

## MULTILAYER TENSOR FACTORIZATION WITH APPLICATIONS TO RECOMMENDER SYSTEMS<sup>1</sup>

BY XUAN BI<sup>\*</sup>, ANNIE QU<sup>†,‡</sup> AND XIAOTONG SHEN<sup>§</sup>

*Yale University,<sup>\*</sup> Yunnan University,<sup>†</sup> University of Illinois at  
Urbana-Champaign<sup>‡</sup> and University of Minnesota<sup>§</sup>*

Recommender systems have been widely adopted by electronic commerce and entertainment industries for individualized prediction and recommendation, which benefit consumers and improve business intelligence. In this article, we propose an innovative method, namely the recommendation engine of multilayers (REM), for tensor recommender systems. The proposed method utilizes the structure of a tensor response to integrate information from multiple modes, and creates an additional layer of nested latent factors to accommodate between-subjects dependency. One major advantage is that the proposed method is able to address the “cold-start” issue in the absence of information from new customers, new products or new contexts. Specifically, it provides more effective recommendations through sub-group information. To achieve scalable computation, we develop a new algorithm for the proposed method, which incorporates a maximum block improvement strategy into the cyclic blockwise-coordinate-descent algorithm. In theory, we investigate algorithmic properties for convergence from an arbitrary initial point and local convergence, along with the asymptotic consistency of estimated parameters. Finally, the proposed method is applied in simulations and IRI marketing data with 116 million observations of product sales. Numerical studies demonstrate that the proposed method outperforms existing competitors in the literature.

**1. Introduction.** Recommender systems have become very important in daily life due to high demand from the entertainment industry and business marketing which produce large amounts of data. In addition, applications of recommender systems have been greatly facilitated by the advancement of statistical and machine learning techniques. These applications include personalized marketing for internet users, merchandise recommendation for retail stores and even individualized gene therapies. Each application involves collecting a wide variety of information, and successful exploitation of such rich information leads to more accurate recommendations. However, this also imposes unprecedented challenges to tradi-

---

Received July 2017; revised September 2017.

<sup>1</sup>Supported in part by NSF Grants DMS-1415500, DMS-1712564, DMS-1721216, DMS-1415308, DMS-1613190 and DMS-1415482, and National Institute on Drug Abuse Grant R01 DA016750.

*MSC2010 subject classifications.* Primary 62M20; secondary 90C26, 68T05.

*Key words and phrases.* Cold-start problem, context-aware recommender system, maximum block improvement, nonconvex optimization, tensor completion.

tional methods due to the large size and complex structure of data. Therefore, more general and integrative recommender systems are urgently needed.

The tensor, also called multidimensional array, is well recognized as a powerful tool to represent complex and unstructured data [49]. It is applied in many areas such as signal processing, neuroimaging and psychometrics (e.g., [25, 26, 29]). In recommender systems, the tensor shows its flexibility to accommodate contextual information, and is also regarded as one of the most effective tools for developing context-aware recommender systems (CARS; [1, 2]). In addition to user and item information from traditional recommender systems, tensor-based recommender systems also take the effect of contextual variables into account, such as time, location, users' companions, stores' promotion strategies, other relevant variables or any combinations thereof. Hence, CARS are capable of utilizing more information and provide more accurate recommendations [7, 41, 43].

Nevertheless, applying the tensor effectively to CARS remains a challenging problem. In matrix recommender systems, the singular value decomposition (SVD) method provides the best low-rank approximation, and is known to be arguably the most effective single procedure [16, 30]. In contrast, the SVD for tensor has more than one definition, and neither of the tensor decompositions inherits all of the desirable properties of a matrix SVD. This imposes a great challenge to generalize matrix decomposition to the tensor framework [48].

One common approach to utilize the tensor structure is to apply latent factor models, where each user, item or context is assigned an individual latent factor to represent their characteristics quantitatively. Existing methods include, but are not limited to, the factor model with temporal dynamics [21] and the Bayesian probabilistic tensor factorization method [47], both of which treat time as a contextual variable. Furthermore, the factorization machine [31, 36, 37] models interactions of all possible pairs of variables, while the multiverse recommendation method applies Tucker decomposition [19]. Other existing methods for CARS include contextual pre- or post-filtering [28, 34].

However, several key issues have not been solved completely. The first is the "cold-start" problem, where available information is not sufficient to provide valid predictions for new users, items or contexts (in the rest of this paper, we use "subject" to denote a user, an item or a context in general). For instance, in latent factor modeling, a latent factor is not estimable if a subject is not available in the training set. The subject's utilities can only be predicted through the average information from other subjects, which may lead to low prediction accuracy. Several solutions have been proposed for the traditional matrix recommendation techniques, for example, imputing pseudo ratings [18], supplementing artificial users and items [35], incorporating content-boosted information [17, 30] or utilizing group information for new subjects [6]. Nevertheless, the "cold-start" problem under CARS is quite challenging and has not been well investigated. One reason is that in addition to

new users and new items, information on new contexts could be insufficiently collected as well. For example, viewers may have different movie-watching experiences with new friends or at a new theater [12], and stores' sales volumes may vary under a new promotion strategy.

Another critical issue involves solving the higher-order tensors (beyond the third-order) problem. Higher-order tensors are very useful, because we might be interested in integrating more than one contextual variable to fully utilize their subject-specific information. For example, when recommending a new destination to travelers, one has to consider several factors, for example, the timing and the cost of the trip, and the crowdedness of the destination, in addition to travelers' travel interests. While some existing methods provide a general methodology on high-order tensors (e.g., [19]), the implementation of higher-order tensors could be challenging. One obstacle is the high computational cost. Another obstacle is that higher-order tensors could result in higher missing rates due to fewer observations at each combination of contextual variables; and this could lead to nonconvergence in computing for some existing methods. One possible solution is to choose only one special contextual variable in the tensor, and treat the rest of the contextual variables as linear covariates (e.g., [21, 47]). However, this may lead to loss of information on subject-specific or group-wise interaction.

In this paper, we propose a novel tensor factorization method, namely the recommendation engine of multilayers (REM). Specifically, we assume a tensor structure where each mode corresponds to *user*, *item* or a *contextual variable*, and each element of the tensor represents a utility, such as a rating or sales volume. The novelty of our method is that we add another layer which categorizes users, items and contexts into the same subgroups if they share similar characteristics. We quantify subgroup effects as random effects [23, 45] through nested-factors modeling, in addition to latent factors in the tensor factorization. Theoretically, we demonstrate the algorithmic properties of the proposed method, which converges to a stationary point from an arbitrary initial point, and local convergence to a local minimum with a linear convergence rate. The estimated parameter achieves asymptotic consistency under the  $L_2$ -loss function and other more general circumstances.

The proposed tensor factorization method has two significant advantages. First, it solves the "cold-start" problem effectively. For a new subject, even if its latent factors are not available, the nested factors from the corresponding subgroups can provide a group-specific estimate, which is more accurate than the average of the available observations. This finding is also supported by simulation studies where the proposed method is more effective than competing tensor factorization methods when the proportion of new subjects is high.

Second, the proposed method is able to accommodate high-order tensors. The difficulty of applying high-order tensors is solved by the proposed nested factors which utilize group-wise information, and hence are more robust to a higher missing rate. In addition, we propose a new algorithm that incorporates the maximum block improvement [9] into the cyclic blockwise-coordinate-descent algorithm.

This avoids the direct operation of large-scale tensors, and makes the estimation of high-order tensors feasible. Furthermore, a parallel computing strategy is implemented to calculate latent factors and nested factors for each subject and subgroup, respectively, which provides scalable and efficient computation.

The rest of this paper is organized as follows. Section 2 provides the background of the tensor factorization and the framework for the context-aware recommender systems. Section 3 introduces the proposed method and algorithm. Theoretical properties are derived in Section 4. Section 5 presents simulation studies to validate the performance of the proposed method. In Section 6, we apply the proposed method to IRI marketing data. Section 7 concludes with a discussion.

**2. Preparations.**

2.1. *Notation and tensor background.* In this subsection, we provide the background of the tensor, and introduce the notation for the proposed method.

We define a  $d$ th order tensor  $\mathbf{Y} \in \mathbb{R}^{n_1 \times \dots \times n_d}$  as a  $d$ -dimensional array, where each order is also called a mode. In the rest of this article, bold capital letters denote tensors, capital letters denote matrices, bold small letters denote vectors and small letters denote scalars.

In contrast to the singular value decomposition for a matrix, the rank and the bases of a tensor cannot be obtained simultaneously. Two different decompositions are commonly adopted. One is the high-order singular value decomposition, which decomposes a tensor into a  $d$ th order core tensor associated with  $d$  orthonormal matrices. This decomposition provides a basis at each mode. However, the core tensor is usually nondiagonal, and the tensor rank is not estimable through high-order singular value decomposition. An alternative choice of decomposition is the Canonical Polyadic Decomposition (CPD), where a tensor is represented as a sum of  $r$  rank-1 tensors. That is,

$$(2.1) \quad \mathbf{Y} \approx \sum_{j=1}^r \mathbf{p}_{\cdot j}^1 \circ \mathbf{p}_{\cdot j}^2 \circ \dots \circ \mathbf{p}_{\cdot j}^d,$$

where  $\circ$  represents the vector outer product, and  $\mathbf{p}_{\cdot j}^k$ ,  $j = 1, \dots, r$ , are  $n_k$ -dimensional vectors corresponding to the  $k$ th mode. Here,  $r$  is called the rank of  $\mathbf{Y}$  if the number of terms  $r$  is minimal. Equivalently, for each element of  $\mathbf{Y}$ , we have

$$(2.2) \quad y_{i_1 i_2 \dots i_d} \approx \sum_{j=1}^r p_{i_1 j}^1 p_{i_2 j}^2 \dots p_{i_d j}^d.$$

In the context of recommender systems, the tensor decomposition technique is geared toward the interpretation of intrinsic data variation. Therefore, we employ the CPD because the rank of a tensor corresponding to the number of latent factors is more important than the orthonormality. For other properties of a tensor, see [20] for an extensive review.

2.2. *Context-aware recommender systems.* In this subsection, we briefly review the tensor application to recommender systems, namely context-aware recommender systems (CARS). See also [2] for a comprehensive review of CARS.

Consider a  $d$ th order tensor ( $d \geq 3$ ) with the first two modes corresponding to *user* and *item* and the other  $(d - 2)$  modes corresponding to *contextual variables*. Let  $n_k$  ( $k = 1, \dots, d$ ) be the number of subjects for the  $k$ th mode, that is,  $n_1$  is the number of users,  $n_2$  is the number of items and  $n_3, \dots, n_d$  are the number of contexts for the  $(d - 2)$  contextual variables, respectively. For the CPD representation in (2.1), let  $P^k = (\mathbf{p}_{\cdot 1}^k, \dots, \mathbf{p}_{\cdot r}^k)_{n_k \times r}$ , where each row of  $P^1$  or  $P^2$  represents the  $r$ -dimensional latent factors for each user or item, and rows of  $P^3, \dots, P^d$  are the latent factors for the contextual variables, respectively. In the rest of this article, we use  $\mathbf{p}_{i_k}^k$  to represent the  $i_k$ th row of  $P^k$ , in contrast to  $\mathbf{p}_{\cdot j}^k$  being the  $j$ th column of  $P^k$ . Each element of  $\mathbf{Y}$  is defined as a utility, for example, a rating, a purchase or a sales volume, and is estimated via (2.2) in CPD. Here, the orthonormality for the latent factors is not required, and each utility  $y_{i_1 i_2 \dots i_d}$  comprises user-, item- and context-specific information. In addition, the CPD requires estimation of only  $\sum_{k=1}^d n_k r$  parameters, which essentially performs a tensor dimension reduction procedure.

In many applications, we may have a nonnegative tensor  $\mathbf{Y}$ . Methods for nonnegative CPD are proposed to address this issue (e.g., [3, 10, 32, 46]), as such decompositions make results meaningful and interpretable. In recommender system problems, however, the direct interpretation of latent factors is less critical, as the relative scale or ranking is important for recommendation. Most importantly, improving prediction accuracy is the ultimate goal of a good recommender system for users. Therefore, nonnegativity is not required here. Technically, we can always standardize  $\mathbf{Y}$  prior to analysis. In addition, nonnegative CPDs usually entail nonnegative latent factors, which might restrict the parameter space to the nonnegative orthant. In the proposed framework, an unbounded parameter space allows more flexibility for extensive search, which may lead to a more satisfactory convergence result.

The most common loss function is the  $L_2$ -loss, which is computationally efficient. Theoretical properties of other loss functions are also considered (e.g., [42, 51]). Let  $\Omega = \{(i_1, i_2, \dots, i_d) : y_{i_1 i_2 \dots i_d} \text{ is observed}\}$  be a set of indices corresponding to observed utilities, and  $|\Omega|$  represent the sample size. Since the number of available observations for each subject might be smaller than the number of latent factors  $r$ , we adopt a penalty function using regularization. For example, when the  $L_2$ -penalty is applied, we have

$$L(P^1, \dots, P^d | \mathbf{Y}) = \sum_{(i_1, \dots, i_d) \in \Omega} (y_{i_1 \dots i_d} - \hat{y}_{i_1 \dots i_d})^2 + \lambda \sum_{k=1}^d \|P^k\|_F^2,$$

where  $\hat{y}_{i_1 \dots i_d} = \sum_{j=1}^r p_{i_1 j}^1 \cdots p_{i_d j}^d$  is the estimated utility provided by (2.2), and  $\|\cdot\|_F$  represents the Frobenius norm. Other regularization methods include, but

are not limited to, the trend filtering penalty when  $d = 3$  [47], and the  $L_0$ - and  $L_1$ -penalty for sparse low-rank pursuit when  $d = 2$  [51].

Algorithms for implementing the CPD include the cyclic coordinate descent algorithm, and the stochastic gradient descent method. Alternatively, [9] propose a maximum block improvement algorithm that only updates the block with the maximum improvement in each iteration instead of updating each block cyclically. This strategy guarantees convergence to a stationary point, and ensures a fast convergence rate in many circumstances.

### 3. A multilayer method.

3.1. *General methodology.* In this section, we develop the methodology for the proposed REM method. Specifically, we adopt the idea of nested factors from the design of experiments to capture between-subject dependency under the CPD framework (2.1).

We assume that subjects can be categorized into subgroups, where subjects within the same subgroup share similar characteristics and are dependent on each other. For subgrouping, we can incorporate prior information such as users' demographic information, item categories and functionality, and contextual similarity. If this kind of information is not available, one can utilize the missing pattern of the tensor data, or the number of records from each user and on each item [38]. As shown in [16], implicit information from the number of records may reflect subjects' behavior that is not available elsewhere, and can help improve recommendation accuracy. In more general situations, clustering methods such as the  $k$ -means can be applied to determine the subgroups. See [44] and [15] on robust approaches to select the number of subgroups.

Suppose the subgroup labels are given, then we formulate each utility as follows:

$$(3.1) \quad \hat{y}_{i_1 i_2 \dots i_d} = \sum_{j=1}^r (p_{i_1 j}^1 + q_{i_1 j}^1)(p_{i_2 j}^2 + q_{i_2 j}^2) \cdots (p_{i_d j}^d + q_{i_d j}^d),$$

where  $p_{i_k j}^k$  is the  $j$ th latent factor for the  $i_k$ th subject from the  $k$ th mode, and  $q_{i_k j}^k$  is the corresponding nested factor,  $j = 1, \dots, r$ ,  $i_k = 1, \dots, n_k$  and  $k = 1, \dots, d$ . We define the  $n_k \times r$ -dimensional matrix  $Q^k$  similar to  $P^k$  as in Section 2.2. Notice that we have  $\mathbf{q}_{i_k}^k = \mathbf{q}_{i'_k}^k$  if subjects  $i_k$  and  $i'_k$  are from the same subgroup. We assume that the number of subgroups for the  $k$ th mode is  $m_k$ , which corresponds to  $m_k$  unique values for  $q_{i_k j}^k$ . We use  $\mathbf{q}_{(u_k)}^k$  occasionally to denote the nested factor associated with the subgroup  $u_k$ ,  $u_k = 1, \dots, m_k$ .

Let  $P = ((P^1)', \dots, (P^d)')$  and  $Q = ((Q^1)', \dots, (Q^d)')$  represent all parameters of interest. We define  $L(P, Q|\mathbf{Y}) = L(P^1, \dots, P^d, Q^1, \dots, Q^d|\mathbf{Y})$  as the overall criterion function:

$$(3.2) \quad L(P, Q|\mathbf{Y}) = \sum_{(i_1, \dots, i_d) \in \Omega} (y_{i_1 \dots i_d} - \hat{y}_{i_1 \dots i_d})^2 + \lambda(\|P\|_F^2 + \|Q\|_F^2),$$

where  $\hat{y}_{i_1 \dots i_d}$  is represented by (3.1) and  $\lambda$  is the penalization coefficient. Here, we adopt the most commonly used  $L_2$ -loss and  $L_2$ -penalty for efficient computation, although other types of loss and penalty functions are also applicable. For example, one may consider the hinge loss or the  $\psi$ -loss [40] for classification, and the absolute loss or Huber loss to achieve robust estimation. Meanwhile, if prior knowledge of the latent and nested factors is available, for example, regarding sparsity or smoothness, then appropriate regularization methods can be applied.

3.2. *Parameter training.* In the following, we discuss how the model parameters are estimated. Mainly, we are interested in finding a solution  $(\hat{P}, \hat{Q})$  aiming at minimizing  $L(P, Q|\mathbf{Y})$ . Let  $\Omega_{i_k}^k = \{(i_1, \dots, i_k, \dots, i_d) : y_{i_1 \dots i_k \dots i_d} \text{ is observed}\}$  be the set of indices where the  $k$ th mode index equals  $i_k$  and the corresponding utilities are observed; namely  $|\Omega_{i_k}^k|$  denotes the number of observations for subject  $i_k$ . Let  $\mathcal{I}_{(u_k)}^k$  be the set of subjects in the subgroup  $u_k$ ,  $u_k = 1, \dots, m_k$ . We assume that  $|\mathcal{I}_{(u_k)}^k| \geq 2$  for each  $u_k$ .

For each mode of the tensor, the partial derivatives of  $L(\cdot|\mathbf{Y})$  have explicit forms with respect to the latent factors or the nested factors, which makes it feasible to apply the blockwise coordinate descent approach. That is,

$$(3.3) \quad \hat{\mathbf{p}}_{i_k}^k = \arg \min_{\mathbf{p}_{i_k}^k} \sum_{\Omega_{i_k}^k} (y_{i_1 \dots i_d} - \hat{y}_{i_1 \dots i_d})^2 + \lambda \|\mathbf{p}_{i_k}^k\|_2^2,$$

for  $i_k = 1, \dots, n_k$ , and

$$(3.4) \quad \hat{\mathbf{q}}_{(u_k)}^k = \arg \min_{\mathbf{q}_{(u_k)}^k} \sum_{i_k \in \mathcal{I}_{u_k}^k} \sum_{\Omega_{i_k}^k} (y_{i_1 \dots i_d} - \hat{y}_{i_1 \dots i_d})^2 + \lambda \|\mathbf{q}_{(u_k)}^k\|_2^2,$$

for  $u_k = 1, \dots, m_k$ , and  $k = 1, \dots, d$ .

The estimation procedure of  $\hat{\mathbf{p}}_{i_k}^k$  in (3.3) is a ridge regression, and does not require knowing  $\hat{\mathbf{p}}_{i'_k}^k$  for  $i'_k \neq i_k$ . Thus, parallel computation is applicable to calculate  $\hat{\mathbf{p}}_1^k, \dots, \hat{\mathbf{p}}_{n_k}^k$  efficiently. This strategy is also applicable to obtaining the  $\hat{\mathbf{q}}_{(u_k)}^k$ 's in (3.4). Therefore, the minimization of  $L(P, Q|\mathbf{Y})$  can be done cyclically through estimating  $P$  and  $Q$ .

Notice that  $\Omega = \bigcup_{i_k=1}^{n_k} \Omega_{i_k}^k$ , and it is possible that  $\Omega_{i_k}^k$  is empty for certain  $i_k$ 's, that is, there is no observation on subject  $i_k$ , as in the case of the ‘‘cold-start’’ problem. Under this circumstance, the latent factor of  $i_k$  is not estimable, and is assigned as  $\mathbf{p}_{i_k}^k = \mathbf{0}$ . The predicted values calculated by existing methods may degenerate to the grand mean or subjects’ main effects. In contrast, the proposed method utilizes the nested factor  $\mathbf{q}_{(u_k)}^k$ , which borrows information from members of the same subgroup. Thus, even for a new subject, the predicted values calculated by (3.1) retain information from other modes, and hence achieve better prediction accuracy.



3.3. *Algorithm.* In contrast to matrix factorization, a tensor decomposition usually entails high computational cost, and hence many algorithms feasible for a matrix may not be scalable for tensor decomposition. For example, it is nearly impossible to embed the back-fitting algorithm into the maximum block improvement (MBI) for tensor data. First, the number of parameters and sample size for a tensor decomposition could be much greater than its matrix counterpart, which makes the computation of the MBI more intensive. Second, since the number of blocks increases significantly, the MBI algorithm may never update certain blocks due to small improvements along these directions, which leads to the estimated values corresponding to these blocks remaining the same as the initial values. To solve these problems, we propose a two-step algorithm, which estimates the latent-factors matrix  $\hat{P}$  and the nested-factors matrix  $\hat{Q}$  iteratively. Within the estimation of each matrix, we apply the MBI algorithm to find the optimal block direction with the largest improvement of estimations.

Specifically, we propose the following algorithm aiming at minimizing (3.2). Let  $(P_s, Q_s)$  denote the estimated  $(P, Q)$  at the  $s$ th iteration, then the improvement of estimations for updating the  $k$ th mode is defined as

$$(3.5) \quad I_s^k = 1 - \frac{L(P_{s-1}^1, \dots, P_{s-1}^{k-1}, P^{k*}, P_{s-1}^{k+1}, \dots, P_{s-1}^d, Q_{s-1} | \mathbf{Y})}{L(P_{s-1}, Q_{s-1} | \mathbf{Y})},$$

and

$$(3.6) \quad J_s^k = 1 - \frac{L(P_{s-1}, Q_{s-1}^1, \dots, Q_{s-1}^{k-1}, Q^{k*}, Q_{s-1}^{k+1}, \dots, Q_{s-1}^d | \mathbf{Y})}{L(P_{s-1}, Q_{s-1} | \mathbf{Y})},$$

where  $P^{k*}$  and  $Q^{k*}$  are the attempted updates for the  $k$ th mode,  $k = 1, \dots, d$ .

One advantage of the proposed algorithm is that it requires small memory storage. Note that at each iteration, only one subject’s information is required to estimate  $\hat{\mathbf{p}}_{i_k}^k$  and one sub-group’s information is required to estimate  $\hat{\mathbf{q}}_{(u_k)}^k$ . Furthermore, the computational complexity of the proposed algorithm is no greater than  $2dn_{\text{iter}}c_{\text{ridge}}$ , where  $n_{\text{iter}}$  is the number of iterations and  $c_{\text{ridge}}$  is the complexity of the ridge regression. In addition, since the MBI algorithm does not update blocks cyclically, it is able to discover and utilize “shortcuts” in optimization, which may significantly reduce the number of iterations.

3.4. *Implementation.* In this subsection, we address several implementation issues. In general, we split the data into a 50% training set, a 25% validation set and a 25% testing set, randomly. The tuning parameter  $\lambda$  is selected to minimize the root mean square error (RMSE) on the validation set, where the RMSE on a set  $\Omega$  is defined as  $\sqrt{\frac{1}{|\Omega|} \sum_{\Omega} (y_{i_1 \dots i_d} - \hat{y}_{i_1 \dots i_d})^2}$ . To improve prediction, we could specify  $\lambda$  differently for each row of  $P$  and  $Q$ . In our numerical study, we use  $\lambda$  uniformly for the latent factors in  $P$ , but use  $\lambda_{(u_k)}^k = \lambda / |\mathcal{I}_{(u_k)}^k|$  for each subgroup  $u_k$  to penalize each parameter equally. Furthermore, we need to choose the number



of latent factors  $r$ . In general,  $r$  is no smaller than the theoretical rank of the tensor in order to represent subjects' characteristics sufficiently well. However, a large  $r$  may lead to intensive computation and possible nonconvergence of algorithms.

Moreover, it is important to determine which contextual variables should be integrated in the tensor. In Section 5.2, we demonstrate that if the true tensor is of high-order, then assuming a low-order tensor structure may lead to a loss of information. That is, it is important to include key contextual variables in the tensor. On the other hand, applying a high-order tensor to formulate a low-order problem is unnecessary and may entail extra computational cost. In practice, we assume that the order of a tensor can be judged from prior knowledge. We acknowledge that determining the order of a tensor remains an open problem.

**4. Theoretical properties.** This section develops theoretical properties for the proposed method. Our contributions are mainly on two aspects. One is on the convergence properties of the proposed algorithm, which converges to a stationary point from an arbitrary initial point, and local convergence to a local minimum (or a global minimum, if one exists) with linear convergence rate. The other contribution is on the statistical properties. We prove the asymptotic consistency of the estimated parameter under  $L_2$  loss and more general criterion functions.

One well-known critical issue is the discrepancy between the algorithmic and statistical properties (e.g., [6, 51]). In some existing works, the statistical framework may require the estimated parameter to be a global minimizer, which might not be existent or attainable.

Our theoretical development, nevertheless, bridges this gap from two aspects. First and foremost, we relax the strict condition such that a global minimizer is no longer required to establish statistical properties, as long as the criterion function converges to its infimum asymptotically. Second, we provide technical solutions on finding possible global minima or satisfactory local minima, with additional computational cost.

**4.1. Identifiability.** Identifiability is critical for tensor representations. For recommender systems, although having identifiable latent factors does not improve prediction accuracy, it could still be important for algorithmic convergence which may lead to favorable statistical properties. Here, we provide sufficient conditions to achieve identifiable latent factors prior to establishing theoretical properties.

In the proposed framework, unidentifiability is attributed to four aspects. The first three aspects are elementary indeterminacies of scaling, permutation and addition, whereas the last one is the so-called nonuniqueness of the CPD with more than one possible combination of rank-one tensors sum to  $\mathbf{Y}$  after controlling for the three elementary indeterminacies [20].

Let  $B = ((B^1)', \dots, (B^d)')'$  where  $B^k = P^k + Q^k$ ,  $k = 1, \dots, d$ . The scaling indeterminacy refers to nonuniqueness with respect to a scale change of each column vector of  $B^k$ . That is, for  $d$  diagonal scaling matrices  $\Gamma^k = \text{diag}(\gamma_1^k, \dots, \gamma_r^k)$ ,

$k = 1, \dots, d$ , we have  $\tilde{B} = ((B^1 \Gamma^1)', \dots, (B^d \Gamma^d)')$  such that  $\prod_{k=1}^d \gamma_j^k = 1$  for  $j = 1, \dots, r$ . The permutation indeterminacy comes from an arbitrary  $r \times r$  permutation matrix  $\Pi$ , such that  $\tilde{B} = ((B^1 \Pi)', \dots, (B^d \Pi)')$ . In addition to scaling and permutation, the proposed tensor representation may suffer from the addition indeterminacy, that is, for arbitrary  $n_k \times r$  matrix  $\Delta^k$ ,  $\tilde{P}^k = P^k + \Delta^k$  and  $\tilde{Q}^k = Q^k - \Delta^k$ .

As a special case of  $d = 2$ , the elementary indeterminacy reduces to the nonsingular transformation indeterminacy for a matrix. Specifically, for a  $r \times r$  nonsingular matrix  $\Upsilon$ ,  $\tilde{B}^1 = B^1 \Upsilon$  and  $\tilde{B}^2 = B^2 (\Upsilon^{-1})'$ . Nevertheless, this issue can always be solved by the singular value decomposition, which imposes orthonormality to column vectors. Therefore, we focus our attention on higher-order tensors with  $d \geq 3$ , although some of the results continue to hold for a matrix.

LEMMA 4.1. *Predicted values given by (3.1) are invariant with respect to scaling, permutation and addition indeterminacies.*

The proof is straightforward by applying the aforementioned definition of scaling, permutation and addition indeterminacies to (3.1), and is hence skipped. We introduce the concept of  $k$ -rank, which is the Kruskal rank introduced in [22]. Specifically, for a matrix  $A$ , the  $k$ -rank of  $A$  is

$$\mathcal{K}_A = \max\{k : \text{any } k \text{ columns of } A \text{ are linearly independent}\}.$$

PROPOSITION 4.1. *Suppose  $\sum_{k=1}^d \mathcal{K}_{B^k} \geq 2r + (d - 1)$ . Minimizers of  $L(P, Q|Y)$  in  $P$  and  $Q$  are unique up to permutation almost surely.*

The above condition  $\sum_{k=1}^d \mathcal{K}_{B^k} \geq 2r + (d - 1)$  is not strong. In numerical studies, factors in  $(P^k + Q^k)$  are usually linearly independent, and hence we have  $\mathcal{K}_{B^k} = r$ . Then this condition reduces to  $r \geq 1 + 1/(d - 2)$  for  $d \geq 3$ , which is achievable even for low-rank high-order tensors.

As shown in Proposition 4.1, the issue of scaling and addition indeterminacies is resolved almost surely through penalization imposed on  $L(P, Q|Y)$ . To treat the permutation indeterminacy, we rearrange  $r$  column vectors  $(\mathbf{p}_{\cdot 1}^k + \mathbf{q}_{\cdot 1}^k, \mathbf{p}_{\cdot 2}^k + \mathbf{q}_{\cdot 2}^k, \dots, \mathbf{p}_{\cdot r}^k + \mathbf{q}_{\cdot r}^k)$  for each mode such that

$$\sum_{k=1}^d \|\mathbf{p}_{\cdot 1}^k + \mathbf{q}_{\cdot 1}^k\|_2^2 \geq \sum_{k=1}^d \|\mathbf{p}_{\cdot 2}^k + \mathbf{q}_{\cdot 2}^k\|_2^2 \geq \dots \geq \sum_{k=1}^d \|\mathbf{p}_{\cdot r}^k + \mathbf{q}_{\cdot r}^k\|_2^2,$$

which is analogous to imposing a descending order of eigenvalues as in matrix decomposition. The rearrangement of column vectors can be implemented during or after the proposed algorithm, since it does not affect the estimation procedure. We acknowledge that the above choice is arbitrary. One may carry out other rearrangements. For example, [50] suggest imposing ordering based on the first element rather than the vector norm. Alternatively, one may utilize prior knowledge if available. In the rest of Section 4, we assume that parameter  $(P, Q)$  is identifiable.

4.2. *Algorithmic properties.*

4.2.1. *Convergence with an arbitrary initial point.* This subsection investigates the convergence property of the proposed algorithm in terms of  $(P, Q)$  from an arbitrary initial point. We first establish the property in a compact parameter space and then generalize it to an unbounded open parameter space with extra conditions.

Let  $\mathcal{D} \subset \mathbb{R}^{2r \sum_{k=1}^d n_k}$  be the parameter space of  $(P, Q)$ . Then  $(\tilde{P}, \tilde{Q}) \in \mathcal{D}$  is called a *blockwise local minimizer* of  $L(\cdot|\mathbf{Y})$  if

$$\tilde{P}^k = \arg \min_{P^k} L(\tilde{P}^1, \dots, \tilde{P}^{k-1}, P^k, \tilde{P}^{k+1}, \dots, \tilde{P}^d, \tilde{Q}^1, \dots, \tilde{Q}^d | \mathbf{Y}),$$

and

$$\tilde{Q}^k = \arg \min_{Q^k} L(\tilde{P}^1, \dots, \tilde{P}^d, \tilde{Q}^1, \dots, \tilde{Q}^{k-1}, Q^k, \tilde{Q}^{k+1}, \dots, \tilde{Q}^d | \mathbf{Y}),$$

which is equivalent to a local minimizer along each block direction. In the rest of Section 4.2, we consider the criterion function  $L(\cdot|\mathbf{Y})$  as defined in (3.2).

LEMMA 4.2. *Suppose  $\mathcal{D}$  is compact, and the iterates obtained from Algorithm 1 have a cluster point  $(\tilde{P}, \tilde{Q})$ . Then  $(\tilde{P}, \tilde{Q})$  is a blockwise local minimizer of the criterion function  $L(\cdot|\mathbf{Y})$ .*

---

**Algorithm 1** A two-step algorithm with parallel computing

---

1. (*Initialization*) Input all observed  $y_{i_1 \dots i_d}$ 's, the rank  $r$ , the tuning parameter  $\lambda$ , initial value  $(P_0, Q_0)$  and a stopping criterion  $\varepsilon = 10^{-4}$ .
2. (*Latent-factors update*) At the  $s$ th iteration ( $s \geq 1$ ), estimate  $P_s$ .
  - (i) For each  $P^k$ , solve (3.3) through parallel computing and obtain  $P^{k*} = (\hat{\mathbf{p}}_1^k, \dots, \hat{\mathbf{p}}_{n_k}^k)'$ . Calculate  $I_s^k$  through (3.5).
  - (ii) Assign  $P_s^{k_0} \leftarrow P^{k_0*}$ , if  $I_s^{k_0} = \max\{I_s^1, \dots, I_s^d\}$ .
3. (*Nested-factors update*) At the  $s$ th iteration ( $s \geq 1$ ), estimate  $Q_s$ .
  - (i) For each  $Q^k$ , solve (3.4) through parallel computing and obtain  $Q^{k*} = (\hat{\mathbf{q}}_1^k, \dots, \hat{\mathbf{q}}_{n_k}^k)'$ . Calculate  $J_s^k$  through (3.6).
  - (ii) Assign  $Q_s^{k_0} \leftarrow Q^{k_0*}$ , if  $J_s^{k_0} = \max\{J_s^1, \dots, J_s^d\}$ .
4. (*Stopping Criterion*) Stop if

$$\max\{I_s^1, \dots, I_s^d, J_s^1, \dots, J_s^d\} < \varepsilon.$$

Set  $(\hat{P}^1, \dots, \hat{P}^d, \hat{Q}^1, \dots, \hat{Q}^d) = (P_s^1, \dots, P_s^d, Q_s^1, \dots, Q_s^d)$ .

Otherwise set  $s \leftarrow s + 1$  and go to step 2.

---

In the following, we assume that  $\mathcal{D}$  is open. We consider  $L(\cdot|\mathbf{Y})$  at each of its block coordinates. Let

$$L^k(P^k) = L(P^k|\mathbf{Y}, P^{(-k)}, Q) \quad \text{and} \quad L^{k+d}(Q^k) = L(Q^k|\mathbf{Y}, P, Q^{(-k)})$$

be  $L(P, Q|\mathbf{Y})$  given  $(P^{(-k)}, Q)$  and  $(P, Q^{(-k)})$ , respectively, where  $X^{(-k)} = (X^1, \dots, X^{k-1}, X^{k+1}, \dots, X^d)$  for  $X = P$  or  $Q$ ,  $k = 1, \dots, d$ . Then Assumption 4.1 implies that the improvement on  $(P^k)$ 's or  $(Q^k)$ 's is not dominated by each other.

ASSUMPTION 4.1. *Let  $\{(P_s, Q_s)\}_{s \geq 1}$  be a sequence of estimated parameters generated by Algorithm 1, where  $s$  represents the  $s$ th iteration. Then*

$$O\left(\max_{k=1, \dots, d} \|\nabla L^k(P_s^k)\|_F\right) \sim O\left(\max_{k=1, \dots, d} \|\nabla L^{k+d}(Q_s^k)\|_F\right).$$

Furthermore, let  $H(L^k)$  be the Hessian matrix of  $L^k(\cdot)$ ,  $k = 1, \dots, 2d$ . We assume that  $\|H(L^k)\|_2 \leq \zeta^k$ , where the constant  $\zeta^k > 0$  is bounded above and may depend on all block coordinates of  $L(\cdot|\mathbf{Y})$  except the  $k$ th block. The following proposition leads to the convergence to a blockwise local minimizer.

PROPOSITION 4.2. *Suppose  $\mathcal{D}$  is open and Assumption 4.1 holds. Let  $\|H(L^k)\|_2 \leq \zeta^k$  for  $k = 1, \dots, 2d$ . Then the sequence  $\{(P_s, Q_s)\}_{s \geq 1}$  obtained from Algorithm 1 converges to a blockwise local minimizer of the criterion function  $L(P, Q|\mathbf{Y})$ .*

Notice that, since the parameter space is open, a blockwise local minimizer satisfies  $\nabla L = \mathbf{0}$ , and hence is a special case of a stationary point.

4.2.2. *Local convergence.* In this subsection, we provide the local convergence property of Algorithm 1. Specifically, we follow [27] and show that Algorithm 1 converges to a local minimum at the linear rate, provided that an initial value is sufficiently close to the local minimizer. Moreover, the same property applies to a global minimum if it exists.

Let  $(\tilde{P}, \tilde{Q})$  be a local minimizer of  $L(P, Q|\mathbf{Y})$ , and  $\tilde{H} = H(L(\tilde{P}, \tilde{Q}|\mathbf{Y}))$  be the Hessian matrix at  $(\tilde{P}, \tilde{Q})$ . We define the *energy norm* based on  $\tilde{H}$  as  $\|(P, Q)\|_E = (\text{vec}(P, Q), \tilde{H}\text{vec}(P, Q))^{1/2}$ .

PROPOSITION 4.3. *Suppose  $\mathcal{D}$  is open, and let  $(\tilde{P}, \tilde{Q}) \in \mathcal{D}$  be a strict local minimizer of  $L(P, Q|\mathbf{Y})$ . For a small neighborhood  $\mathcal{V}$  of  $(\tilde{P}, \tilde{Q})$ , suppose  $(P_{s_0}, Q_{s_0}) \in \mathcal{V}$  for some  $s_0 \geq 0$ . Then a sequence  $\{(P_s, Q_s)\}_{s \geq s_0} \subset \mathcal{V}$  obtained from Algorithm 1 exists, and converges to  $(\tilde{P}, \tilde{Q})$  at least linearly in the energy norm. That is, there exists  $\mu \in [0, 1)$ , such that*

$$\|(P_{s+1}, Q_{s+1}) - (\tilde{P}, \tilde{Q})\|_E \leq \mu \|(P_s, Q_s) - (\tilde{P}, \tilde{Q})\|_E.$$

Suppose the tolerance error of Algorithm 1 is  $\epsilon$ , and  $s_0 = 0$ . Then the number of iterations has an upper bound:

$$n_{\text{iter}} \sim O(\{\log \epsilon - \log \|(P_0, Q_0) - (\tilde{P}, \tilde{Q})\|_E\} / \log \mu).$$

Several methods are available to obtain a good initial value. One suggestion is to increase the sample size  $|\Omega|$  [4, 49]. For an  $n_1 \times \cdots \times n_d$ -dimensional tensor, the sample size can potentially reach  $|\Omega| = n_1 \cdots n_d$ , while the number of parameters is  $r \sum_{k=1}^d n_k$ . Since each step of the algorithm is a ridge regression, a larger sample size leads to more accurate estimation. However, sample size is not the only determinant. As discussed in [13] and [20], there does not exist a best rank- $r$  approximation for high-order tensors in general. Alternatively, one could employ the branch-and-bound technique [11, 24], or utilize multiple random start points. These techniques could result in a satisfactory local minimum if computational capacity allows.

Since the parameter space is open, a global minimum is also a local minimum if it exists. Therefore, Proposition 4.3 applies if an initial value is in a small neighborhood of the global minimum.

**COROLLARY 4.1.** *Suppose  $\mathcal{D}$  is open, and that a global minimizer  $(\tilde{P}, \tilde{Q})$  of  $L(P, Q|\mathbf{Y})$  exists. Let  $H$  be positive definite. For a small neighborhood  $\mathcal{V}$  of  $(\tilde{P}, \tilde{Q})$ , suppose  $(P_{s_0}, Q_{s_0}) \in \mathcal{V}$  for some  $s_0 \geq 0$ , then a sequence  $\{(P_s, Q_s)\}_{s \geq s_0} \subset \mathcal{V}$  obtained from Algorithm 1 exists, and converges at least linearly to  $(\tilde{P}, \tilde{Q})$  in the energy norm. That is, there exists  $\mu \in [0, 1)$ , such that*

$$\|(P_{s+1}, Q_{s+1}) - (\tilde{P}, \tilde{Q})\|_E \leq \mu \|(P_s, Q_s) - (\tilde{P}, \tilde{Q})\|_E.$$

The proof is a straightforward exercise of Proposition 4.3 given that  $\nabla L = \mathbf{0}$  and  $\tilde{H}$  is positive. Nevertheless, the existence of a global minimum is not guaranteed. For instance, as demonstrated in [33] and [13], a tensor might be approximated arbitrarily well by lower-rank tensors. De Silva and Lim (2008) also demonstrate that the best rank- $r$  approximation problem could be ill-posed: for tensors with some rank  $r$  in certain tensor spaces, there is a strictly positive probability that a global minimum cannot be obtained. For the recommender system framework, this issue is further complicated since a large proportion of tensor entries is missing. In the proposed setting, however, the criterion function  $L(\cdot|\mathbf{Y})$  is always bounded below by zero. Therefore, even if a global minimum does not exist, we can still minimize  $L(\cdot|\mathbf{Y})$  such that it is sufficiently close to its infimum.

**4.3. Asymptotic properties.** In this subsection, we derive asymptotic properties for the proposed method. Specifically, we prove consistency of estimated parameters when the sample size goes to infinity. The result holds true under the  $L_2$ -loss, or more general loss functions with additional smoothness conditions. As illustrated in the previous section, the global minimum of the criterion function

may not exist, but the criterion function is bounded below by zero. We demonstrate that our asymptotic properties still hold even if the estimated parameter is not a global minimizer, as long as the criterion function converges to its infimum.

4.3.1. *Consistency under the  $L_2$ -loss.* In this section, we focus on the asymptotic properties of the predicted values instead of the latent and nested factors  $(P, Q)$ , since prediction accuracy instead of parameter estimation is the primary concern.

For this purpose, we redefine the parameter space. Suppose  $\mathbf{Y}$  is a  $d$ th order tensor with dimension  $n_1 \times \dots \times n_d$ . Let  $\mathbf{Y} = \Theta + \mathcal{E}$ , where  $\Theta = E(\mathbf{Y})$  is our primary interest, and  $\mathcal{E}$  is an  $n_1 \times \dots \times n_d$ -dimensional random error. We assume that  $\mathcal{E}$  has i.i.d. elements with mean 0, variance  $\sigma^2$  and a finite moment generating function at an open interval containing 0. For an arbitrary element  $y_{i_1 i_2 \dots i_d}$  of  $\mathbf{Y}$ , the  $L_2$ -loss function is

$$l(\Theta, y_{i_1 i_2 \dots i_d}) = (y_{i_1 i_2 \dots i_d} - \theta_{i_1 i_2 \dots i_d})^2,$$

where  $\theta_{i_1 i_2 \dots i_d} = \sum_{j=1}^r (p_{i_1 j}^1 + q_{i_1 j}^1)(p_{i_2 j}^2 + q_{i_2 j}^2) \dots (p_{i_d j}^d + q_{i_d j}^d)$  is the corresponding element in  $\Theta$  and is a function of  $P$  and  $Q$ . Notice that each  $l(\Theta, y_{i_1 i_2 \dots i_d})$  relies on  $\Theta$  only through  $\theta_{i_1 i_2 \dots i_d}$ . Since, in practice, the utilities are usually nonnegative and finite, we assume that  $\|(P, Q)\|_\infty \leq c_0$ , where  $c_0$  is a positive constant. Let  $J(\Theta)$  be a nonnegative penalty function. Then the overall criterion function is redefined as

$$(4.1) \quad L(\Theta|\mathbf{Y}) = \sum_{(i_1, \dots, i_d) \in \Omega} l(\Theta, y_{i_1 i_2 \dots i_d}) + \lambda_{|\Omega|} J(\Theta) \quad \text{for } \Theta \in \mathcal{S},$$

where  $\lambda_{|\Omega|}$  is the penalization coefficient and  $\mathcal{S} \subseteq \mathbb{R}^{n_1 \times \dots \times n_d}$  is the parameter space for  $\Theta$ .

Let  $\Theta_0$  be the unique true parameter. We assume that  $\hat{\Theta}_{|\Omega|}$  is a sample estimator of  $\Theta_0$  satisfying

$$(4.2) \quad L(\hat{\Theta}_{|\Omega|}|\mathbf{Y}) \leq \inf_{\Theta \in \mathcal{S}} L(\Theta|\mathbf{Y}) + \tau_{|\Omega|},$$

where  $\lim_{|\Omega| \rightarrow \infty} \tau_{|\Omega|} = 0$ . Note that  $\hat{\Theta}_{|\Omega|}$  is identifiable by Proposition 4.1. Condition (4.2) implies that  $\hat{\Theta}_{|\Omega|}$  converges to a global minimizer of  $L(\Theta|\mathbf{Y})$  as  $|\Omega| \rightarrow \infty$ . However, the verification of (4.2) could still be challenging in that  $L$  is nonconvex in  $\Theta$  given missing values associated with  $\mathbf{Y}$ . Nevertheless, for a finite sample, condition (4.2) does not impose any restrictions on the existence of a global minimum, and  $\hat{\Theta}_{|\Omega|}$  is not required to be a global minimizer even if one exists. In practice, one could use a good initial point to sufficiently reduce the value of  $L$ , as discussed in Section 4.2.2, which would lead to more satisfactory numerical results.

Let  $l_{\Delta}(\Theta|\cdot) = l(\Theta, \cdot) - l(\Theta_0, \cdot)$  be the loss difference, and

$$(4.3) \quad K(\Theta_0, \Theta) = \frac{1}{n_1 \cdots n_d} \sum_{i_1=1}^{n_1} \cdots \sum_{i_d=1}^{n_d} E\{l_{\Delta}(\Theta, y_{i_1 i_2 \cdots i_d})\}$$

be the expected loss difference. Since  $\Theta_0$  is the unique true parameter, we have  $K(\Theta_0, \Theta) \geq 0$  for all  $\Theta \in \mathcal{S}$  and  $K = 0$  only if  $\Theta = \Theta_0$ . We define the distance between  $\Theta$  and  $\Theta_0$  as  $\rho(\Theta_0, \Theta) = K^{1/2}(\Theta_0, \Theta)$ , and let

$$(4.4) \quad V(\Theta_0, \Theta) = \frac{1}{n_1 \cdots n_d} \sum_{i_1=1}^{n_1} \cdots \sum_{i_d=1}^{n_d} \text{Var}\{l_{\Delta}(\Theta, y_{i_1 i_2 \cdots i_d})\}.$$

Then, under the  $L_2$ -loss, we have  $K(\Theta_0, \Theta) = \frac{1}{n_1 \cdots n_d} \|\Theta_0 - \Theta\|^2$  and  $V(\Theta_0, \Theta) = \frac{4\sigma^2}{n_1 \cdots n_d} \|\Theta_0 - \Theta\|^2$ , where  $\|\cdot\|$  stands for the Euclidean norm of the vectorized tensor.

**THEOREM 4.1.** *Suppose  $\hat{\Theta}_{|\Omega|}$  is a sample estimator satisfying (4.2). Then we have*

$$P(\rho(\hat{\Theta}_{|\Omega|}, \Theta_0) \geq \eta_{|\Omega|}) \leq 7 \exp(-c_1 |\Omega| \eta_{|\Omega|}^2),$$

where  $c_1 \geq 0$  is a constant,  $\eta_{|\Omega|} = \max(\varepsilon_{|\Omega|}, \lambda_{|\Omega|}^{1/2})$ , and  $\varepsilon_{|\Omega|} \sim \frac{1}{|\Omega|^{1/2}}$  is the best possible rate achieved when  $\lambda_{|\Omega|} \sim \varepsilon_{|\Omega|}^2$ .

Theorem 1 states that if the penalty term shrinks to zero at a rate no slower than the rate  $\varepsilon_{|\Omega|}^2$  as the sample size tends to infinity, then the proposed method can achieve the convergence rate of  $\frac{1}{|\Omega|^{1/2}}$ , the same rate as the maximum likelihood estimator. In addition, the development of the convergence rate is under the  $L_2$  distance, which, as a special case of the Kullback–Leiber information, is stronger than the commonly used Hellinger distance [39].

**4.3.2. General asymptotic properties.** Next, we develop the estimation consistency under more general settings. For each element of  $\mathbf{Y}$ , we assume

$$E(y_{i_1 \cdots i_d}) = \nu(\theta_{i_1 \cdots i_d}),$$

where  $\nu(\cdot)$  is a mean function. For example, when  $\mathbf{Y}$  is binary, we might adopt the logistic link  $\nu(\theta) = \frac{\exp(\theta)}{1 + \exp(\theta)}$ , and when  $\mathbf{Y}$  is ordinal, we have  $\nu(\theta) = \exp(\theta)$ . We also assume  $\text{Var}(y_{i_1 \cdots i_d}) < \infty$ .

In this general setting,  $l(\cdot, \cdot)$  in (4.1) is not necessarily an  $L_2$ -loss function. Let  $K(\Theta_0, \Theta)$  and  $V(\Theta_0, \Theta)$  be defined as in (4.3) and (4.4), respectively, and let  $\rho(\Theta_0, \Theta) = K^{1/2}(\Theta_0, \Theta)$ . Here,  $K(\cdot, \cdot)$  reduces to the Kullback–Leiber pseudo distance if  $l(\cdot, \cdot)$  corresponds to a log-likelihood. Let  $W_p^\alpha[a, b]^{n_1 \times \cdots \times n_d}$  be a Sobolev space with finite  $L_p$ -norm, where  $a$  and  $b$  are some constants and  $\alpha$  is the parameter associated with the degree of smoothness of functions [14].



ASSUMPTION 4.2. For each  $y_{i_1 \dots i_d}$ , suppose

$$|l(\Theta_0, y_{i_1 \dots i_d}) - l(\Theta, y_{i_1 \dots i_d})| \leq g(y_{i_1 \dots i_d}) \|\Theta_0 - \Theta\|,$$

where  $g(\cdot)$  satisfies  $E[\exp\{t_0 g(y_{i_1 \dots i_d})\}] \leq c_2 < \infty$ , for a constant  $c_2$  and some constants  $t_0$  around 0. In particular, there exists a constant  $c'_2 > 0$ , such that  $E\{g^2(y_{i_1, \dots, i_d})\} \leq c'_2$  for all  $y_{i_1 \dots i_d}$ 's.

ASSUMPTION 4.3. Suppose there exist  $\delta > 0$  and  $\beta \in [0, 1)$ , such that for a  $\delta$ -ball centered at  $\Theta_0$ , we have  $\rho(\Theta_0, \Theta) \geq c_3 \|\Theta_0 - \Theta\|^{\frac{1}{1+\beta}}$ , where  $c_3 \geq 0$  is a constant.

Assumption 4.3 indicates that in a neighborhood of the true parameter  $\Theta_0$ , the distance  $\rho(\Theta_0, \cdot)$  is no smaller than the Euclidean distance up to a certain order. On the contrary, if  $\rho(\Theta_0, \cdot)$  is dominated by the Euclidean distance for all neighborhoods of  $\Theta_0$ , then the convergence result under  $\rho(\cdot, \cdot)$  can be shown similar to the proof of Theorem 4.1.

THEOREM 4.2. Let  $\hat{\Theta}_{|\Omega|}$  be a sample estimator satisfying (4.2). Assume that  $l_\Delta \in W_p^\alpha[a, b]^{n_1 \times \dots \times n_d}$ , where  $p > 2$ , and that Assumptions 4.2 and 4.3 hold. Then

$$P(\rho(\hat{\Theta}_{|\Omega|}, \Theta_0) \geq \eta_{|\Omega|}) \leq 7 \exp(-c_4 |\Omega| \eta_{|\Omega|}^2),$$

where  $c_4 \geq 0$  is a constant, and  $\eta_{|\Omega|} = \max(\varepsilon_{|\Omega|}, \lambda_{|\Omega|}^{1/2})$  with

$$\varepsilon_{|\Omega|} \sim \begin{cases} \left(\frac{1}{|\Omega|^{1/2}}\right)^{\frac{2\omega}{2\omega+1}} & \text{if } \omega > \frac{1}{2} \\ \left(\frac{1}{|\Omega|^{1/2}}\right)^\omega & \text{if } \omega \leq \frac{1}{2} \end{cases}$$

being the best possible rate, which can be achieved when  $\lambda_{|\Omega|} \sim \varepsilon_{|\Omega|}^2$ . Here,  $\omega = \alpha/\gamma$ , and  $\gamma = \sum_{k=1}^d (n_k + m_k)r$  is the total number of parameters.

The assumption of  $p > 2$  can be relaxed to  $p \geq 2$  if  $p\omega$  does not go to 0 as  $|\Omega| \rightarrow \infty$ . Notice that the convergence rate in Theorem 4.2 becomes  $\varepsilon_{|\Omega|} \sim \frac{1}{|\Omega|^{1/2}}$  if  $\omega = \infty$ , which is the convergence rate of the maximum likelihood estimator achieved in Theorem 4.1.

**5. Simulation studies.** In this section, we perform simulations to compare the proposed method (REM) with five competing tensor factorization methods. Three methods are existing methods, namely Bayesian probabilistic tensor factorization (BPTF; [47]), the factorization machine (libFM; [36]) and the Gaussian process

factorization machine (GPFM; [31]). The programming codes of these three methods can be obtained from <https://www.cs.cmu.edu/~lxiong/bptf/bptf.html>, <http://www.libfm.org/>, and <http://trungngv.github.io/gpfm/>, respectively. Since subgroup information is available, another naive but effective tensor factorization method is to conduct CPD for each subgroup separately, and combine the final result. This method is referred to as the groupwise Canonical Polyadic Decomposition (GCPD). In addition, we also investigate the performance of matrix factorization (MF) under the tensor framework, which is the misspecified proposed method with  $d = 2$  and ignoring contextual information.

*5.1. The “cold-start” problem.* The first simulation study is designed to compare the performance of each method under various severity levels of the “cold-start” problem. Specifically, we consider a third-order tensor with *user*, *item* and one *contextual variable*. We set the number of users  $n_1 = 400$ , the number of items  $n_2 = 1100$  and the number of contexts  $n_3 = 9$ . We assume that the users, items and contexts are from  $m_1 = 10$ ,  $m_2 = 11$  and  $m_3 = 3$  subgroups, respectively, and assume the number of latent factors  $r = 3$ . We generate each latent factor  $\mathbf{p}_{i_k}^k \stackrel{iid}{\sim} N(0, \mathbf{I}_r)$  for  $i_k = 1, \dots, n_k$ , and  $k = 1, 2, 3$ . To distinguish different subgroups, we set the nested factors as a simple ordered sequence, where  $\mathbf{q}_{(u_1)}^1 = (-5.5 + u_1)\mathbf{1}_r$ ,  $\mathbf{q}_{(u_2)}^2 = (-3.6 + 0.6u_2)\mathbf{1}_r$ , and  $\mathbf{q}_{(u_3)}^3 = (-4 + 2u_3)\mathbf{1}_r$  for  $u_k = 1, \dots, m_k$ . Users, items and contexts are evenly assigned to each subgroup.

For each simulation, we generate  $N = n_1 n_2 n_3 (1 - \pi_0)$  entries out of the entire tensor, where  $\pi_0 = 80\%$ ,  $95\%$  or  $99\%$  are the missing percentages. Furthermore, we use  $\phi_{cs}$  to measure the severity of the “cold-start” problem, and  $\phi_{cs} = 30\%$ ,  $60\%$  or  $95\%$  represents the proportion of the testing data whose utilities are about new items and are not available from the training set. Each utility is generated by  $y_{i_1 i_2 i_3} = \sum_{j=1}^r (p_{i_1 j}^1 + q_{i_1 j}^1)(p_{i_2 j}^2 + q_{i_2 j}^2)(p_{i_3 j}^3 + q_{i_3 j}^3)/3 + \varepsilon$ , where  $\varepsilon \sim N(0, 1)$  is the random error.

For all methods, we assume that  $r = 3$  is known. For REM and GCPD, we also assume that the subgroup memberships are correctly specified, and the tuning parameter  $\lambda$  is preselected from grid points ranging from 1 to 11. Since the subgroup structure is assumed to be known, these two methods have more advantage and are expected to have better performance. For MF, we assume the same setting as the proposed method. For BPTF, we choose the number of Gibbs samples to be 50 and keep the remaining parameters by their default choices. For libFM, we use their default setting, and for GPFM, we select the radial basis function kernel with noise being equal to 10 and the standard deviation of latent variables being equal to 1. All methods are replicated by 200 simulation runs.

Table 1 provides the performance of each method based on the root mean square error (RMSE) and the mean absolute error (MAE), where the MAE is defined as  $\frac{1}{|\Omega|} \sum_{\Omega} |y_{i_1 \dots i_d} - \hat{y}_{i_1 \dots i_d}|$ . We observe that most methods perform worse when either the missing percentage or the severity of the “cold-start” problem increases.

TABLE 1

The proposed method (REM) is compared to matrix factorization (MF), groupwise Canonical Polyadic Decomposition (GCPD), Bayesian probabilistic tensor factorization (BPTF; [47]), factorization machine (libFM; [36]) and the Gaussian process factorization machine (GPFM; [31]) on simulated third-order tensors with different missing rates and different degrees of the “cold-start” problem; the RMSE and the MAE are provided with standard error in each parenthesis;  $\pi_0$  and  $\phi_{cs}$  represent the missing rate and the severity of the “cold-start” problem, respectively; and the simulation results are based on 200 replications

			$\phi_{cs} = 0.3$	$\phi_{cs} = 0.6$	$\phi_{cs} = 0.95$	
$\pi_0 = 80\%$	REM	RMSE	4.586 (0.562)	6.467 (0.823)	8.228 (1.052)	
		MAE	2.210 (0.214)	3.606 (0.432)	5.292 (0.687)	
	MF	RMSE	10.307 (1.012)	10.279 (1.020)	10.373 (1.062)	
		MAE	6.789 (0.658)	6.783 (0.673)	6.822 (0.700)	
	GCPD	RMSE	5.892 (0.754)	8.083 (0.851)	9.949 (0.974)	
		MAE	2.781 (0.226)	4.467 (0.421)	6.389 (0.628)	
	BPTF	RMSE	5.793 (0.590)	8.142 (0.837)	10.192 (1.050)	
		MAE	2.645 (0.222)	4.442 (0.438)	6.518 (0.685)	
	libFM	RMSE	10.361 (1.070)	10.373 (1.070)	10.355 (1.073)	
		MAE	6.778 (0.707)	6.786 (0.713)	6.768 (0.718)	
	GPFM	RMSE	9.017 (1.391)	9.852 (1.894)	11.068 (1.406)	
		MAE	5.215 (1.131)	5.624 (0.761)	6.992 (0.814)	
	$\pi_0 = 95\%$	REM	RMSE	3.322 (0.510)	4.658 (0.773)	6.082 (1.055)
			MAE	1.760 (0.196)	2.702 (0.401)	3.943 (0.674)
MF		RMSE	10.768 (1.179)	10.728 (1.148)	10.656 (1.160)	
		MAE	7.055 (0.761)	7.045 (0.751)	6.992 (0.761)	
GCPD		RMSE	13.489 (2.059)	12.247 (1.632)	10.461 (1.153)	
		MAE	6.213 (0.826)	6.386 (0.767)	6.638 (0.701)	
BPTF		RMSE	5.847 (0.696)	8.173 (0.881)	10.197 (1.050)	
		MAE	2.736 (0.411)	4.498 (0.507)	6.524 (0.683)	
libFM		RMSE	10.384 (1.074)	10.395 (1.071)	10.359 (1.076)	
		MAE	6.789 (0.717)	6.799 (0.716)	6.769 (0.719)	
GPFM		RMSE	10.397 (1.072)	10.406 (1.070)	10.369 (1.078)	
		MAE	6.812 (0.718)	6.822 (0.717)	6.792 (0.720)	
$\pi_0 = 99\%$		REM	RMSE	3.361 (1.032)	4.329 (1.379)	4.988 (1.582)
			MAE	1.865 (0.414)	2.588 (0.722)	3.289 (1.021)
	MF	RMSE	12.647 (1.940)	12.612 (1.883)	12.379 (1.706)	
		MAE	8.016 (1.071)	8.016 (1.050)	7.903 (0.981)	
	GCPD	RMSE	10.925 (1.191)	10.808 (1.124)	10.283 (1.058)	
		MAE	6.748 (0.755)	6.808 (0.721)	6.685 (0.691)	
	BPTF	RMSE	8.246 (2.829)	9.243 (1.737)	10.268 (1.093)	
		MAE	4.696 (1.917)	5.537 (1.256)	6.616 (0.723)	
	libFM	RMSE	10.387 (1.058)	10.389 (1.058)	10.342 (1.075)	
		MAE	6.790 (0.707)	6.792 (0.713)	6.759 (0.717)	
	GPFM	RMSE	10.450 (1.068)	10.456 (1.072)	10.410 (1.084)	
		MAE	6.874 (0.712)	6.878 (0.722)	6.845 (0.723)	

In contrast, the proposed method is relatively robust against these changes. Specifically, REM performs the best across all settings, especially in the worst setting when the missing percentage is 99% and the proportion of new items reaches 95%, that is, 95% of the items are new and are not available from the training set. In this scenario, the proposed method is at least 100% better than other methods in terms of both the RMSE and the MAE.

*5.2. High-order tensors.* We design the second simulation study to evaluate the performance of the proposed method under fourth-order tensors. Specifically, we let the number of users and the number of items be the same, namely, 500 or 1000, and the number of contexts be 4 for the two contextual variables. Furthermore, we allow 10 subgroups for users and for items, and 2 subgroups for each of the two contextual variables. We also assume that the number of members in each subgroup is the same. The number of latent factors is  $r = 3$ , and each latent factor is generated from an i.i.d. standard normal distribution. The nested factors for users and items are  $(-5.5 + u)\mathbf{1}_r$  for  $u = 1, \dots, 10$ , and the nested factors for the two contextual variables are  $-0.25 \cdot \mathbf{1}_r$  and  $0.25 \cdot \mathbf{1}_r$  corresponding to the two subgroups. Each utility is generated as  $y_{i_1 i_2 i_3 i_4} = \sum_{j=1}^r (p_{i_1 j}^1 + q_{i_1 j}^1)(p_{i_2 j}^2 + q_{i_2 j}^2)(p_{i_3 j}^3 + q_{i_3 j}^3)(p_{i_4 j}^4 + q_{i_4 j}^4)/4 + \varepsilon$ , where  $\varepsilon \sim N(0, 1)$  is the random error.

For each replication in the tensor, we assume that the missing percentage  $\pi_0 = 95\%$ ,  $97\%$  or  $99\%$ , corresponding to the high-missing situation for high-order tensors. Furthermore, we assume that 30% of the items are not available in the training set, that is, the ‘‘cold-start’’ severity level is fixed. The tuning parameter selection for each method is the same as in simulation study 1. All methods are replicated by 200 simulations. REM, GCPD, libFM and GPFM are able to utilize the fourth order of the tensor, while MF and BPTF use up to the second and the third order, respectively.

Table 2 provides the comparisons of the proposed method and other methods under the fourth-order tensor setting. It is clear that REM has the overall best performance in terms of both RMSEs and MAEs. For the two misspecified methods, namely the MF and BPTF, their performances are similar to each other and close to the unreported grand mean imputation, although BPTF has better performance when the missing percentage is low. For the three correctly-specified competing methods, namely GCPD, libFM and GPFM, their performances are not significantly better than the misspecified methods, although they utilize all tensor information. In contrast, the proposed method provides much smaller RMSEs and MAEs in all settings. For example, when the missing percentage is 99% and the number of users and items are both equal to 1000, the existing methods perform mostly similar to that of the grand mean imputation, producing RMSEs slightly above 4 and MAEs around 2.6. That is, these methods do not utilize subject-specific information effectively. In contrast, the proposed method is able to improve on both the RMSE and the MAE by more than 100%.

TABLE 2

The proposed method (REM) is compared to matrix factorization (MF), groupwise Canonical Polyadic Decomposition (GCPD), Bayesian probabilistic tensor factorization (BPTF; [47]), factorization machine (libFM; [36]) and the Gaussian process factorization machine (GPFM; [31]) on simulated fourth-order tensors with different missing rates and different number of users and items; the RMSE and the MAE are provided with standard error in each parenthesis;  $\hat{d}$  is the assumed tensor order for each method with the true order being 4;  $\pi_0$ ,  $n_1$  and  $n_2$  represent the missing rate, the number of users and the number of items, respectively; and the simulation results are based on 200 replications

		$\hat{d}$		$n_1 = n_2 = 500$	$n_1 = n_2 = 1000$
$\pi_0 = 95\%$	REM	$\hat{d} = 4$	RMSE	1.427 (0.351)	1.799 (0.570)
			MAE	1.015 (0.138)	1.148 (0.202)
	MF	$\hat{d} = 2$	RMSE	4.040 (1.206)	4.099 (1.373)
			MAE	2.582 (0.679)	2.601 (0.759)
	GCPD	$\hat{d} = 4$	RMSE	4.691 (1.906)	4.565 (1.807)
			MAE	2.599 (0.732)	2.395 (0.638)
	BPTF	$\hat{d} = 3$	RMSE	3.944 (1.212)	3.896 (1.322)
			MAE	2.472 (0.650)	2.454 (0.722)
	libFM	$\hat{d} = 4$	RMSE	4.057 (1.172)	4.136 (1.423)
			MAE	2.578 (0.655)	2.585 (0.773)
	GPFM	$\hat{d} = 4$	RMSE	3.958 (1.103)	3.052 (1.194)
			MAE	2.533 (0.640)	1.916 (0.602)
$\pi_0 = 97\%$	REM	$\hat{d} = 4$	RMSE	1.512 (0.670)	1.689 (0.524)
			MAE	1.050 (0.227)	1.108 (0.174)
	MF	$\hat{d} = 2$	RMSE	4.048 (1.210)	4.103 (1.375)
			MAE	2.588 (0.682)	2.605 (0.761)
	GCPD	$\hat{d} = 4$	RMSE	4.640 (1.792)	5.384 (1.876)
			MAE	2.665 (0.795)	2.779 (0.699)
	BPTF	$\hat{d} = 3$	RMSE	4.047 (1.264)	3.952 (1.345)
			MAE	2.503 (0.662)	2.473 (0.726)
	libFM	$\hat{d} = 4$	RMSE	4.087 (1.214)	4.198 (1.371)
			MAE	2.593 (0.677)	2.648 (0.768)
	GPFM	$\hat{d} = 4$	RMSE	4.115 (1.215)	3.604 (1.128)
			MAE	2.616 (0.682)	2.341 (0.634)
$\pi_0 = 99\%$	REM	$\hat{d} = 4$	RMSE	2.780 (1.915)	1.880 (1.540)
			MAE	1.579 (0.742)	1.187 (0.433)
	MF	$\hat{d} = 2$	RMSE	4.108 (1.233)	4.124 (1.390)
			MAE	2.625 (0.692)	2.621 (0.769)
	GCPD	$\hat{d} = 4$	RMSE	4.168 (1.364)	4.453 (1.680)
			MAE	2.567 (0.718)	2.611 (0.753)
	BPTF	$\hat{d} = 3$	RMSE	4.399 (1.477)	4.174 (1.497)
			MAE	2.620 (0.706)	2.540 (0.754)
	libFM	$\hat{d} = 4$	RMSE	4.124 (1.226)	4.164 (1.399)
			MAE	2.613 (0.678)	2.624 (0.761)
	GPFM	$\hat{d} = 4$	RMSE	4.121 (1.221)	4.248 (1.391)
			MAE	2.619 (0.680)	2.678 (0.778)

**6. IRI marketing data.** In this section, we apply the proposed method to IRI marketing data [8]. The data contain 116.3 million observations of average sales volumes collected from 2447 grocery stores on 161,114 products from 2001 to 2011. There are 30 promotion strategies for various products to attract consumers. Each observation consists of a store ID, a product ID, a promotion strategy and the corresponding average sales volume. The 2447 stores are selected from 47 markets across the United States, where demographic information within two miles of each store is collected as well. The 161,114 products include all items sold from these stores during the 11-year period. These products can be classified into 31 categories, including beer, coffee, frozen pizza, paper towels, etc. The 30 promotion strategies are combinations of 5 advertisement features, 3 types of merchandise display and an indicator on whether the product has a price reduction of more than 5%. The data have a 99% missing rate after being reorganized into a third-order tensor by store, product and promotion.

The goal of our study is to predict the average sales volume of each product from each store, and the average sales volume when a particular promotion strategy is applied. Through this prediction procedure, we are able to potentially recommend the most profitable products for each store, and evaluate how each specific promotion strategy plays a role for each product sales.

The data are randomly split into a 50% training set, a 25% validation set and a 25% testing set. The random split is replicated 50 times. Sales volumes from each category of product are standardized before analysis to avoid large differences of sales volumes from different categories. We compare the proposed method with the existing methods listed in Section 5.1. In addition, we also compare it to the grand mean imputation where all missing elements in the tensor are imputed by the mean of the observed values. Tuning parameters for each method are selected from a wide range of grid points to minimize the RMSE on the validation set. Most methods require that the number of latent factors  $r \geq 15$  in order to capture a majority of the data variation. For the proposed method, we classify stores, products and promotion strategies into subgroups based on their geographical locations (the first digit of the zip code), product categories and whether a price reduction is applied, respectively.

Since the data are standardized, the grand mean imputation returns an RMSE close to 1, which can be regarded as a benchmark basis for comparison. Table 3 indicates that the proposed REM has the best performance in terms of both RMSE and MAE. Specifically, REM improves the RMSE of the MF by the largest percentage, which demonstrates the great advantage of incorporating contextual information through tensor structure. In addition, the RMSE of the proposed method is less than that of BPTF and libFM, illustrating that REM has better prediction accuracy among the competing tensor factorization methods. GPFM does not converge due to the high missing rate and the large number of parameters involved. Meanwhile, BPTF has nearly the same MAE as the proposed method, but produces a larger RMSE, indicating that BPTF's performance is possibly better than

TABLE 3

The proposed method (REM) is compared to matrix factorization (MF), groupwise Canonical Polyadic Decomposition (GCPD), Bayesian probabilistic tensor factorization (BPTF; [47]), factorization machine (libFM; [36]), the Gaussian process factorization machine (GPFM; [31]), and the grand mean imputation (GMI; as a reference level) on 116 million IRI marketing data points; GPFM is not included because it fails to converge; and the other results are based on 50 replications of random testing sets. In the first tabular, the comparison is made based on root mean square error (RMSE), mean absolute error (MAE) and computational time in hours (Comp. Time); and in the second tabular, the relative improvement of the proposed method over existing methods is provided

	RMSE	MAE	Comp. Time (h)
REM	0.637 (0.009)	0.209 (0.001)	3.9
MF	0.969 (0.010)	0.371 (0.007)	0.7
GCPD	0.640 (0.010)	0.229 (0.001)	5.4
BPTF	0.782 (0.155)	0.209 (0.001)	8.4
libFM	0.705 (0.010)	0.236 (0.001)	0.5
GMI	1.000 (0.006)	0.392 ( $4.483 \times 10^{-5}$ )	N/A

  

	RMSE	MAE
MF	34.2%	43.7%
GCPD	0.5%	8.7%
BPTF	18.5%	0%
libFM	9.6%	11.4%
GMI	36.3%	46.7%

the proposed method for certain subjects, but has inferior performance for the rest of the subjects.

Furthermore, the naive method GCPD has the second best performance. This might be explained by the following two reasons. One is that the GCPD utilizes additional subgroup information as does the REM, which is not applicable to the other competing methods. The other is that, unlike in the simulation studies, the size of the IRI data is large enough so that the number of observations within each subgroup is sufficient to make good estimations.

Most of the numerical studies are implemented on Dell C8220 computing sleds equipped with two 10-core Intel Xeon E5-2670V2 processors and 64 GB RAM. The running time for each method is provided in Table 3. Notice that BPTF's method also requires large memory storage due to the high demand of the Gibbs' samples. Reducing the number of Gibbs' samples may lead to more efficient computation but less accurate predictions.

**7. Discussion.** In this article, we propose a new tensor-based recommender system which makes recommendations through incorporating contextual informa-



tion. A unique contribution of our method is that we achieve tensor completion through utilizing an additional layer of nested factors, in addition to applying the latent factors as in the commonly used canonical polyadic decomposition. The nested factors incorporate subject-subject dependency, which are estimated based on subgrouping. This leads to a significant advantage in solving the “cold-start” problem effectively. That is, for a new subject, information from other subgroup members can be borrowed to make more accurate predictions even though the subject’s own information is not collected sufficiently. In addition, the proposed method is able to address high-order tensors which are beyond the third order. Existing methods are not effective in handling the high-order tensor problem due to the high computational cost and the high missing rate. We propose a new algorithm that integrates maximum block improvement into the blockwise coordinate descent algorithm, which avoids operating high-order tensors directly and, therefore, achieves scalable computation. Moreover, the proposed nested factors borrow information from all members in the same subgroup, and hence accommodate a high missing rate.

The proposed method also shows excellent numerical performance and theoretical properties. In Section 6, the proposed method is applied to IRI marketing data which consists of 116 million observations. The proposed method improves prediction accuracy compared to existing methods with relatively small computational cost. In theory, we demonstrate the convergence properties of the proposed algorithm, which converges to a stationary point from an arbitrary initial point, and local convergence to a local minimum with linear convergence rate. The estimated parameter achieves asymptotic consistency under the  $L_2$ -loss function and other more general circumstances.

**Acknowledgments.** The authors thank the Associate Editor and two anonymous reviewers for their suggestions and helpful feedback which improved the paper significantly.

#### SUPPLEMENTARY MATERIAL

**Supplement to “Multilayer tensor factorization with applications to recommender systems.”** (DOI: [10.1214/17-AOS1659SUPP](https://doi.org/10.1214/17-AOS1659SUPP); .pdf). Technical proof of all lemmas, propositions and theorems are provided in the supplementary material [5].

#### REFERENCES

- [1] ADOMAVICIUS, G. and TUZHILIN, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.* **17** 734–749.
- [2] ADOMAVICIUS, G. and TUZHILIN, A. (2011). Context-aware recommender systems. In *Recommender Systems Handbook* 217–253. Springer, Berlin.

- [3] ASWANI, A. (2016). Low-rank approximation and completion of positive tensors. *SIAM J. Matrix Anal. Appl.* **37** 1337–1364. [MR3551210](#)
- [4] BHOJANAPALLI, S. and SANGHAVI, S. (2015). A new sampling technique for tensors. Preprint. Available at [arXiv:1502.05023](#).
- [5] BI, X., QU, A. and SHEN, X. (2018). Supplement to “Multilayer tensor factorization with applications to recommender systems.” DOI:[10.1214/17-AOS1659SUPP](#).
- [6] BI, X., QU, A., WANG, J. and SHEN, X. (2017). A group specific recommender system. *J. Amer. Statist. Assoc.* **112** 1344–1353.
- [7] BOBADILLA, J., ORTEGA, F., HERNANDO, A. and GUTIÉRREZ, A. (2013). Recommender systems survey. *Knowl.-Based Syst.* **46** 109–132.
- [8] BRONNENBERG, B. J., KRUGER, M. W. and MELA, C. F. (2008). Database paper—The IRI marketing data set. *Mark. Sci.* **27** 745–748.
- [9] CHEN, B., HE, S., LI, Z. and ZHANG, S. (2012). Maximum block improvement and polynomial optimization. *SIAM J. Optim.* **22** 87–107.
- [10] CHI, E. C. and KOLDA, T. G. (2012). On tensors, sparsity, and nonnegative factorizations. *SIAM J. Matrix Anal. Appl.* **33** 1272–1299.
- [11] CLAUSEN, J. (1999). Branch and bound algorithms-principles and examples. Technical Report, Univ. Copenhagen.
- [12] COLOMBO-MENDOZA, L. O., VALENCIA-GARCÍA, R., RODRÍGUEZ-GONZÁLEZ, A., ALOR-HERNÁNDEZ, G. and SAMPER-ZAPATER, J. J. (2015). RecomMetz: A context-aware knowledge-based mobile recommender system for movie showtimes. *Expert Syst. Appl.* **42** 1202–1222.
- [13] DE SILVA, V. and LIM, L.-H. (2008). Tensor rank and the ill-posedness of the best low-rank approximation problem. *SIAM J. Matrix Anal. Appl.* **30** 1084–1127. [MR2447444](#)
- [14] DEVORE, R. A. and LORENTZ, G. G. (1993). *Constructive Approximation*. Springer, Berlin.
- [15] FANG, Y. and WANG, J. (2012). Selection of the number of clusters via the bootstrap method. *Comput. Statist. Data Anal.* **56** 468–477.
- [16] FEUERVERGER, A., HE, Y. and KHATRI, S. (2012). Statistical significance of the Netflix challenge. *Statist. Sci.* **27** 202–231. [MR2963993](#)
- [17] FORBES, P. and ZHU, M. (2011). Content-boosted matrix factorization for recommender systems: Experiments with recipe recommendation. In *Proceedings of the Fifth ACM Conference on Recommender Systems* 261–264. ACM, New York.
- [18] GOLDBERG, K., ROEDER, T., GUPTA, D. and PERKINS, C. (2001). Eigentaste: A constant time collaborative filtering algorithm. *Inf. Retr.* **4** 133–151.
- [19] KARATZOGLOU, A., AMATRIAIN, X., BALTRUNAS, L. and OLIVER, N. (2010). Multiverse recommendation: N-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the Fourth ACM Conference on Recommender Systems* 79–86. ACM, New York.
- [20] KOLDA, T. G. and BADER, B. W. (2009). Tensor decompositions and applications. *SIAM Rev.* **51** 455–500.
- [21] KOREN, Y. (2010). Collaborative filtering with temporal dynamics. *Commun. ACM* **53** 89–97.
- [22] KRUSKAL, J. B. (1977). Three-way arrays: Rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear Algebra Appl.* **18** 95–138. [MR0444690](#)
- [23] LAIRD, N. M. and WARE, J. H. (1982). Random-effects models for longitudinal data. *Biometrics* **38** 963–974.
- [24] LAND, A. H. and DOIG, A. G. (1960). An automatic method of solving discrete programming problems. *Econometrica* **28** 497–520.
- [25] LI, L. and ZHANG, X. (2017). Parsimonious Tensor Response Regression. *J. Amer. Statist. Assoc.* **112** 1131–1146. [MR3735365](#)

- [26] LI, Z., SUK, H.-I., SHEN, D. and LI, L. (2016). Sparse multi-response tensor regression for Alzheimer's disease study with multivariate clinical assessments. *IEEE Trans. Med. Imag.* **35** 1927–1936.
- [27] LI, Z., USCHMAJEV, A. and ZHANG, S. (2015). On convergence of the maximum block improvement method. *SIAM J. Optim.* **25** 210–233.
- [28] LOMBARDI, S., ANAND, S. S. and GORGOGLIONE, M. (2009). Context and customer behaviour in recommendation. In *Workshop on Context-Aware Recommender Systems*.
- [29] MIRANDA, M., ZHU, H. and IBRAHIM, J. G. (2015). TPRM: Tensor partition regression models with applications in imaging biomarker detection. Preprint. Available at [arXiv:1505.05482](https://arxiv.org/abs/1505.05482).
- [30] NGUYEN, J. and ZHU, M. (2013). Content-boosted matrix factorization techniques for recommender systems. *Stat. Anal. Data Min.* **6** 286–301. [MR3092059](https://doi.org/10.1002/sam.1125)
- [31] NGUYEN, T. V., KARATZOGLOU, A. and BALTRUNAS, L. (2014). Gaussian process factorization machines for context-aware recommendations. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval* 63–72. ACM, New York.
- [32] PAATERO, P. (1999). The multilinear engine—a table-driven, least squares program for solving multilinear problems, including the  $n$ -way parallel factor analysis model. *J. Comput. Graph. Statist.* **8** 854–888. [MR1748971](https://doi.org/10.1198/0898/854)
- [33] PAATERO, P. (2000). Construction and analysis of degenerate PARAFAC models. *J. Chemom.* **14** 285–299.
- [34] PALMISANO, C., TUZHILIN, A. and GORGOGLIONE, M. (2008). Using context to improve predictive modeling of customers in personalization applications. *IEEE Trans. Knowl. Data Eng.* **20** 1535–1549.
- [35] PARK, S.-T., PENNOCK, D., MADANI, O., GOOD, N. and DECOSTE, D. (2006). Naïve filterbots for robust cold-start recommendations. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM 699–705.
- [36] RENDLE, S. (2012). Factorization machines with libFM. *ACM Trans. Intell. Syst. Technol.* **3** 57.
- [37] RENDLE, S., GANTNER, Z., FREUDENTHALER, C. and SCHMIDT-THIEME, L. (2011). Fast context-aware recommendations with factorization machines. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval* 635–644. ACM, New York.
- [38] SALAKHUTDINOV, R., MNIH, A. and HINTON, G. (2007). Restricted Boltzmann machines for collaborative filtering. In *Proceedings of the 24th International Conference on Machine Learning* 791–798. ACM, New York.
- [39] SHEN, X. (1998). On the method of penalization. *Statist. Sinica* **8** 337–357. [MR1624410](https://doi.org/10.1007/BF02429140)
- [40] SHEN, X., TSENG, G. C., ZHANG, X. and WONG, W. H. (2003). On  $\psi$ -learning. *J. Amer. Statist. Assoc.* **98** 724–734.
- [41] SHI, Y., LARSON, M. and HANJALIC, A. (2014). Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Computing Surveys (CSUR)* **47** 3.
- [42] SREBRO, N., ALON, N. and JAAKKOLA, T. S. (2005). Generalization error bounds for collaborative prediction with low-rank matrices. In *Advances in Neural Information Processing Systems* 17 5–27.
- [43] VERBERT, K., MANOUSELIS, N., OCHOA, X., WOLPERS, M., DRACHSLER, H., BOSNIC, I. and DUVAL, E. (2012). Context-aware recommender systems for learning: A survey and future challenges. *IEEE Transactions on Learning Technologies* **5** 318–335.
- [44] WANG, J. (2010). Consistent selection of the number of clusters via crossvalidation. *Biometrika* **97** 893–904.

- [45] WANG, P., TSAI, G. and QU, A. (2012). Conditional inference functions for mixed-effects models with unspecified random-effects distribution. *J. Amer. Statist. Assoc.* **107** 725–736. [MR2980080](#)
- [46] WELLING, M. and WEBER, M. (2001). Positive tensor factorization. *Pattern Recogn. Lett.* **22** 1255–1261.
- [47] XIONG, L., CHEN, X., HUANG, T.-K., SCHNEIDER, J. and CARBONELL, J. G. (2010). Temporal collaborative filtering with Bayesian probabilistic tensor factorization. In *Proceedings of the 2010 SIAM International Conference on Data Mining* SIAM, Philadelphia, PA.
- [48] YUAN, M. and ZHANG, C.-H. (2016). Incoherent tensor norms and their applications in higher order tensor completion. Preprint. Available at [arXiv:1606.03504](#).
- [49] YUAN, M. and ZHANG, C.-H. (2016). On tensor completion via nuclear norm minimization. *Found. Comput. Math.* **16** 1031–1068.
- [50] ZHOU, H., LI, L. and ZHU, H. (2013). Tensor regression with applications in neuroimaging data analysis. *J. Amer. Statist. Assoc.* **108** 540–552.
- [51] ZHU, Y., SHEN, X. and YE, C. (2016). Personalized prediction and sparsity pursuit in latent factor models. *J. Amer. Statist. Assoc.* **111** 241–252.

X. BI  
DEPARTMENT OF BIOSTATISTICS  
YALE UNIVERSITY  
NEW HAVEN, CONNECTICUT 06520  
USA  
E-MAIL: [xuan.bi@yale.edu](mailto:xuan.bi@yale.edu)

A. QU  
DEPARTMENT OF STATISTICS  
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN  
CHAMPAIGN, ILLINOIS 61820  
USA  
E-MAIL: [anniequ@illinois.edu](mailto:anniequ@illinois.edu)

X. SHEN  
SCHOOL OF STATISTICS  
UNIVERSITY OF MINNESOTA  
MINNEAPOLIS, MINNESOTA 55455  
USA  
E-MAIL: [xshen@umn.edu](mailto:xshen@umn.edu)