recommendation methods on testing *Long-tail Items* for rating prediction task.

Datasets	*Epinions*		*Douban*		*Dianping*		*Yelp*		*Movielens (10M)*		*Netflix*	
Metrics	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
PMF	1.3295	1.5297	0.9367	1.0822	1.2133	1.4531	0.9734	1.1855	0.9346	1.1422	1.0531	1.2988
BPMF	1.3157	1.5154	0.9255	1.0767	1.2031	1.4453	0.9671	1.1785	0.9378	1.1387	1.0453	1.2894
MRMA	1.2795	1.4673	0.8513	0.9984	1.0814	1.3166	0.8998	1.1194	0.8328	1.0465	0.9603	1.1806
MF-MNAR	1.3071	1.4942	0.9041	1.0521	1.1523	1.3981	0.9345	1.1562	0.8823	1.0921	0.9983	1.2216
GPMF	1.2956	1.4823	0.9022	1.0503	1.1512	1.3969	0.9294	1.1515	0.8783	1.0885	0.9821	1.2141
DFC	1.3211	1.5206	0.9123	1.0682	1.1823	1.4267	0.9511	1.1682	0.9088	1.1124	1.0321	1.2766
LLORMA	1.3042	1.4921	0.8872	1.0215	1.1024	1.3383	0.9411	1.1612	0.8612	1.0722	0.9871	1.2185
bACCAMS	1.2991	1.4856	0.8713	1.0124	1.0942	1.3315	0.9241	1.1496	0.8576	1.0688	0.9781	1.2035
WEMAREC	1.2913	1.4796	0.8675	1.0067	1.0913	1.3296	0.9155	1.1353	0.8473	1.0577	0.9721	1.1982
SMA	1.2894	1.4769	0.8631	1.0054	1.0884	1.3234	0.9084	1.1316	0.8415	1.0532	0.9687	1.1915
MPMA	1.2861	1.4732	0.8594	1.0014	1.0862	1.3216	0.9064	1.1284	0.8407	1.0516	0.9652	1.1874
GLOMA	1.2781	1.4658	0.8562	0.9998	1.0846	1.3196	0.9013	1.1238	0.8396	1.0501	0.9614	1.1819
ALoMA-s	1.2755*	1.4640*	0.8488*	0.9953*	1.0793*	1.3146*	0.8967*	1.1164*	0.8285*	1.0437*	0.9583*	1.1786*
ALoMA	**1.2726**	**1.4631**	**0.8451**	**0.9923**	**1.0763**	**1.3121**	**0.8932**	**1.1124**	**0.8235**	**1.0406**	**0.9535**	**1.1738**

preprocessed by the publisher, the rating density is much larger than the other four datasets. As shown in Table 6, the proposed **ALoMA** and **ALoMA-s** consistently and significantly outperform the baselines. All Global LRMA methods with non-random missing data (MF-MNAR, GPMF) are superior to PMF and BPMF in *Near-cold-start Users* view, which indicates that non-random missing data are helpful to provide accurate recommendation on *Near-cold-start Users*. Notably, **ALoMA**



(a) MAE/*All Users* (b) MAE/*Near-cold-start Users* (c) MAE/*Long-tail items*

(d) RMSE/*All Users* (e) RMSE/*Near-cold-start Users* (f) RMSE/*Long-tail Items*

Fig. 5. The relative improvements of **ALoMA** vs. six baselines on six datasets in terms of (a) MAE/*All Users*, (b) MAE/*Near-cold-start Users*, (c) MAE/*Long-tail Items*, (d) RMSE/*All Users*, (e)RMSE/*Near-cold-start Users* and (f) RMSE/*Long-tail Items*.

outperforms all of baselines since **ALoMA** simultaneoulsy considers the local structures, global information and missing mechanisms.

In real applications, such as e-commercial platform, only a small part of items are popular, while a large fraction of items have few ratings, which is called as long-tail phenomenon [2]. Although the amount of users relating to each individual tail item is small in absolute numbers, they cover a substantial fraction of all users. Additionally, one user's rarer purchases in e-commerce are also more informative of their tastes than their purchases of popular items. Hence, making use of the tail items is important to predict the user preference in modern recommendation systems. In other words, it will be valuable if the recommendation systems could recommend the *Long-tail Items* to the proper users.

In order to investigate how the proposed **ALoMA** model handles the *Long-tail Items*, all items are ranked according to their rating frequency in an ascending order, and then the top 20% items are selected as the *Long-tail Items*. Table 7 shows recommendation performance of **ALoMA**, **ALoMA-s** and twelve baselines on the *Long-tail Items* in six datasets. It is exciting that the proposed model **ALoMA** is the best one and **ALoMA-s** is the second best one in all cases. This experimental result demonstrates that appropriate data assumption including global/local structure and missing mechanisms, are beneficial to mine the latent user/item factors, accurately approximate the incomplete rating matrix, and improve the recommendation quality no matter the items have sufficient ratings or not.

For demonstrating the efficiency of the proposed **ALoMA** method intuitively, we calculate and record improvements between **ALoMA** and six baselines (the first two better baselines from each category). Fig. 5 gives the relative improvements that **ALoMA** achieves related to six baselines on six datasets. Although the percentage of relative improvements are small, small improvements can lead to significant differences of recommendations in practice [16]. More specifically, we conduct

Table 8. Comparing different recommendation methods on testing *All Users* for ranking estimation task in *Yelp* dataset.

Datasets	*Yelp*					
Metrics	Re@5	Re@10	Pre@5	Pre@10	NDCG@5	NDCG@10
PMF	0.4522	0.4749	0.5621	0.6122	0.6877	0.7123
BPMF	0.4653	0.4861	0.5684	0.6185	0.6893	0.7162
MRMA	0.5085	0.5425	0.6013	0.6402	0.7025	0.7406
MF-MNAR	0.4796	0.5185	0.5785	0.7264	0.6923	0.7306
GPMF	0.4837	0.5214	0.5799	0.7283	0.6937	0.7324
DFC	0.4633	0.5073	0.5711	0.6203	0.6895	0.7173
LLORMA	0.4728	0.5104	0.5731	0.6217	0.6905	0.7283
bACCAMS	0.4882	0.5221	0.5846	0.6235	0.6913	0.7295
WEMAREC	0.4912	0.5269	0.5913	0.6302	0.6932	0.7311
SMA	0.4934	0.5298	0.5942	0.6336	0.6948	0.7326
MPMA	0.5039	0.5381	0.5966	0.6357	0.6972	0.7358
GLOMA	0.5066	0.5403	0.5983	0.6374	0.7002	0.7385
ALoMA-s	0.5112*	0.5458*	0.6025*	0.6413*	0.7039*	0.7415*
ALoMA	**0.5133**	**0.5473**	**0.6044**	**0.6421**	**0.7063**	**0.7438**

Table 9. Comparing different recommendation methods on testing *All Users* for ranking estimation task in *Movielens (10M)* dataset.

Datasets	*Movielens (10M)*					
Metrics	Re@5	Re@10	Pre@5	Pre@10	NDCG@5	NDCG@10
PMF	0.4732	0.4823	0.5652	0.5693	0.6723	0.7352
BPMF	0.4853	0.4894	0.5707	0.5715	0.6845	0.7467
MRMA	0.7085	0.7178	0.7302	0.7506	0.7219	0.7743
MF-MNAR	0.5341	0.5433	0.6631	0.6725	0.6953	0.7538
GPMF	0.5496	0.5589	0.6731	0.6796	0.7012	0.7593
DFC	0.5011	0.5128	0.6233	0.6321	0.6871	0.7493
LLORMA	0.6734	0.6821	0.7231	0.7345	0.7066	0.7632
bACCAMS	0.6839	0.6974	0.7241	0.7375	0.7094	0.7678
WEMAREC	0.6922	0.7005	0.7256	0.7423	0.7176	0.7703
SMA	0.7023	0.7123	0.7294	0.7497	0.7134	0.7679
MPMA	0.7032	0.7153	0.7298	0.7502	0.7214	0.7712
GLOMA	0.7052	0.7164	0.7316	0.7514	0.7231	0.7742
ALoMA-s	0.7103*	0.7246*	0.7315*	0.7511*	0.7306*	0.7762*
ALoMA	**0.7133**	**0.7273**	**0.7345**	**0.7531**	**0.7321**	**0.7796**

paired t-test (confidence 0.95) between **ALoMA** and each baseline with five-fold cross-validation results. The p-values in all cases are less than 10^{-5}, which indicates that our improvements are statistically significant at the 5% level. Therefore, based on these observations, we can say **ALoMA** consistently outperforms the state-of-the-art recommendation methods and significantly improves the recommendation performance.

Except for the evaluation on testing rating prediction, we investigate the performance of ranking prediction. Table 8, 9, 10 and 11 list the Recall, Precision, and NDCG of **ALoMA** and **ALoMA-s** by comparing twelve baselines on ranking estimation for two representative datasets *Yelp* and *Movielens (10M)* in two views (*All Users* and *Near-cold-start Users*). In both cases, **ALoMA** performs better than others. Notably, in terms of NDCG, all methods perform better in the case of *All Users* than in *Near-cold-start Users* on both datasets. The main reasons are two-fold. On one hand, it is more difficult for algorithms to correctly rank a big set than a small one under the NDCG metric. On the other hand, *Near-cold-start Users* usually occupy much less observed ratings in testing set

Table 10. Comparing different recommendation methods on testing *Near-cold-start Users* for ranking estimation task in *Yelp* dataset.

Datasets	*Yelp*					
Metrics	Re@5	Re@10	Pre@5	Pre@10	NDCG@5	NDCG@10
PMF	0.2212	0.2441	0.2581	0.2611	0.6552	0.6771
BPMF	0.2305	0.2437	0.2624	0.2655	0.6593	0.6814
MRMA	0.2815	0.2963	0.2917	0.3056	0.6803	0.7059
MF-MNAR	0.2479	0.2573	0.2711	0.2756	0.6657	0.6872
GPMF	0.2498	0.2621	0.2756	0.2794	0.6684	0.6896
DFC	0.2521	0.2689	0.2671	0.2803	0.6613	0.6892
LLORMA	0.2655	0.2813	0.2793	0.2914	0.6689	0.6926
bACCAMS	0.2689	0.2847	0.2823	0.2944	0.6703	0.6948
WEMAREC	0.2711	0.2864	0.2857	0.2972	0.6715	0.6969
SMA	0.2748	0.2895	0.2866	0.2989	0.6754	0.7003
MPMA	0.2764	0.2912	0.2884	0.3012	0.6784	0.7021
GLOMA	0.2795	0.2944	0.2904	0.3039	0.6812	0.7074
ALoMA-s	0.2822*	0.2982*	0.2928*	0.3069*	0.6839*	0.7088*
ALoMA	**0.2836**	**0.2994**	**0.2941**	**0.3098**	**0.6845**	**0.7103**

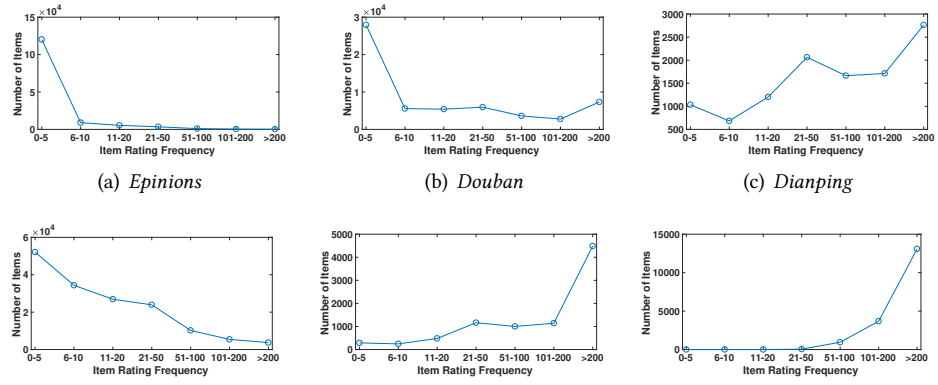
Table 11. Comparing different recommendation methods on testing *Near-cold-start Users* for ranking estimation task in *Movielens (10M)* dataset.

Datasets	*Movielens (10M)*					
Metrics	Re@5	Re@10	Pre@5	Pre@10	NDCG@5	NDCG@10
PMF	0.3132	0.3223	0.4732	0.4789	0.6323	0.6952
BPMF	0.3298	0.3285	0.4796	0.4854	0.6358	0.6994
MRMA	0.4812	0.4801	0.5396	0.5402	0.6594	0.7303
MF-MNAR	0.3684	0.3622	0.4974	0.4996	0.6456	0.7163
GPMF	0.3721	0.3673	0.5034	0.5051	0.6483	0.7192
DFC	0.3588	0.3532	0.4988	0.5006	0.6432	0.7124
LLORMA	0.4593	0.4567	0.5152	0.5189	0.6488	0.7201
bACCAMS	0.4656	0.4613	0.5216	0.5230	0.6503	0.7225
WEMAREC	0.4703	0.4675	0.5267	0.5281	0.6524	0.7258
SMA	0.4757	0.4736	0.5302	0.5311	0.6558	0.7271
MPMA	0.4782	0.4762	0.5352	0.5359	0.6572	0.7295
GLOMA	0.4801	0.4789	0.5374	0.5385	0.6598	0.7311
ALoMA-s	0.4819*	0.4813*	0.5403*	0.5421*	0.6618*	0.7329*
ALoMA	**0.4833**	**0.4821**	**0.5421**	**0.5431**	**0.6631**	**0.7343**

and thereby the item sets to be ranked are very small. Those experiments demonstrate that **ALoMA** can not only provide accurate rating prediction but also correctly achieve item ranking for each user. Similar results are obtained for other four datasets.

5.3.3 Effect of Item Rating Frequency. Sparsity is a challenge problem in recommendation system (e.g., the density of rating matrix in *Epinions*, *Douban*, *Dianping*, *Yelp*, *Movielens (10M)* and *Netflix* are only 0.01%, 0.222%, 0.13%, 0.00032%, 1.31% and 1.17% respectively). In order to investigate how the proposed **ALoMA** deals with such challenging data, we statistics the performance on items with different rating frequencies (i.e., the number of item's ratings). In experiments, the rating frequency is split into seven groups: 0-5, 6-10, 11-20, 21-50, 51-100, 101-200 and >200. The rating frequency distributions on different datasets are shown in Fig. 6. It can be seen that each dataset has its own characteristics on different item groups. For *Epinions*, *Douban* and *Yelp*, the number of items decreases with the increasing of rating frequency, which confirms that the rating matrix is sparse because most items only have few ratings. For *Dianping*, *Movielens (10M)* and *Netflix*, the

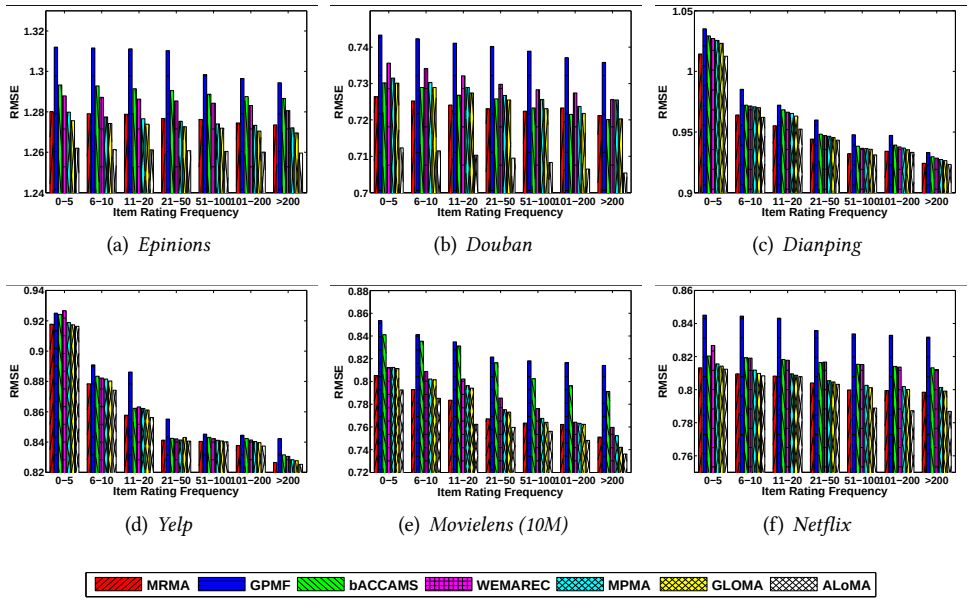
number of items increases with the increasing of rating frequency, thus, they are relatively denser than other three datasets.



(a) *Epinions* (b) *Douban* (c) *Dianping*

(d) *Yelp* (e) *Movielens (10M)* (f) *Netflix*

Fig. 6. The item rating frequency distribution in (a) *Epinions*, (b) *Douban*, (c) *Dianping*, (d) *Yelp*, (e) *Movielens (10M)* and (f) *Netflix*.

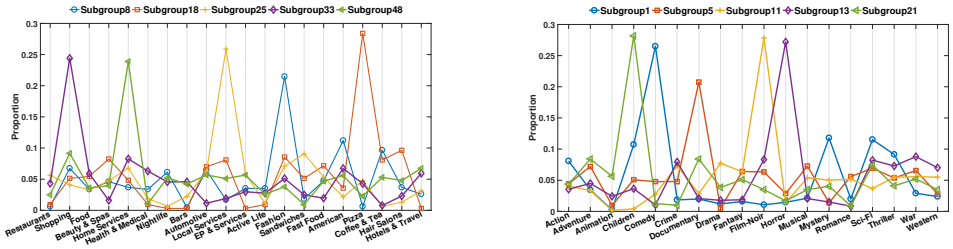


(a) *Epinions* (b) *Douban* (c) *Dianping*

(d) *Yelp* (e) *Movielens (10M)* (f) *Netflix*

Fig. 7. Comparisons of six baselines and the proposed **ALoMA** on all items with difference rating degrees for (a) *Epinions*, (b) *Douban*, (c) *Dianping*, (d) *Yelp*, (e) *Movielens (10M)* and (f) *Netflix*.

The recommendation performance on each rating frequency group of six datasets are shown in Fig. 7 (in terms of RMSE). All methods have the similar trends with respect to different item rating frequency, i.e., RMSE becomes better and better with the increasing of item rating frequency, which indicates that item rating frequency plays an important role in recommendation performance. From Fig. 6, it can be seen that items in each dataset have a significantly varying number of ratings. In



(a) Item category distribution on *Yelp* (b) Item category distribution on *Movielens (10M)*

Fig. 8. The category distribution in five example submatrices obtained by **ALoMA** on *Yelp* and *Movielens (10M)*.

this case, it is necessary to treat users/items differently. As expected, the proposed **ALoMA** outputs the highest quality for all rating frequency groups on six datasets. This result further demonstrates that **ALoMA** has the ability to effectively handle the recommendation data with complex structure.

5.3.4 Explainable Submatrix. One of the main contributions of **ALoMA** is to determine user-item co-clusters (i.e., submatrices) by sufficiently exploiting both rating and missing indicator information. In each submatrix, we expect that the users have the similar taste and the items has the similar characteristics. To confirm this, taking *Yelp* and *Movielens (10M)* datasets as examples, we investigate the corresponding semantical information of each submatrix. In *Yelp* and *Movielens (10M)*, each item (i.e., business on *Yelp* and movie on *Movielens*) is marked by one or more category labels, and all items belong to 1240 and 18 categories on *Yelp* and *Movielens (10M)* respectively. By counting the frequency that each category appears related to items, we selected top 20 popular categories in *Yelp* and all categories in *Movielens (10M)*. The category information is listed in Table 12.

Table 12. Top 20 popular categories in *Yelp* and all in *Movielens (10M)*.

Datasets	Item Categories				
Yelp	*Restaurants*	*Shopping*	*Food*	*Beauty&Spas*	*Home Services*
	Health&Medical	*Nightlife*	*Bars*	*Automotive*	*Local Services*
	EP&Services	*Active Life*	*Fashion*	*Sandwiches*	*Fast Food*
	American	*Pizza*	*Coffee&Tea*	*Hair Salons*	*Hotels&Travel*
Movielens (10M)	*Action*	*Adventure*	*Animation*	*Children*	*Comedy*
	Crime	*Documentary*	*Drama*	*Fantasy*	*Film-Noir*
	Horror	*Musical*	*Mystery*	*Romance*	*Sci-Fi*
	Thriller	*War*	*Western*		

For the p-th submatrix output by **ALoMA**, let $m^{(p)}$ be the number of items and $r_j^{(p)}$ indicate the number of rates on the j-th item ($1 \le j \le m^{(p)}$), where the number of items belonging to the q-th category is denoted as $m_q^{(p)}$ and $\sum_{q=1}^{n_l} m_q^{(p)} = m^{(p)}$. In this case, $\{m_q^{(p)}/m^{(p)}\}_{q=1}^{n_l}$ indicates the category distribution in the p-th submatrix, where n_l is the number of categories. The distribution of item categories in five submatrices on *Yelp* and *Movielens (10M)* are shown in Fig. 8. Obviously, each submatrix has its most related category, such as '*Fashion*', '*Pizza*', '*Local Services*', '*Shopping*', and '*Home Services*' for submatrix8, submatrix18, submatrix25, submatrix33, submatrix48 on *Yelp*, and '*Comedy*', '*Documentary*', '*Film-Noir*', '*Horror*' and '*Children*' respectively for submatrix1, submatrix5, submatrix11, submatrix13, submatrix21 on *Movielens (10M)*. Meanwhile, the top ten items (with largest $r_j^{(p)}$) of these five submatrices are listed in Table 13 and Table 14. As expected,

Table 13. Top ten items (Business) and their corresponding related categories in five example submatrices obtained by **ALoMA** from *Yelp* dataset.

Subgroup ID	Items (Business)			Category
Subgroup8	①Clothes Minded ④The Dredgers Union ⑦Walmart ⑩Broad Lingerie	②The Vault ⑤Old Navy ⑧A Second Look · · ·	③Marshalls ⑥Family Dollar ⑨Denim Kings & Nails	*Fashion*
Subgroup18	①Nello's Pizza Mesa ④Doughboys Pizza ⑦La Stellina ⑩Italian Oven	②Pizza Hut ⑤7 Star Thai Cuisine ⑧Casanova Brothers Pizza · · ·	③Pizzeria Amici ⑥Tavern on Park Restaurant ⑨China House	*Pizza*
Subgroup25	①Royal Mail ④Fixt Wireless Repair ⑦Saabr Sharpening Service ⑩Flair Cleaner	②Yorktown Shoe Repair ⑤Croach ⑧Public Storage · · ·	③ L & M Monogramming ⑥The Postal Route ⑨TigerDirect	*Local Services*
Subgroup33	①House 15143 ④Cloverdale Mall ⑦Plato's Closet Scarborough ⑩ Noffi Jewelry	②Mdi Rock ⑤ Freda's ⑧Walmart Supercenter · · ·	③REI ⑥Seefu Hair Yorkville ⑨Tan Phat Market	*Shopping*
Subgroup48	①Canyon Creek Village & Mirror ④House 15143 & Heating ⑦Lifetime Water Systems ⑩Regus	②DrainWorks ⑤Home EVER Inc. ⑧Favorite Maids · · ·	③Dhi Title ⑥Sos Exterminating ⑨Now Plumbing	*Home Service*

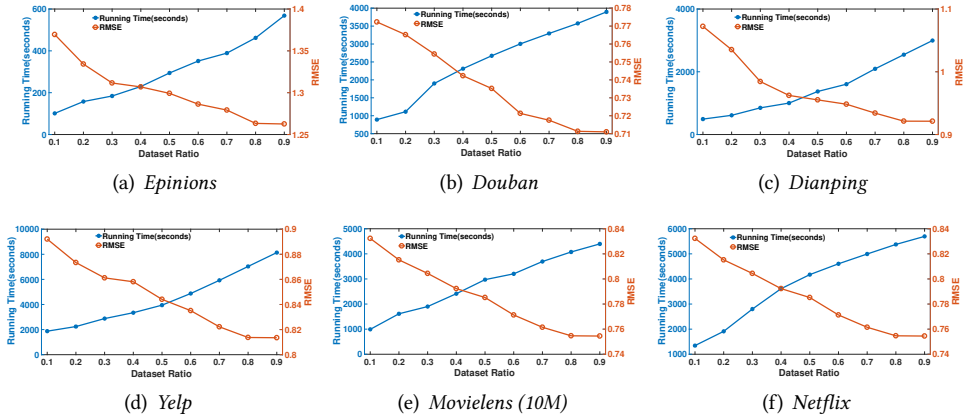
Table 14. Top ten items (Movies) and their corresponding related categories in five example submatrices obtained by **ALoMA** from *Movielens (10M)* dataset.

Subgroup ID	Items (Movie)			Category
Subgroup1	①Toy Story ④Forrest Gump ⑦Babe ⑩Ghost	②Fargo & Spa ⑤Back to the Future ⑧Pretty Woman · · ·	③Pulp Fiction ⑥Batman Forever ⑨Men in Black & Nails	*Comedy*
Subgroup5	①Hoop Dreams ④Crumb ⑦Buena Vista Social Club ⑩Looking for Richard	②Roger & Me ⑤Spellbound ⑧Trekkies · · ·	③Super Size Me ⑥When We Were Kings ⑨Stop Making Sense	*Documentary*
Subgroup11	①Blade Runner ④Third Man ⑦Big Sleep ⑩Blood Simple	②Sin City ⑤Strangers on a Train ⑧Mulholland Drive · · ·	③Dark City ⑥Sunset Blvd ⑨Notorious	*Film-Noir*
Subgroup13	①Alien ④Scream ⑦Jurassic Park 2 ⑩Alien3	②Aliens ⑤Seven ⑧Rocky Horror Picture Show · · ·	③Silence of the Lambs ⑥The Vampire Chronicles ⑨Mummy	*Horror*
Subgroup21	①Lion King ④Beauty and the Beast ⑦Wizard of Oz ⑩Home Alone	②Toy Story ⑤Who Framed Roger Rabbit? ⑧Monsters · · ·	③E.T. the Extra-Terrestrial ⑥Shrek ⑨Jumanji	*Children*

the items in each submatrix obviously have similar semantic characteristics and are related to the corresponding category. This result can further be used to interpret the recommendation results.

5.3.5 Scalability Analysis. We investigate the scalability of the **ALoMA** model. Note that our method is implemented in C++ at the hosts with Inter(R) Xeon(R) 2.0GHz CPU E7-4820 v2 having 64 processors, where each processor has two cores and the memory is 64GB. The operating system is Red Hat Enterprise Linux OS release 6.5. Meanwhile, the Inter(R) parallel studio XE 2017 composer edition for cpp is used to compile implemented code.

Specifically, we randomly select a subset of ratings as training set according to a fixed ratio (from 0.1 to 0.9 with step 0.1) and fix the testing set. For each ratio, ten subsets are extracted as training data and the averaged results (running time and RMSE) are recorded in Fig. 9. Obviously, the recommendation performance becomes better and better with the increasing of training data



(a) *Epinions* (b) *Douban* (c) *Dianping*

(d) *Yelp* (e) *Movielens (10M)* (f) *Netflix*

Fig. 9. Effect of training data size on **ALoMA** in terms of running time (seconds) and RMSE for (a) *Epinions*, (b) *Douban*, (c) *Dianping*, (d) *Yelp*, (e) *Movielens (10M)* and (f) *Netflix*.

size. Meanwhile, the training computational complexity (i.e., running time) is linearly scalable to the training data size.

6 CONCLUSIONS

In this paper, we propose a full Bayesian graphical model (**ALoMA**) to adaptively identify submatrices, determine the optimal rank for each submatrix, learn the submatrix-specific user/item latent factors and estimate the importance of latent feature with missing mechanisms. The generalization error bounds of **ALoMA** and its approximation guarantee are theoretically given. The experimental results on six real-world datasets have shown that **ALoMA** can effectively improve the recommendation accuracy in both rating prediction and ranking estimation tasks and friendly provide interpretable results.

In **ALoMA**, most variables are assumed following *Gaussian* distribution. However, this assumption may be violated in real applications, thus, it will be interesting to employ sparsity-favoring distributions such as *spike-and-slab Laplace* distribution for sparse recommendation data. Meanwhile, only rating information is considered here, which is more likely to suffer from cold-start problem. Thus, it is interesting to integrate the available and precious resources (such as social network, item contents, and user reviews) to design more effective and explainable recommendation method.

REFERENCES

[1] Michal Aharon, Michal Elad, and Alfred Bruckstein. 2006. SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing* 54, 11 (2006), 4311–4322.
[2] Chris Anderson. 2006. *The long tail: why the future of business is selling less of more.* Hyperion.
[3] Christophe Andrieu, Nando De Freitas, Arnaud Doucet, and Michael I Jordan. 2003. An introduction to mcmc for machine learning. *Machine Learning* 50, 1-2 (2003), 5–43.
[4] Alex Beutel, Amr Ahmed, and Alex Smola. 2015. ACCAMS: Additive co-clustering to approximate matrices succinctly. In *Proceedings of the International Conference on World Wide Web*. 119–129.
[5] Alex Beutel, Kenton Murray, Christos Faloutsos, and Alexander J Smola. 2014. Cobafi: collaborative bayesian filtering. In *Proceedings of the International Conference on World Wide Web*. ACM, 97–108.
[6] Bence Bolgar and Peter Antal. 2016. Bayesian matrix factorization with non-random missing data using informative gaussian process priors and soft evidences. In *Proceedings of the International Conference on Probabilistic Graphical Models*. 25–36.
[7] Emmanuel J Candes and Yaniv Plan. 2011. Matrix completion with noise. *Proc. IEEE* 98, 6 (2011), 925–936.

[8] Emmanuel J Candès and Benjamin Recht. 2009. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics* 9, 6 (2009), 717.

[9] Chao Chen, Dongsheng Li, Qin Lv, Junchi Yan, Stephen M Chu, and Li Shang. 2016. MPMA: Mixture probabilistic matrix approximation for collaborative filtering.. In *Proceedings of the International Joint Conference on Artificial Intelligent*. 1382–1388.

[10] Chao Chen, Dongsheng Li, Qin Lv, Junchi Yan, Li Shang, and Stephen M Chu. 2017. GLOMA: Embedding global information in local matrix approximation models for collaborative filtering.. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 1295–1301.

[11] Chao Chen, Dongsheng Li, Yingying Zhao, Qin Lv, and Li Shang. 2015. WEMAREC: Accurate and scalable recommendation through weighted and ensemble matrix approximation. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 303–312.

[12] Loc Do and Hady Wirawan Lauw. 2016. Probabilistic models for contextual agreement in preferences. *ACM Transation on Information System* 34, 4, Article 21 (June 2016), 33 pages.

[13] Thomas L Griffiths, Michael I Jordan, Joshua B Tenenbaum, and David M Blei. 2004. Hierarchical topic models and the nested chinese restaurant process. In *Proceedings of the International Conference on Neural Information Processing Systems*. 17–24.

[14] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. 2004. Evaluating collaborative filtering recommender systems. *ACM Transation on Information System* 22, 1 (Jan. 2004), 5–53.

[15] JosÂťe Miguel HernÂťandez-Lobato, Neil Houlsby, and Zoubin Ghahramani. 2014. Probabilistic matrix factorization with non-random missing data. In *Proceedings of the International Conference on Machine Learning*. 1512–1520.

[16] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[17] Joonseok Lee, Seungyeon Kim, Guy Lebanon, and Yoram Singer. 2013. Local low-rank matrix approximation. In *Proceedings of the International Conference on Machine Learning*. 82–90.

[18] Dongsheng Li, Chao Chen, Wei Liu, Tun Lu, Ning Gu, and Stephen Chu. 2017. Mixture-rank matrix approximation for collaborative filtering. In *Proceedings of the International Conference on Neural Information Processing Systems*. 477–485.

[19] Dongsheng Li, Chao Chen, Qin Lv, Junchi Yan, Li Shang, and Stephen Chu. 2016. Low-rank matrix approximation with stability. In *Proceedings of the International Conference on Machine Learning*. 295–303.

[20] Hui Li, Dingming Wu, Wweibing Tang, and Ninos Mamoulis. 2015. Overlapping community regularization for rating prediction in social recommender systems. In *Proceedings of ACM Conference on Recommender Systems*. 27–34.

[21] Roderick JA Little and Donald B Rubin. 2014. *Statistical analysis with missing data* (2nd ed.). Vol. 333. John Wiley & Sons.

[22] Hao Ma, Dengyong Zhou, Chao Liu, Michael R. Lyu, and Irwin King. 2011. Recommender systems with social regularization. In *Proceedings of ACM Conference on Web Search and Web Data Mining*. ACM, 287–296.

[23] Lester W Mackey, Michael I Jordan, and Ameet Talwalkar. 2011. Divide-and-conquer matrix factorization. In *Proceedings of the International Conference on Neural Information Processing Systems*. 1134–1142.

[24] Benjamin Marlin, Richard S Zemel, Sam Roweis, and Malcolm Slaney. 2012. Collaborative filtering and the missing at random assumption. *arXiv preprint arXiv:1206.5267* (2012).

[25] Benjamin M. Marlin and Richard S. Zemel. 2009. Collaborative prediction and ranking with non-random missing data. In *Proceedings of the ACM Conference on Recommender Systems*. 5–12.

[26] Paolo Massa and Paolo Avesani. 2007. Trust-aware recommender systems. In *Proceedings of ACM Conference on Recommender Systems*. 17–24.

[27] Radford M. Neal. 1996. *Bayesian learning for neural networks*. Springer. 456–456 pages.

[28] Shohei Ohsawa, Yachiko Obara, and Takayuki Osogami. 2016. Gated probabilistic matrix factorization: learning users' attention from missing values. In *Proceedings of the International Joint Conference on Artificial Intelligence*. AAAI Press, 1888–1894.

[29] Ruslan Salakhutdinov and Andriy Mnih. 2007. Probabilistic matrix factorization. In *Proceedings of International Conference on Machine Learning*. 880–887.

[30] Ruslan Salakhutdinov and Andriy Mnih. 2008. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the International Conference on Machine Learning*. 880–887.

[31] Lei Shi, Wayne Xin Zhao, and Yi-Dong Shen. 2017. Local representative-based matrix factorization for cold-start recommendation. *ACM Transation on Information System* 36, 2, Article 22 (Aug. 2017), 28 pages.

[32] Ruoyu Sun and Zhi-Quan Luo. 2016. Guaranteed matrix completion via non-convex factorization. *IEEE Transactions on Information Theory* 62, 11 (2016), 6535–6579.

[33] Keqiang Wang, Wayne Xin Zhao, Hongwei Peng, and Xiaoling Wang. 2016. Bayesian probabilistic multi-topic matrix factorization for rating prediction.. In *Proceedings of the International Joint Conference on Artificial Intelligent*. 3910–3916.

[34] David P Wipf and Bhaskar D Rao. 2007. An empirical bayesian strategy for solving the simultaneous sparse approximation problem. *IEEE Transation on Signal Processing* 55, 7 (2007), 3704–3716.
[35] Yao Wu, Xudong Liu, Min Xie, Martin Ester, and Qing Yang. 2016. CCCF: Improving collaborative filtering via scalable user-item co-clustering. In *Proceedings of the International Conference on Web Search and Data Mining*. ACM, 73–82.
[36] Menghao Zhang, Binbin Hu, Chuan Shi, and Bai Wang. 2017. Local low-rank matrix approximation with preference selection of anchor points. In *Proceedings of the International Conference on World Wide Web*. 1395–1403.
[37] Yongfeng Zhang, Min Zhang, Yiqun Liu, and Shaoping Ma. 2013. Improve collaborative filtering through bordered block diagonal form matrices. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 313–322.

Received Aug. 2018; revised * 20**; accepted * 20**