Journal of Machine Learning Research 22 (2021) 1-35 Submitted 9/19;Revised 12/20; Published 2/21

Dynamic Tensor Recommender Systems

Yanqing Zhang ZHANGYANQING@YNU.EDU.CN
Department of Statistics
Yunnan University
Kunming, 650504, China
Xuan Bi XBI@UMN.EDU
Carlson School of Management
University of Minnesota
Minneapolis, MN, 55455-0438, USA
Niansheng Tang NSTANG@YNU.EDU.CN
Department of Statistics
Yunnan University
Kunming, 650504, China
Annie Qu AQU2@UCI.EDU
Department of Statistics
University of California
Irvine, CA, 92697-3425, USA

Editor: Animashree Anandkumar

Abstract

Recommender systems have been extensively used by the entertainment industry, business marketing and the biomedical industry. In addition to its capacity of providing preference-based recommendations as an unsupervised learning methodology, it has been also proven useful in sales forecasting, product introduction and other production related businesses. Since some consumers and companies need a recommendation or prediction for future budget, labor and supply chain coordination, dynamic recommender systems for precise forecasting have become extremely necessary. In this article, we propose a new recommendation method, namely the dynamic tensor recommender system (DTRS), which aims particularly at forecasting future recommendation. The proposed method utilizes a tensor-valued function of time to integrate time and contextual information, and creates a time-varying coefficient model for temporal tensor factorization through a polynomial spline approximation. Major advantages of the proposed method include competitive future recommendation predictions and effective prediction interval estimations. In theory, we establish the convergence rate of the proposed tensor factorization and asymptotic normality of the spline coefficient estimator. The proposed method is applied to simulations, IRI marketing data and Last.fm data. Numerical studies demonstrate that the proposed method outperforms existing methods in terms of future time forecasting.

Keywords: Contextual information, Dynamic recommender systems, Polynomial spline approximation, Prediction interval, Product sales forecasting

1. Introduction

Recommender systems (RS) are widely used in our daily lives, such as for selecting movies, restaurants, news articles, or online shopping. As one of the information filtering techniques, RS can help users to find interesting items through combining several information sources, e.g., users' ratings and purchasing histories, item profiles and sales volumes, time, location, and companion or promotion strategies. Particularly, incorporating time is useful in RS since users' purchase behaviors are dynamic and often highly dependent on seasonal and time factors, and business sectors also rely on dynamic recommendations to track users' changing purchase interests over time. Thus, it is essential to capture information related to time and develop time-dependent RS, and we refer this as dynamic RS (DRS).

However, developing competitive DRS brings new challenges. First, since data are streaming in over time and are time-dependent, general RS methods which are not capable of capturing time-dependency features may have reduced recommendation accuracy. Second, forecasting future recommendations accurately is also a great challenge for DRS due to the complexity of changing users' interests. For example, users might like to watch news on weekdays, but watch movies on weekends. A shoe store sells more sandals in summer and more snow boots in winter. It is important to borrow information from historical data in developing trends. Many RS methods are not designed to capture trends and predict future recommendations. In addition, as data are streaming in over time, future recommendations could involve new users or new items, whose information is not available from historical data. This is also a common problem encountered in RS, referred as the "cold start" problem.

General RS approaches include content-based filtering and collaborative filtering (CF). Traditionally, content-based filtering methods recommend similar types of items by matching a user's preferred item profile with current item's profile (e.g., Salter and Antonopoulos, 2006; Son and Kim, 2017). In contrast, CF methods recommend items by predicting item ratings for the active user based on ratings from other similar users (e.g., Herlocker et al., 2004; Luo et al., 2012). On the basis of CF methods, research work related to DRS have been developed in recent years (e.g., Koren, 2009; Gultekin and Paisley, 2014; Yu et al., 2016; Wu et al., 2017; Guo et al., 2018; Xiong et al., 2010; Rafailidis and Nanopoulos, 2014; Bi et al., 2018; Wu et al., 2019). However, most of these methods can only make recommendations for observed discrete time points, and are not designed for future recommendation prediction on unobserved time points. Liao et al. (2018) constructed dynamic tensors by means of combining tensors in tensor stream. Song et al. (2019) used temporal matrix factorization to construct temporal recommender model assuming that users' current interests are transformed from the previous time step with a Markov property. Liu and Ye (2020) proposed a dynamic three-way granularity recommendation based on matrix factorization. However, neither of these methods can handle higher-orders tensors. Moreover, these methods cannot make recommendations for future time points.

To make future recommendations, Yu et al. (2016) developed a CF method incorporating a time series model, and Wu et al. (2017, 2019) proposed CF methods incorporating long short-term memory modeling, but they cannot deal with new users, items or contextual variables. Xiong et al. (2010) used a Bayesian estimation procedure with a time-dependent constraint to estimate DRS for new users and items but cannot deal with new time points. Bi et al. (2018) created an additional layer of nested latent factors for new time points,

users and items. However, Xiong et al. (2010) and Bi et al. (2018) can only estimate the components of a tensor at fixed time points instead of at any time point in a continuous time interval. In addition, for forecasting at future time points, their methods may involve an increasing number of parameters if time is treated as an additional tensor mode, which could be computationally costly.

Currently, there are several dynamic recommender systems based on neural network approaches. For example, Ko et al. (2016) used Gated Recurrent Units (GRUs) to build collaborative sequence model. Devooght and Bersini (2017) utilized a long short-term memory (LSTM) method to address changes in the interests of a user. Wei et al. (2017) utilized the stacked denoising autoencoder (SDAEs) to extract features of items. Livne et al. (2019) applied a LSTM encoder-decoder network on sequences of contextual information. However, none of these methods are able to accommodate contextual information, and solve the "cold-start" problem simultaneously. Some of these methods may have obvious hysteresis in forecasting, which could influence the accuracy of recommendation.

In this article, we propose a tensor-valued function of time for estimating the DRS and build a new time-varying coefficient model based on tensor canonical polyadic decomposition (CPD) framework; namely, the dynamic tensor recommender system (DTRS). Specifically, we introduce a tensor-valued function of time with each mode corresponding to *user*, *item* or a *contextual variable*, where each component of the tensor is a function of time and has intra-cluster correlation. In the CPD framework, we build a time-varying coefficient model incorporating group information of time points, users, items and contexts. We approximate each coefficient function by a polynomial spline and employ group factors to explore homogeneous group effects. We adopt the weighted least square approach to incorporate intra-cluster correlation for more efficient estimation. In addition, we construct the prediction intervals of estimators of tensor components to forecast the confidence range of predicted values. In theory, we establish the convergence rate of the proposed tensor factorization and the asymptotic property of the spline parametric estimator.

The proposed method has two significant contributions. First, it can effectively provide recommendations for an entire future interval as opposed to a series of limited time points. This is because the proposed method integrates time dependency feature to the dynamic recommender systems using the time-varying coefficient model in tensor factorization to capture dynamic trends of recommender systems. In addition, the proposed method can achieve accurate forecasts for long time period through the spline extrapolation technique. Furthermore, the proposed subgroup factors extract homogeneous information from the same group to provide recommendation forecasting for future time points, and consequently solves the "cold start" problem.

Second, we establish the asymptotic distribution of the proposed estimators in that statistical inferences such as prediction interval can be formulated. In practice, it is desirable to know the upper and lower bounds for predictions, e.g., the highest possible cost, or the future sales volumes or revenues in the worst case scenario. However, existing methods on prediction intervals are mostly univariate or multivariate time series, and the prediction intervals for user-item-context interactions under a tensor framework have not been developed. In contrast, our approach allows prediction intervals for each element of a tensor-valued function, which provides a more complete picture of the dynamic recommender system over

time. Our numerical studies also demonstrate that the proposed approach provides effective prediction interval estimators.

The remainder of the paper is organized as follows. Section 2 introduces the notation and background on tensor and tensor factorization. Section 3 presents the proposed method and its implementation. Theoretical properties are derived in Section 4. Section 5 presents simulation studies to assess the performance of the proposed approach. In Section 6, we apply the proposed method to the IRI marketing data and Last.fm data. Concluding remarks and discussion are provided in Section 7.

2. Notation and Background

In this section, we introduce some notation and the background of the tensor and classical DRSs. Throughout this article, we use blackboard capital letters for sets, e.g., $\mathbb{T}, \mathbb{I}$, small letters for scalars, e.g., $x, y \in \mathbb{R}$, bold small letters for vectors, e.g., $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, bold capital letters for matrices, e.g., $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{n_1 \times n_2}$, and Euler script fonts for tensors, e.g., $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ $(d > 2)$.

A dth-order tensor is an array with d dimensions $(d > 2)$, which is an extension of a matrix to higher order. Here d represents the tensor's order. We denote the component $(i_1, i_2, \cdots, i_d)$ of a dth-order tensor $\mathcal{Y}$ by $y_{i_1 i_2 \cdots i_d}$, where $i_k = 1, 2, \ldots, n_k$, and k is called a mode of the tensor $(k = 1, 2, \ldots, d)$. In particular, a tensor $\mathcal{Y}$ is called a rank-one tensor if it can be written as $\mathcal{Y} = \mathbf{p}^1 \circ \mathbf{p}^2 \circ \cdots \circ \mathbf{p}^d$, where the symbol $\circ$ represents the vector outer product, and $\mathbf{p}^k = (p_1^k, p_2^k, \cdots, p_{n_k}^k)^\top$ is a n_k-dimensional latent factor corresponding to the kth mode. That is, each component of the tensor is the product of the corresponding vector components: $y_{i_1 i_2 \cdots i_d} = p_{i_1}^1 p_{i_2}^2 \cdots p_{i_d}^d$.

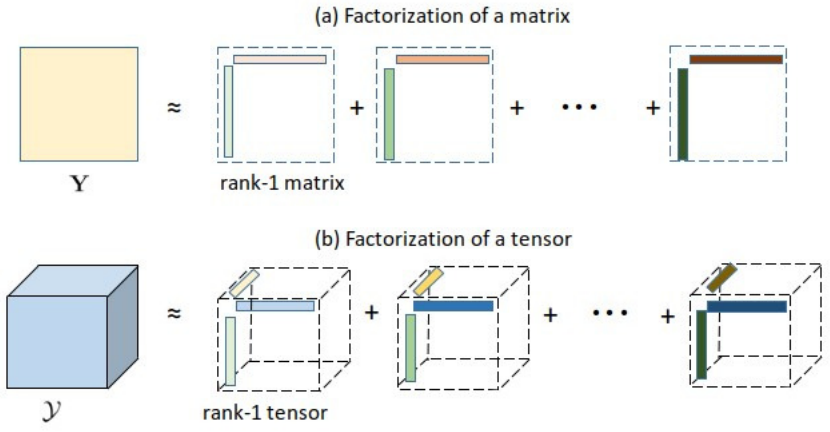

Figure 1: Illustration of factorizations of a matrix and a third-order tensor. (a) factorization of a matrix into r rank-1 matrices, (b) CPD of a third-order tensor into r rank-1 tensors.

The canonical polyadic decomposition (CPD) is commonly adopted in tensor decomposition, which decomposes a tensor as a sum of r rank-one tensors. That is:

$$\mathcal{Y} \approx \sum_{j=1}^{r} \mathbf{p}_{\cdot j}^{1} \circ \mathbf{p}_{\cdot j}^{2} \circ \cdots \circ \mathbf{p}_{\cdot j}^{d},$$

where $\mathbf{p}_{\cdot j}^{k} = (p_{1j}^{k}, \cdots, p_{n_k j}^{k})^{\top}$ is a n_k-dimensional latent factor corresponding to the kth mode for $k = 1, \ldots, d; j = 1, \ldots, r$. Equivalently, each component of $\mathcal{Y}$ is

$$y_{i_1 i_2 \cdots i_d} \approx \sum_{j=1}^{r} p_{i_1 j}^{1} p_{i_2 j}^{2} \cdots p_{i_d j}^{d}.$$

The CPD can be considered to be a higher-order generalization of matrix factorisation. Figure 1 illustrates a matrix factorization of a matrix and a CPD of a third-order tensor. An extensive review of tensors and other forms of tensor decomposition are discussed in Kolda and Bader (2009).

Let $\mathbf{P}^k = (\mathbf{p}_{\cdot 1}^{k}, \mathbf{p}_{\cdot 2}^{k}, \ldots, \mathbf{p}_{\cdot r}^{k})_{n_k \times r}$ and $\boldsymbol{\theta} = \{\mathbf{P}^1, \mathbf{P}^2, \cdots, \mathbf{P}^d\}$. We can estimate $\boldsymbol{\theta}$ via minimizing a loss function (e.g., L_2 loss). However, the non-convexity of the loss function could impose computational complexity due to numerical instability or even non-convergence (de Silva and Lim, 2008; Frolov and Oseledets, 2017). A common approach to alleviate the non-convexity problem is to introduce regularization. That is, an objective function with regularization as the following:

$$L(\boldsymbol{\theta}|\mathcal{Y}) = Q(\mathcal{Y}, \boldsymbol{\theta}) + J(\boldsymbol{\theta}),$$

where Q is a loss function and J is a penalty function, such as L_2, L_1 or L_0 penalties, or a fused Lasso.

Specially, the optimization problem solves $\boldsymbol{\theta}^* = \arg\min L(\boldsymbol{\theta}|\mathcal{Y})$, where $\boldsymbol{\theta}^*$ defines an optimal set of model parameters. In the case of squared loss function with an L_2-penalty, the objective function is

$$L(\boldsymbol{\theta}|\mathcal{Y}) = \sum_{(i_1, i_2, \ldots, i_d) \in \Omega} (y_{i_1 i_2 \cdots i_d} - \sum_{j=1}^{r} p_{i_1 j}^{1} p_{i_2 j}^{2} \cdots p_{i_d j}^{d})^2 + \lambda \sum_{k=1}^{d} \|\mathbf{P}^k\|_F^2,$$

where $\|\cdot\|_F$ represents the Frobenius norm, and $\Omega = \{(i_1, i_2, \ldots, i_d) : y_{i_1 i_2 \ldots i_d} \text{ is observed}\}$ is a set of indices corresponding to the observed components. Notice that, in the context of RS, the set Ω may not contain all indices of the tensor components and could be a small fraction of the entire tensor size, since the majority of the tensor components could be missing. Major algorithms for implementing the optimization problem include the cyclic coordinate descent algorithm, the stochastic gradient descent method and the maximum block improvement algorithm (Chen et al., 2012).

Following the tensor techniques, the classical DRSs can incorporate time as an additional mode of a tensor, that is, $\mathcal{Y} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d \times T}$, where the last mode is a time mode at fixed time points $\{t_1, t_2, \cdots, t_T\}$. The classical DRSs use CPD to obtain component estimators, that is,

$$y_{i_1 i_2 \cdots i_d t} \approx \sum_{j=1}^{r} p_{i_1 j}^{1} p_{i_2 j}^{2} \cdots p_{i_d j}^{d} q_{tj} \quad \text{for} \quad t = t_1, t_2, \cdots, t_T,$$

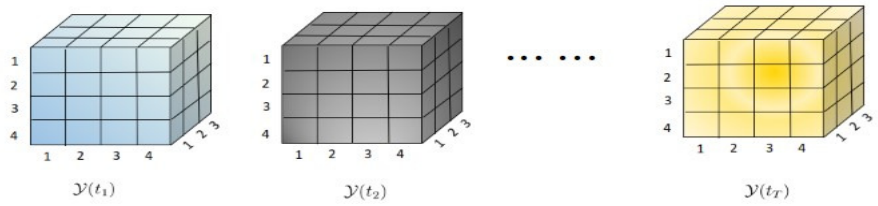

Figure 2: Third-order tensor-valued process.

where $\mathbf{q}_{\cdot j} = (q_{t_1 j}, \cdots, q_{t_T j})^\top$ is a T-dimensional latent factor corresponding to the time mode. However, the classical DRSs can only estimate the values $y_{i_1 i_2 \cdots i_d t}$ at fixed time points t. If one needs to estimate the values $y_{i_1 i_2 \cdots i_d t}$ for $t \in (t_i, t_{i+1})$, where $i \in \{1, 2, \cdots, T\}$, the classical DRSs are not applicable. Moreover, if one needs to forecast the values $y_{i_1 i_2 \cdots i_d t}$ for $t > t_T$, the classical DRSs need to extend the time mode to future time points. However, this involves an increasing number of parameters over time which could be computationally infeasible. In addition, the classical methods only focus on the estimations of the tensor components but do not provide statistical inference, e.g., the estimation of prediction intervals. In practice, providing the upper and lower bounds of predictions are also crucial in decision making. In the following, we pursue an alternative approach to solve this problem.

3. The Proposed Method

3.1 General Methodology

In this subsection, we develop the methodology for the proposed DTRS method. Specifically, we adopt the ideas of time-varying coefficient model framework to generalize the CPD to capture the trends of the DRS, and classify time points into subgroups to infer new time point trends through existing time points of the same group.

We consider a dth-order tensor-valued function $\mathcal{Y}(t) \in \mathbb{R}^{n_1 \times n_1 \times \ldots \times n_d}$, where the value at time t is a d-dimensional array. The tensor set $\mathbb{Y} = \{\mathcal{Y}(t) : t \in \mathbb{T}\}$ is the corresponding stochastic process defined on a compact interval $\mathbb{T}$. Without loss of generality, let $\mathbb{T}$ be a closed interval $[0, 1]$. Figure 2 illustrates an example of a tensor-valued process with $d = 3$. In the DRS, the tensor-valued process could be the rating or sale volume of items or products from users or stores given contexts. We assume that time points can be categorized into different subgroups, where time points of the same group have common information. For example, in our numerical studies, time points in the same month from the twelve months of each year are categorized in the same group. In addition to time, we also categorize subjects from other modes into subgroups if they share similar characteristics, for example, stores of the same market and products of the same product category.

Given the subgroup labels, we assume that each component of $\mathcal{Y}(t)$ can be estimated:

$$y_{i_1 i_2 \cdots i_d}(t) \approx \sum_{j=1}^{r} h_j(t) p^1_{i_1 j} p^2_{i_2 j} \cdots p^d_{i_d j} + g(t) q^1_{i_1} q^2_{i_2} \cdots q^d_{i_d}, \tag{1}$$

where $p^k_{i_k j}$ and $q^k_{i_k}$ are the jth latent factor and the subgroup factor for the i_kth subject from the kth mode, respectively. Here, $h_j(t)$ is a trend function of time for j, and $g(t) = \sum_{e=1}^{m_{d+1}} g_e(t) I(t \in s_e)$, where $I(t \in s_e)$ is an indicator function and assigns the interval s_e on

the eth subgroup, $g_e(t)$ is a trend function corresponding to the eth subgroup, and m_{d+1} is the number of subgroups for time. We have $q^k_{i_k} = q^k_{i'_k} = q^k_{(e_k)}$ if the i_kth and i'_kth subjects are from the e_kth subgroup ($e_k = 1, 2, \ldots, m_k$), where $q^k_{(e_k)}$ is the subgroup factor associated with the e_kth subgroup, and m_k is the number of subgroups for the kth mode. We denote the set of observed time points for the component $y_{i_1 i_2 \cdots i_d}(t)$ by $\mathbb{T}_{i_1 i_2 \ldots i_d}$, and the number of components of this set by $|\mathbb{T}_{i_1 i_2 \ldots i_d}|$. Let $\mathbf{y}_{i_1 i_2 \ldots i_d} = \{y_{i_1 i_2 \cdots i_d}(t)\}_{t \in \mathbb{T}_{i_1 i_2 \ldots i_d}}$. We assume that the covariance matrix is $\mathrm{cov}(\mathbf{y}_{i_1 i_2 \ldots i_d}) = \mathbf{\Sigma}^0_{i_1 i_2 \ldots i_d}$, typically not an identity matrix due to the intra-cluster correlation arising from repeated observed data.

Equation (1) adopts the idea of varying-coefficient models to create a CPD for tensor data. Varying-coefficient models are a useful tool to explore dynamic patterns, and have been applied to modeling and predicting longitudinal, functional and time series data (Huang and Shen, 2004; Fan and Zhang, 2008). Based on the varying-coefficient models, through the equations (1), we can obtain estimators of the component of tensor-value function at any time points in a continuous time interval (e.g., $t \in (a, b)$) instead of at fixed time points as in the DRS approaches (e.g., Xiong et al., 2010; Bi et al., 2018). The first part of equation (1) is an individual-level factor model which takes into account the heterogeneity of subjects and trend of time, and the time-varying coefficients $h_j(t)$ ($j = 1, \ldots, r$) reflect the dynamic features. The second part of equation (1) is a subgroup-level factor model to capture common features from the same subgroups, where the subgroup factors can accommodate new subjects from any mode at future time points, and the $g(t)$ allows time variables to follow a subgroup function of time such that we can predict future time points via borrowing information from existing time points of the same group.

To capture these trend functions, we adopt the polynomial splines to approximate $h_j(t)$ and $g_e(t)$. Let $\{\nu_{ji}\}_{i=1}^{a_N}$ be interior knots within $\mathbb{T}$, and Υ_j be a partition of $\mathbb{T}$ with a_N knots, that is $\Upsilon_j = \{0 = \nu_{j0} < \nu_{j1} < \cdots < \nu_{ja_N} < \nu_{ja_N+1} = 1\}$ for $j = 1, 2, \ldots, d$. The polynomial splines of an order $\kappa+1$ are functions with κ-degree of polynomials on intervals $[\nu_{ji-1}, \nu_{ji})$ for $i = 1, 2, \ldots, a_N$ and $[\nu_{ja_N}, \nu_{ja_N+1}]$, and have $\kappa - 1$ continuous derivatives globally. Denote a spline bases vector of the space of such spline functions as $\mathbf{B}_j(t) = (B_{j1}(t), \ldots, B_{jM}(t))^\top$, where $M = a_N + \kappa + 1$ as the number of spline bases. The function $h_j(t)$ ($j = 1, 2, \ldots, d$) can be approximated by

$$\hat{h}_j(t) \;\; = \;\; \textstyle\sum_{i=1}^{M} \alpha_{ji} B_{ji}(t) = \boldsymbol{\alpha}_j^\top \mathbf{B}_j(t),$$

where $\boldsymbol{\alpha}_j = (\alpha_{j1}, \alpha_{j2}, \ldots, \alpha_{jM})^\top$ is a coefficient vector. Spline functions can be B-spline or truncated polynomial functions. For example, for the truncated polynomial function, $\mathbf{B}_j(t) = (1, t, \ldots, t^\kappa, (t - \nu_{j1})_+^\kappa, \ldots, (t - \nu_{ja_N})_+^\kappa)^\top$, and the $(t - \nu)_+$ is $t - \nu$ if $t > \nu$ and 0 otherwise.

Similarly, let $\{\omega_{ei}\}_{i=1}^{a_N}$ be interior knots within $\mathbb{T}$, $\Gamma_e = \{0 = \omega_{e0} < \omega_{e1} < \cdots < \omega_{ea_N} < \omega_{ea_N+1} = 1\}$, and $\mathbf{A}_e(t) = (A_{e1}(t), \ldots, A_{eM}(t))^\top$ be a vector of spline bases for $e = 1, 2, \ldots, m_{d+1}$. The $g_e(t)$ can be approximated by

$$\hat{g}_e(t) = \sum_{i=1}^{M} \beta_{ei} A_{ei}(t) = \boldsymbol{\beta}_e^\top \mathbf{A}_e(t),$$

where $\boldsymbol{\beta}_e = (\beta_{e1}, \beta_{e2}, \ldots, \beta_{eM})^\top$. Based on equation (1), the prediction can be obtain as follows

$$\hat{y}_{i_1 i_2 \cdots i_d}(t) = \sum_{j=1}^{r} \hat{h}_j(t) p_{i_1 j}^1 p_{i_2 j}^2 \cdots p_{i_d j}^d + \hat{g}(t) q_{i_1}^1 q_{i_2}^2 \cdots q_{i_d}^d, \tag{2}$$

where $\hat{g}(t) = \sum_{e=1}^{m_{d+1}} \hat{g}_e(t) I(t \in s_e)$. The equation (2) can capture trends of the DRS sufficiently through the polynomial spline approximations of time-varying coefficient functions. In addition, since the spline approximation is computationally fast (Xue and Yang, 2006), the equation (2) can achieve the spline estimates of the coefficients efficiently, and this is especially advantageous in estimating high-dimensional parameters in RS. In contrast to these approaches like Xiong et al. (2010) and Bi et al. (2018), equation (2) can achieve forecasting at any future time points without requiring an increasing number of parameters over time. Note that the proposed method does not require the same number of knots and the same degree polynomial for either trend functions. In order to reduce the computational cost, we fixed the same numbers of knots and the same degree polynomial. We can also adopt different number of knots or different degree polynomial for different trend functions $g(t)$ and $h_j(t)$ respectively, or apply existing methods (Van Loock et al., 2011; Yuan et al., 2013; Dung and Tjahjowidodo, 2017) to identify the number of knots.

Due to the intra-cluster correlation, it is important to incorporate intra-cluster correlation into RS. However, in practice, the covariance matrix $\boldsymbol{\Sigma}^0_{i_1 i_2 \ldots i_d}$ is often unknown. We adopt an invertible working covariance matrix, denoted as $\boldsymbol{\Sigma}_{i_1 i_2 \ldots i_d}$, to take into account the intra-cluster correlation. Let $\mathbf{P} = (\mathbf{P}^{1\top}, \cdots, \mathbf{P}^{d\top})$, $\mathbf{q} = (\mathbf{q}^{(1)\top}, \cdots, \mathbf{q}^{(d)\top})^\top$, $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1^\top, \ldots, \boldsymbol{\alpha}_r^\top)^\top$, $\boldsymbol{\beta} = (\boldsymbol{\beta}_1^\top, \ldots, \boldsymbol{\beta}_{m_{d+1}}^\top)^\top$, and $\boldsymbol{\gamma} = (\boldsymbol{\alpha}^\top, \boldsymbol{\beta}^\top)^\top$, where $\mathbf{P}^k = (\mathbf{p}^k_{\cdot 1}, \ldots, \mathbf{p}^k_{\cdot r})$, $\mathbf{p}^k_{\cdot j} = (p^k_{1j}, \cdots, p^k_{n_k j})^\top$, $\mathbf{q}^{(k)} = (q^k_{(1)}, \ldots, q^k_{(m_k)})^\top$, and $k = 1, \ldots, d$. Define $\boldsymbol{\theta} = \{\mathbf{P}, \mathbf{q}, \boldsymbol{\gamma}\}$ as parameters of interest. Considering the intra-cluster correlation and non-convexity problem, we define the following weighted penalized objective function:

$$L(\boldsymbol{\theta}|\mathbb{Y}) = \sum_{(i_1, i_2, \cdots, i_d) \in \Omega} (\mathbf{y}_{i_1 i_2 \ldots i_d} - \widehat{\mathbf{y}}_{i_1 i_2 \ldots i_d})^\top \boldsymbol{\Sigma}^{-1}_{i_1 i_2 \ldots i_d} (\mathbf{y}_{i_1 i_2 \ldots i_d} - \widehat{\mathbf{y}}_{i_1 i_2 \ldots i_d}) + \lambda(\|\mathbf{P}\|_F^2 + \|\mathbf{q}\|_2^2 + \|\boldsymbol{\gamma}\|_2^2), \tag{3}$$

where λ is the penalized parameter, $\Omega = \{(i_1, i_2, \ldots, i_d) : y_{i_1 i_2 \cdots i_d}(t) \text{ is observed at some } t\}$, $\|\cdot\|_2$ is the Euclidean norm, and $\widehat{\mathbf{y}}_{i_1 i_2 \ldots i_d} = \{\hat{y}_{i_1 i_2 \cdots i_d}(t)\}_{t \in \mathbb{T}_{i_1 i_2 \ldots i_d}}$ is a $|\mathbb{T}_{i_1 i_2 \ldots i_d}| \times 1$ vector.

The matrix $\boldsymbol{\Sigma}_{i_1 i_2 \ldots i_d}$ is an approximation of the true covariance $\boldsymbol{\Sigma}^0_{i_1 i_2 \ldots i_d}$, and can be modeled as $\boldsymbol{\Sigma}_{i_1 i_2 \ldots i_d} = \mathbf{V}^{1/2}_{i_1 i_2 \ldots i_d} \mathbf{R}_{i_1 i_2 \ldots i_d} \mathbf{V}^{1/2}_{i_1 i_2 \ldots i_d}$, where $\mathbf{V}_{i_1 i_2 \ldots i_d}$ is a diagonal matrix of the marginal variance of $\mathbf{y}_{i_1 i_2 \ldots i_d}$, and $\mathbf{R}_{i_1 i_2 \ldots i_d}$ is a working correlation matrix for $\mathbf{y}_{i_1 i_2 \ldots i_d}$. Some commonly used working correlation structures include independence, exchangeable, and first-order autoregressive process (AR-1), among others. Given a working correlation structure, the working correlation matrix depends on fewer nuisance parameters which can be estimated by the residual-based moment method (Liang and Zeger, 1986). The proposed method is robust to the misspecification of correlation structure as indicated by our numerical examples.

3.2 Parameter Estimation

In this subsection, we discuss parameter estimation by minimizing (3). Let $\mathbf{p}^k_{i_k} = (p^k_{i_k 1}, \cdots, p^k_{i_k r})^\top$ and $\Omega^k_{i_k} = \{(i_1, \ldots, i_k, \ldots, i_d) : y_{i_1 \cdots i_k \cdots i_d}(t) \text{ is observed at some } t \text{ given } i_k\}$ be the

set of indices with the fixed kth mode index i_k, where the corresponding components are observed at some time points. We assume that the number of observations for each time subgroup s_e is larger or equal than 2 for $e = 1, \ldots, m_{d+1}$, and the number of observations for each subgroup e_k from the kth mode is larger or equal than 2 for $e_k = 1, \ldots, m_k; k = 1, \ldots, d$. The partial derivatives of the objective function (3) have explicit forms with respect to the individual factors, the subgroup factors and the spline coefficients, which makes it feasible to apply the blockwise coordinate descent approach (BCD). That is, for $i_k = 1, \ldots, n_k$ and $k = 1, \ldots, d$,

$$\widehat{\mathbf{p}}_{i_k}^k = \arg\min_{\mathbf{p}_{i_k}^k} \sum_{\Omega_{i_k}^k} (\mathbf{y}_{i_1 \cdots i_k \cdots i_d} - \widehat{\mathbf{y}}_{i_1 \cdots i_k \cdots i_d})^\top \boldsymbol{\Sigma}_{i_1 \cdots i_k \cdots i_d}^{-1} (\mathbf{y}_{i_1 \cdots i_k \cdots i_d} - \widehat{\mathbf{y}}_{i_1 \cdots i_k \cdots i_d}) + \lambda \|\mathbf{p}_{i_k}^k\|_2^2, \quad (4)$$

$$\widehat{\mathbf{q}}^{(k)} = \arg\min_{\mathbf{q}^{(k)}} \sum_{\Omega} (\mathbf{y}_{i_1 i_2 \cdots i_d} - \widehat{\mathbf{y}}_{i_1 i_2 \cdots i_d})^\top \boldsymbol{\Sigma}_{i_1 i_2 \cdots i_d}^{-1} (\mathbf{y}_{i_1 i_2 \cdots i_d} - \widehat{\mathbf{y}}_{i_1 i_2 \cdots i_d}) + \lambda \|\mathbf{q}^{(k)}\|_2^2, \quad (5)$$

$$\widehat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \sum_{\Omega} (\mathbf{y}_{i_1 i_2 \cdots i_d} - \widehat{\mathbf{y}}_{i_1 i_2 \cdots i_d})^\top \boldsymbol{\Sigma}_{i_1 i_2 \cdots i_d}^{-1} (\mathbf{y}_{i_1 i_2 \cdots i_d} - \widehat{\mathbf{y}}_{i_1 i_2 \cdots i_d}) + \lambda \|\boldsymbol{\alpha}\|_2^2, \quad (6)$$

$$\widehat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \sum_{\Omega} (\mathbf{y}_{i_1 i_2 \cdots i_d} - \widehat{\mathbf{y}}_{i_1 i_2 \cdots i_d})^\top \boldsymbol{\Sigma}_{i_1 i_2 \cdots i_d}^{-1} (\mathbf{y}_{i_1 i_2 \cdots i_d} - \widehat{\mathbf{y}}_{i_1 i_2 \cdots i_d}) + \lambda \|\boldsymbol{\beta}\|_2^2. \quad (7)$$

In fact, the estimation procedure of $\widehat{\mathbf{p}}_{i_k}^k$ in (4) is a ridge regression, and does not require knowing $\mathbf{p}_{i'_k}^k$ for $i'_k \neq i_k$. Thus, parallel computation is applicable to calculate $\widehat{\mathbf{p}}_1^k, \ldots, \widehat{\mathbf{p}}_{n_k-1}^k$ and $\widehat{\mathbf{p}}_{n_k}^k$ efficiently. The minimization of $L(\boldsymbol{\theta}|\mathbb{Y})$ can be done cyclically through estimating $\mathbf{P}$, $\mathbf{q}$, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. Notice that $\Omega = \cup_{i_k=1}^{n_k} \Omega_{i_k}^k$, and it is possible that $\Omega_{i_k}^k$ is empty for certain i_k's, that is, there is no observation on the subject i_k. Under this circumstance, the individual factor of the i_k subject is assigned as $\mathbf{p}_{i_k}^k = \mathbf{0}$, and the predicted values may degenerate to the subgroup-level factor model by utilizing information from members of the same subgroup.

3.3 Implementation

In the following, we discuss several implementation issues. To solve the objective function (3), we incorporate the maximum block improvement (MBI) strategy (Chen et al., 2012) into the BCD algorithm cyclically as in Bi et al. (2018). The MBI has two advantages over traditional cyclic BCD algorithms. First, it has a good algorithmic property which guarantees convergence to a stationary point, whereas traditional BCDs may end up with certain points where the criterion function ceases to decrease (Chen et al., 2012). Second, the MBI has the capability of choosing descending directions and hence has the possibility to discover "shortcuts", which may reduce the computational time significantly. Let $\widehat{\boldsymbol{\theta}}_l$ be an estimator of $\boldsymbol{\theta}$ at the lth iteration, $\boldsymbol{\theta}_a$ be a subset of $\boldsymbol{\theta}$, $\boldsymbol{\theta}^c$ be the complementary set of $\boldsymbol{\theta}_a$, and $\widehat{\boldsymbol{\theta}}_a^*$ be the attempted update of $\boldsymbol{\theta}_a$. The improvement of the $\widehat{\boldsymbol{\theta}}_a^*$ is defined as

$$J_{\hat{\theta}_a^*} = 1 - \frac{L(\widehat{\boldsymbol{\theta}}_a^*, \widehat{\boldsymbol{\theta}}_{l-1}^c|\mathbb{Y})}{L(\widehat{\boldsymbol{\theta}}_{l-1}|\mathbb{Y})}. \quad (8)$$

We summarize the implementation of the specifical algorithm as follows.

Algorithm Implementation Algorithm

1: (Initialization) Input all observed $y_{i_1 i_2 \cdots i_d}(t)$'s, the number of factors r, tuning parameter λ, initial value $\boldsymbol{\theta}_0$ and a stopping criterion $\varepsilon = 10^{-4}$.

2: (Individual factors update) At the lth iteration, estimate $\{\mathbf{P}^1, \mathbf{P}^2, \cdots, \mathbf{P}^d, \boldsymbol{\alpha}\}$.

(i) For each $\mathbf{P}^k$, solve (4) through parallel computing and obtain $\widehat{\mathbf{P}}^{k*}$. Then calculate $J_{\widehat{\mathbf{P}}^{k*}}$ through (8).

(ii) For $\boldsymbol{\alpha}$, solve (6) and obtain $\widehat{\boldsymbol{\alpha}}^*$. Then calculate $J_{\hat{\alpha}^*}$ through (8).

(iii) Assign
$\widehat{\mathbf{P}}_l^k \leftarrow \widehat{\mathbf{P}}^{k*}$, if $J_{\widehat{\mathbf{P}}^{k*}} = \max\{J_{\widehat{\mathbf{P}}^{1*}}, J_{\widehat{\mathbf{P}}^{2*}}, \cdots, J_{\widehat{\mathbf{P}}^{d*}}, J_{\hat{\alpha}^*}\}$.
$\widehat{\boldsymbol{\alpha}}_{(l)} \leftarrow \widehat{\boldsymbol{\alpha}}^*$, if $J_{\hat{\alpha}^*} = \max\{J_{\widehat{\mathbf{P}}^{1*}}, J_{\widehat{\mathbf{P}}^{2*}}, \cdots, J_{\widehat{\mathbf{P}}^{d*}}, J_{\hat{\alpha}^*}\}$.

3: (Subgroup factors update) At the lth iteration, estimate $\{\mathbf{q}^{(1)}, \mathbf{q}^{(2)}, \cdots, \mathbf{q}^{(d)}, \boldsymbol{\beta}\}$.

(i) For every $\mathbf{q}^{(k)}$, solve (5) and obtain $\widehat{\mathbf{q}}^{(k)*}$. Then calculate $J_{\widehat{\mathbf{q}}^{(k)*}}$ through (8).

(ii) For $\boldsymbol{\beta}$, solve (7) and obtain $\widehat{\boldsymbol{\beta}}^*$. Then calculate $J_{\hat{\beta}^*}$ through (8).

(iii) Assign
$\widehat{\mathbf{q}}_l^{(k)} \leftarrow \widehat{\mathbf{q}}^{(k)*}$, if $J_{\widehat{\mathbf{q}}^{(k)*}} = \max\{J_{\widehat{\mathbf{q}}^{(1)*}}, J_{\widehat{\mathbf{q}}^{(2)*}}, \cdots, J_{\widehat{\mathbf{q}}^{(d)*}}, J_{\hat{\beta}^*}\}$.
$\widehat{\boldsymbol{\beta}}_{(l)} \leftarrow \widehat{\boldsymbol{\beta}}^*$, if $J_{\hat{\beta}^*} = \max\{J_{\widehat{\mathbf{q}}^{(1)*}}, J_{\widehat{\mathbf{q}}^{(2)*}}, \cdots, J_{\widehat{\mathbf{q}}^{(d)*}}, J_{\hat{\beta}^*}\}$.

4: (Stopping Criterion) Stop if $\max\{J_{\widehat{\mathbf{P}}^{1*}}, J_{\widehat{\mathbf{P}}^{2*}}, \cdots, J_{\widehat{\mathbf{P}}^{d*}}, J_{\hat{\alpha}^*}, J_{\widehat{\mathbf{q}}^{(1)*}}, \cdots, J_{\widehat{\mathbf{q}}^{(d)*}}, J_{\hat{\beta}^*}\} < \varepsilon$. Set the final estimator $\widehat{\boldsymbol{\theta}} = \widehat{\boldsymbol{\theta}}_l$. Otherwise set $l \leftarrow l + 1$ and go to step 2.

To select tuning parameter λ, we search the one from grid points minimizing the root mean square error on the validation set, defined as $[\sum_{(i_1,\ldots,i_d,t)\in\Gamma}\{y_{i_1\ldots i_d}(t) - \hat{y}_{i_1\ldots i_d}(t)\}^2/|\Gamma|]^{1/2}$, where Γ is the set of indices and times of observed data. We choose the number of individual latent factors r such that it is sufficiently large and leads to stable estimation. In general, the r is no smaller than the theoretical rank of the tensor in order to represent subjects' latent features sufficiently well, but not so large as to over-burden the computational cost.

An appropriate selection of the knot sequence is important to efficiently implement the proposed method. In practice, knot locations are usually chosen to be equally-spaced over the range of data or placed at evenly-spaced quantiles of data. Since there are high-dimensional factor parameters, for simplicity we set the number of knots to be the integer part of $N^{1/(2\kappa+3)}$, where $N = |\Omega|$ and κ is the degree of polynomials. One can also choose other methods to select the number of knots such as the AIC or BIC procedures (Xue and Yang, 2006). The degree of polynomials κ is commonly chosen as $1, 2$, or 3. In our numerical study, we set $\kappa = 2$ and adopt truncated polynomial bases. One can also use different degrees and spline bases for different time-varying coefficients.

Another important issue is in selection of contextual variables as tensor modes. In practice, the chosen number of contexts is often pre-specified based on domain knowledge. A contextual variable can be considered an additional tensor mode of a higher-order tensor if users' and items' behaviors are distinctive under different values of the contextual variable.

On the one hand, a higher-order tensor with more contextual variables allows higher-order interactions and hence provides more accurate estimation. On the other hand, a higher-order tensor entails more complex and intensive computation, and may lead to overfitting. It is not suggested to assign too many contextual variables as additional tensor modes, which remains open to discussion regarding the number of contextual variables. In our numerical studies, promotion strategies are incorporated as a contextual variable, since users' and items' behaviors are distinctive under different promotion strategies. In general practice, however, we assume that the order of a tensor can be determined based on prior knowledge.

4. Theoretical Properties

In this section, we provide asymptotic properties for the proposed method and the estimation of prediction intervals. Specifically, we establish the convergence rate of the proposed tensor factorization and the asymptotic normality of the spline coefficient estimator. Following asymptotic normality, we can also construct the estimation of the prediction interval of the component. Note that identifiability is critical for tensor representation. We first present the sufficient conditions to ensure identifiability of the proposed tensor modeling as follows.

Proposition 1 *If $\sum_{k=1}^{d} K_k \geq 2r + d + 1$ holds, minimizers of $L(\mathbf{P}, \mathbf{q}, \boldsymbol{\alpha}, \boldsymbol{\beta}|\mathbb{Y})$ in $\mathbf{P}$, $\mathbf{q}$, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ given fixed spline bases are unique up to permutation almost surely, where K_k is the Kruskal rank of $(\mathbf{P}^k, \mathbf{q}^k)$, and $\mathbf{q}^k = (q_1^k, q_2^k, \cdots, q_{n_k}^k)^\top$.*

Proposition 1 shows that the proposed tensor modeling is identifiable up to permutation almost surely. To address permutation indeterminacy, we could align the factors according to a descending order of the first row of mode-1 factor matrix $\mathbf{P}^1$, that is, $p_{11}^1 \geq p_{12}^1 \geq \cdots \geq p_{1r}^1$, following the method in Zhang et al. (2014). The rearrangement can be implemented during or after the proposed algorithm, since it does not affect the estimation procedure. In the rest of Section 4, we assume that the parameters are identifiable.

Let $\boldsymbol{u}_{i_1 i_2 \ldots i_d} = \{(p_{i_1 1}^1 p_{i_2 1}^2 \cdots p_{i_d 1}^d), (p_{i_1 2}^1 p_{i_2 2,}^2 \cdots p_{i_d 2}^d), \cdots, (p_{i_1 r}^1 p_{i_2 r}^2 \cdots p_{i_d r}^d), (q_{i_1}^1 q_{i_2}^2 \cdots q_{i_d}^d)\}^\top$, $\mathcal{U} \in \mathbb{R}^{n_1 \times \ldots \times n_d \times (r+1)}$ consist of $\boldsymbol{u}_{i_1 i_2 \ldots i_d}$, $\mathbf{f}(t) = \{h_1(t), h_2(t), \ldots, h_r(t), g(t)\}^\top$, $\mathbf{F}_{i_1 i_2 \ldots i_d} \in \mathbb{R}^{|\mathbb{T}_{i_1 i_2 \cdots i_d}| \times (r+1)}$ be the matrix consisting of $\mathbf{f}(t)$ for all $t \in \mathbb{T}_{i_1 i_2 \cdots i_d}$. Considering random errors based on the equation (1), we denote $y_{i_1 i_2 \cdots i_d}(t)$ as $y_{i_1 i_2 \cdots i_d}(t) = \mathbf{f}(t)^\top \boldsymbol{u}_{i_1 i_2 \ldots i_d} + \varepsilon_{i_1 i_2 \ldots i_d}(t)$ for $t \in \mathbb{T}_{i_1 i_2 \cdots i_d}$, where $\varepsilon_{i_1 i_2 \ldots i_d}(t)$ is a random error with mean zero and finite variance. Let $\boldsymbol{\varepsilon}_{i_1 i_2 \ldots i_d} = \{\varepsilon_{i_1 i_2 \ldots i_d}(t)\}_{t \in \mathbb{T}_{i_1 i_2 \ldots i_d}}$ be a $|\mathbb{T}_{i_1 i_2 \ldots i_d}| \times 1$ vector. We have $\mathrm{cov}(\boldsymbol{\varepsilon}_{i_1 i_2 \ldots i_d}) = \mathrm{cov}(\mathbf{y}_{i_1 i_2 \ldots i_d}) = \boldsymbol{\Sigma}^0_{i_1 i_2 \ldots i_d}$. Thus, the corresponding vector form is

$$\mathbf{y}_{i_1 i_2 \ldots i_d} = \mathbf{F}_{i_1 i_2 \ldots i_d} \boldsymbol{u}_{i_1 i_2 \ldots i_d} + \boldsymbol{\varepsilon}_{i_1 i_2 \ldots i_d},$$

Let $J(\mathcal{U})$ be a non-negative penalty function of $\mathcal{U}$. The overall criterion given $h_j(\cdot)$ and $g(\cdot)$ is redefined as

$$L(\mathcal{U}|\mathbb{Y}) = \sum_{(i_1, i_2, \cdots, i_d) \in \Omega} (\mathbf{y}_{i_1 i_2 \ldots i_d} - \mathbf{F}_{i_1 i_2 \ldots i_d} \boldsymbol{u}_{i_1 i_2 \ldots i_d})^\top \boldsymbol{\Sigma}^{-1}_{i_1 i_2 \ldots i_d} (\mathbf{y}_{i_1 i_2 \ldots i_d} - \mathbf{F}_{i_1 i_2 \ldots i_d} \boldsymbol{u}_{i_1 i_2 \ldots i_d}) + \lambda J(\mathcal{U}) \tag{9}$$

for $\mathcal{U} \in \mathbb{S}$, where $\mathbb{S}$ is the parameter space for $\mathcal{U}$.

Based on the proposed method, $\widehat{\mathbf{y}}_{i_1i_2\ldots i_d}$ can be rewritten as $\widehat{\mathbf{y}}_{i_1i_2\ldots i_d} = \mathbf{W}_{i_1i_2\ldots i_d}\boldsymbol{\gamma}$, where $\mathbf{W}_{i_1i_2\ldots i_d} = (\mathbf{X}_{i_1i_2\ldots i_d1}, \cdots, \mathbf{X}_{i_1i_2\ldots i_dr}, \mathbf{Z}_{i_1i_2\ldots i_d1}, \cdots, \mathbf{Z}_{i_1i_2\ldots i_dm_{d+1}})$, $\mathbf{X}_{i_1i_2\ldots i_dj} = u_{i_1i_2\ldots i_dj}\mathbf{B}_{i_1i_2\ldots i_dj}$, $\mathbf{Z}_{i_1i_2\ldots i_de} = u_{i_1i_2\ldots i_d(r+1)}\mathbf{A}_{i_1i_2\ldots i_de}$, in which $\mathbf{B}_{i_1i_2\ldots i_dj} = \{\mathbf{B}_j(t)^\top\}_{t\in\mathbb{T}_{i_1i_2\ldots i_d}} \in \mathbb{R}^{|\mathbb{T}_{i_1i_2\cdots i_d}|\times M}$, and $\mathbf{A}_{i_1i_2\ldots i_de} = \{I(t\in s_e)\mathbf{A}_e(t)^\top\}_{t\in\mathbb{T}_{i_1i_2\ldots i_d}} \in \mathbb{R}^{|\mathbb{T}_{i_1i_2\cdots i_d}|\times M}$ for $j = 1, 2, \ldots, r$, $e = 1, 2, \ldots, m_{d+1}$. By the approximation theory (de Boor, 2001), there exists a constant $C > 0$, the spline functions $\widetilde{h}_j(t) = \boldsymbol{\alpha}_{0j}^\top\mathbf{B}_j(t)$ and $\widetilde{g}_e(t) = \boldsymbol{\beta}_{0e}^\top\mathbf{A}_e(t)$ such that $\sup_{t\in\mathbb{T}}|h_j(t) - \widetilde{h}_j(t)| \le Ca_N^{-\xi}$ and $\sup_{t\in\mathbb{T}}|g_e(t) - \widetilde{g}_e(t)| \le Ca_N^{-\xi}$ for any $j = 1, \ldots, r$, $e = 1, \ldots, m_{d+1}$. Denote $\boldsymbol{\gamma}_0 = (\boldsymbol{\alpha}_0^\top, \boldsymbol{\beta}_0^\top)^\top$, and let $N = |\Omega|$ be the number of components of the set Ω, $\lambda_{\min}\{\cdot\}$ and $\lambda_{\max}\{\cdot\}$ be the smallest and largest eigenvalues of any symmetric matrix, respectively. We require the following regularity conditions to establish the asymptotic properties.

(C1) The functions $h_j(\cdot)$ and $g_e(\cdot)$ are ξth-order continuously differential for some $\xi \ge 2$, all $j = 1, \ldots, d$, and $e = 1, \ldots, m_{d+1}$. The density function of design points t is absolutely continuous and bounded away from zero and infinity on a compact support $\mathbb{T}$.

(C2) The knots sequences Υ_j and Γ_e are quasi-uniform for $j = 1, \ldots, d$ and $e = 1, \ldots, m_{d+1}$; that is, there exists a constant $c > 0$, such that

$$\max_{j=1,\ldots,d}\frac{\max_{i=0,\ldots,a_N}(\nu_{ji+1} - \nu_{ji})}{\min_{i=0,\ldots,a_N}(\nu_{ji+1} - \nu_{ji})} \le c, \quad \text{and} \quad \max_{e=1,\ldots,m_{d+1}}\frac{\max_{i=0,\ldots,a_N}(\omega_{ei+1} - \omega_{ei})}{\min_{i=0,\ldots,a_N}(\omega_{ei+1} - \omega_{ei})} \le c.$$

(C3) There exist positive constants σ_1^2 and σ_2^2 such that the covariance matrix $\boldsymbol{\Sigma}_{i_1i_2\ldots i_d}^0$ of random error $\boldsymbol{\varepsilon}_{i_1\ldots i_d}$ satisfies that $\sigma_1^2 \le \lambda_{\min}\{\boldsymbol{\Sigma}_{i_1i_2\ldots i_d}^0\} \le \lambda_{\max}\{\boldsymbol{\Sigma}_{i_1i_2\ldots i_d}^0\} \le \sigma_2^2$.

(C4) There exist some positive constants c_1 and c_2 such that $c_1 \le \lambda_{\min}\{\boldsymbol{\Sigma}_{i_1i_2\ldots i_d}^{-1}\boldsymbol{\Sigma}_{i_1i_2\ldots i_d}^0\} \le \lambda_{\max}\{\boldsymbol{\Sigma}_{i_1i_2\ldots i_d}^{-1}\boldsymbol{\Sigma}_{i_1i_2\ldots i_d}^0\} \le c_2$.

(C5) $T_{\max} = \max_{(i_1,\cdots,i_d)\in\Omega}\{|\mathbb{T}_{i_1\cdots i_d}|\} = o_p(N^\tau)$, $T_{\min} = \min_{(i_1,\cdots,i_d)\in\Omega}\{|\mathbb{T}_{i_1\cdots i_d}|\} = o_p(N^\upsilon)$ for $0 \le \tau/2 < \upsilon \le \tau < 1$, and $\lambda = o_p(1)$.

Conditions (C1)-(C3) are standard in the polynomial spline framework. Similar conditions are also presented in Huang (2003) and Claeskens et al. (2009). In particular, condition (C1) imposes a smoothness condition of trend functions and a mild condition on time density, and guarantees that the observation time points are randomly scattered. Condition (C2) indicates that the adjacent distances among the knot sequence are comparable. Condition (C3) implies that the eigenvalues of random errors are bounded. Condition (C4) implies that the difference between the working covariance and true covariance matrices is bounded. Condition (C5) implies that the number of the observed time points grows as the number of the observed components of the tensor increases, to ensure the convergence of the proposed tensor factorization. The following theorem establishes the convergence rate for the proposed tensor factorization.

Theorem 2 *Under conditions (C1)-(C5), if the penalty function $J(\mathcal{U})$ has bounded first and second derivatives at true parameter $\mathcal{U}_0$, as $N \to \infty$, on a δ-ball centered at $\mathcal{U}_0$ for some $\delta > 0$, there exists a minimizer $\widehat{\mathcal{U}}$ of (9) such that*

$$\sum_{(i_1,i_2,\cdots,i_d)\in\Omega}\|\mathbf{F}_{i_1i_2\cdots i_d}(\widehat{\mathbf{u}}_{i_1i_2\cdots i_d} - \mathbf{u}_{0i_1i_2\cdots i_d})\|_2^2/N = O_p(N^{-1+2(\tau-\upsilon)}).$$

Theorem 2 provides the convergence rate of the proposed method given trend functions. When $\tau = \upsilon$, that is, $T_{\max}$ and $T_{\min}$ have the same order, the convergence rate of the estimator $\widehat{\mathcal{U}}$ reaches the optimal rate $N^{-1/2}$. Meanwhile, if the order of $T_{\max}$ is $\sqrt{N}$ faster than that of $T_{\min}$, that is, $\tau - \upsilon = 0.5$, then $\widehat{\mathcal{U}}$ will not converge to the true $\mathcal{U}_0$. This implies that to guarantee consistency of the tensor factorization, one should collect sufficient observations even for the least popular user-item-context combinations. In the following theorem, we establish the asymptotic property of the spline coefficient estimator.

Theorem 3 *Under conditions (C1)-(C5), if* $\lim_{N\to\infty} a_N \log a_N / N = 0$ *and* $\lim_{N\to\infty} a_N^{-\xi} N^{\tau} = 0$, *then for any vector* $\mathbf{c}$ *whose components are not all zero, the parametric estimator* $\widehat{\boldsymbol{\gamma}}$ *by (6) and (7) satisfies*

$$\mathbf{c}^\top(\widehat{\boldsymbol{\gamma}} - \boldsymbol{\gamma}_0) var\{\mathbf{c}^\top(\widehat{\boldsymbol{\gamma}} - \boldsymbol{\gamma}_0)\}^{-1/2} \xrightarrow{\mathrm{L}} N(0,1),$$

where $var\{\mathbf{c}^\top(\widehat{\boldsymbol{\gamma}} - \boldsymbol{\gamma}_0)\} = \mathbf{c}^\top \boldsymbol{\Psi}^{-1}\boldsymbol{\Phi}\boldsymbol{\Psi}^{-1}\mathbf{c} = O_p(a_N N^{-1+\tau-2\upsilon})$, $\boldsymbol{\Psi} = \sum_{(i_1,\ldots,i_d)\in\Omega} \mathbf{W}_{i_1\ldots i_d}^\top \boldsymbol{\Sigma}_{i_1\ldots i_d}^{-1}$. $\mathbf{W}_{i_1\ldots i_d}$, *and* $\boldsymbol{\Phi} = \sum_{(i_1,\ldots,i_d)\in\Omega} \mathbf{W}_{i_1\ldots i_d}^\top \boldsymbol{\Sigma}_{i_1\ldots i_d}^{-1} \boldsymbol{\Sigma}_{i_1\ldots i_d}^{0} \boldsymbol{\Sigma}_{i_1\ldots i_d}^{-1} \mathbf{W}_{i_1\ldots i_d}$.

Theorem 3 establishes the asymptotic normality of the spline coefficient estimator. The convergence rate of the spline coefficient estimator is $O_p(a_N N^{-1+\tau-2\upsilon})$. If $T_{\max}$ and $T_{\min}$ have the same order, $var\{\mathbf{c}^\top(\widehat{\boldsymbol{\gamma}} - \boldsymbol{\gamma}_0)\} = O_p(a_N/N^{1+\upsilon})$, and similar results can be found in Huang et al. (2004). The asymptotic variance in Theorem 3 depends on the working covariance matrix and the true covariance matrix. When the working covariance matrices are equal to the true covariance matrices, the asymptotic variance of the proposed estimator reaches the minimum in the sense of Lower order and the proposed estimator is asymptotic efficient.

More importantly, the result of Theorem 3 is the key foundation for constructing prediction intervals. First, we derive the standard error for the spline parametric estimates given a fixed λ using the sandwich covariance formula $\widehat{\mathrm{Cov}}(\widehat{\boldsymbol{\gamma}}) = (\widehat{\boldsymbol{\Psi}}+\lambda\mathbf{I})^{-1}\widehat{\boldsymbol{\Phi}}(\widehat{\boldsymbol{\Psi}}+\lambda\mathbf{I})^{-1}$, where $\widehat{\boldsymbol{\Psi}} = \sum_{(i_1,i_2,\ldots,i_d)\in\Omega} \widehat{\mathbf{W}}_{i_1i_2\ldots i_d}^\top \boldsymbol{\Sigma}_{i_1i_2\ldots i_d}^{-1} \widehat{\mathbf{W}}_{i_1i_2\ldots i_d}$, $\widehat{\boldsymbol{\Phi}} = \sum_{(i_1,i_2,\ldots,i_d)\in\Omega} \{\widehat{\mathbf{W}}_{i_1i_2\ldots i_d}^\top \boldsymbol{\Sigma}_{i_1i_2\ldots i_d}^{-1} (\mathbf{y}_{i_1i_2\ldots i_d} - \widehat{\mathbf{W}}_{i_1i_2\ldots i_d}\widehat{\boldsymbol{\gamma}})\}^{\otimes 2}$, $\otimes$ operation is the vector operation $\boldsymbol{a}^{\otimes 2} = \boldsymbol{a}\boldsymbol{a}^\top$, and $\mathbf{I}$ is an identity matrix. Since $\hat{y}_{i_1i_2\ldots i_d}(t) = \widehat{\mathbf{w}}_{i_1i_2\ldots i_d t}^\top \widehat{\boldsymbol{\gamma}}$, and $\widehat{\mathbf{w}}_{i_1i_2\ldots i_d t}$ is the tth column of estimator $\widehat{\mathbf{W}}_{i_1i_2\ldots i_d}^\top$, a $100(1-\sigma)\%$ prediction interval (Chatfield, 1993) of $\hat{y}_{i_1i_2\ldots i_d}(t)$ is

$$\hat{y}_{i_1i_2\ldots i_d}(t) \pm \phi_{\sigma/2}\sqrt{\mathrm{var}\{e_{i_1i_2\ldots i_d}(t)\}}, \tag{10}$$

where $\phi_{\sigma/2}$ is the $100(1-\sigma)$th percentile of the standard normal distribution, and the $\mathrm{var}\{e_{i_1i_2\ldots i_d}(t)\}$ is the variance of the prediction error and can be estimated as:

$$\widehat{\mathrm{var}}\{e_{i_1i_2\ldots i_d}(t)\} = \widehat{\mathbf{w}}_{i_1i_2\ldots i_d t}^\top \widehat{\mathrm{Cov}}(\widehat{\boldsymbol{\gamma}}) \widehat{\mathbf{w}}_{i_1i_2\ldots i_d t} + \widehat{\mathrm{var}}\{\varepsilon_{i_1i_2\ldots i_d}(t)\}. \tag{11}$$

The first term in equation (11) is due to estimation error, and the second term can be estimated by the mean squared error on training data.

5. Simulation Studies

In this section, we perform simulation studies to compare the proposed method (DTRS) with competing methods, including Bayesian probabilistic tensor factorization (BPTF, Xiong et al., 2010) and the recommendation engine of multilayers (REM, Bi et al., 2018). We assess forecasting performance via examining the root mean square error (RMSE) and the mean absolute error (MAE), where the RMSE is defined as $[\sum_{(i_1,\ldots,i_d,t)\in\Gamma}\{y_{i_1\ldots i_d}(t) - \hat{y}_{i_1\ldots i_d}(t)\}^2/|\Gamma|]^{1/2}$, the MAE is defined as $\sum_{(i_1,\ldots,i_d,t)\in\Gamma}|y_{i_1\ldots i_d}(t) - \hat{y}_{i_1\ldots i_d}(t)|/|\Gamma|$, and Γ is the set of indices and time of observed data. Moreover, we evaluate the coverage probability of the prediction interval estimated by the proposed method with 95% nominal coverage probability (PICP) .

5.1 Simple Tensor Function

In the simulation, we consider a third-order tensor function of time with user, context and item modes. We set the numbers of users, contexts and items to $n_1 = 100$, $n_2 = 9$, and $n_3 = 100$, respectively. We assume that users, contexts, items and time points are from $m_1 = 10$, $m_2 = 3$, $m_3 = 10$ and $m_4 = 4$ subgroups, respectively. Users, contexts, items and time points are evenly assigned to each subgroup. The number of latent factors is set as $r = 3$. We generate tensor functions at time points $t \sim U(0, 1)$ by generating its components as $y_{i_1i_2i_3}(t) = \sum_{j=1}^{r} h_j(t)p^1_{i_1j}p^2_{i_2j}p^3_{i_3j} + g(t)q^1_{i_1}q^2_{i_2}q^3_{i_3} + \varepsilon_{i_1i_2i_3}(t)$ for $i_k = 1, \ldots, n_k$, $k = 1, 2, 3$, where the latent factors $\mathbf{p}^k_{i_k} \sim N(0, \mathrm{I}_r)$, trend functions $h_1(t) = \sin(0.3\pi t)$, $h_2(t) = 8t(1 - t) - 1$ and $h_3(t) = \cos(0.2\pi t) + 1$. To distinguish different subgroups, we set the subgroup factors as a simple sequence, where $\mathbf{q}^1_{(e_1)} = -1+0.4e_1$, $\mathbf{q}^2_{(e_2)} = -1.2+0.6e_2$ and $\mathbf{q}^3_{(e_3)} = -0.4+0.2e_3$ for $e_k = 1, \ldots, m_k$ and $k = 1, 2, 3$. The function $g(t) = \sum_{e=1}^{m_4} g_e(t)I(t \in s_e)$, where $g_1(t) = 2t - 1$, $g_2(t) = 8(t - 0.5)^3$, $g_3(t) = \sin(0.1\pi t) + \cos(\pi t)$, and $g_4(t) = -5\exp(t)+10$. The error $\boldsymbol{\varepsilon}_{i_1i_2i_3} = (\varepsilon_{i_1i_2i_3}(t_1), \ldots, \varepsilon_{i_1i_2i_3}(t_T))^\top$ follows a multivariate normal distribution with mean 0 and a common marginal variance 1, and the correlation structure is either independence or AR-1 with correlation $\rho = 0.85$.

In each simulation, we consider the number of time points as $T = T_1 + T_2$, where the tensor data in the first $T_1 = 12$ time points are set as the training data, and the tensor data in the last T_2 time points are used as the testing data. For evaluating the forecasting performance at future time points, we consider $T_2 = 8$ or 12. Considering the missing case, we generate $n_1n_2n_3T(1 - \pi_m)$ components out of the tensor functions, where π_m is the missing percentage and set as 80%. Furthermore, we use $\pi_{cs} = 30\%$ to represent the proportion of new items in the testing data unavailable from the training set. To illustrate the effect of incorporating intra-cluster correlation on estimation efficiency, we compare the estimation efficiency of the proposed methods using different working correlation structures: independent or AR-1, denoted as DTRSin and DTRSar, respectively.

According to Xiong et al. (2010) and Bi et al. (2018), BPTF and REM methods model fourth-order tensor with user, context, item and time modes. For all methods, we assume that the subgroup structure and the number of latent factors are known. For REM and the proposed methods, the tuning parameter λ is pre-selected from grid points ranging from 0 to 20. The validation set is the data from the last four time points of the training set. For BPTF, we keep the remaining parameters by their default choices and obtain a forecast

Table 1: Average RMSE and MAE of all approaches. The PICP is the average coverage probability of the 95% prediction interval. The RMSE, MAE and PICP are provided with standard error based on 100 simulations in each parenthesis.

True structure:		Independent		AR	
Method		$T_2 = 8$	$T_2 = 12$	$T_2 = 8$	$T_2 = 12$
DTRSin	RMSE	**1.570**(0.196)	**1.660**(0.389)	1.597(0.192)	1.707(0.524)
	MAE	**1.092**(0.091)	**1.132**(0.160)	1.115(0.091)	1.160(0.208)
	PICP	**0.949**(0.015)	**0.953**(0.017)	0.946(0.017)	0.952(0.018)
DTRSar	RMSE	1.625(0.244)	1.696(0.286)	**1.576**(0.190)	**1.632**(0.200)
	MAE	1.133(0.118)	1.170(0.159)	**1.099**(0.085)	**1.130**(0.102)
	PICP	0.943(0.019)	0.947(0.021)	**0.947**(0.015)	**0.949**(0.018)
REM	RMSE	2.502(0.322)	2.494(0.307)	2.498(0.304)	2.494(0.305)
	MAE	1.654(0.178)	1.640(0.170)	1.650(0.166)	1.643(0.172)
	PICP	–	–	–	–
$\text{BPTF}_{\text{bayes}}$	RMSE	2.675(0.742)	2.930(0.965)	2.724(0.863)	3.181(1.148)
	MAE	1.810(0.427)	1.958(0.547)	1.826(0.495)	2.104(0.654)
	PICP	–	–	–	–
$\text{BPTF}_{\text{basic}}$	RMSE	2.142(0.221)	2.145(0.211)	2.136(0.222)	2.144(0.206)
	MAE	1.446(0.116)	1.454(0.115)	1.441(0.117)	1.453(0.111)
	PICP	–	–	–	–
$\text{BPTF}_{\text{double}}$	RMSE	2.388(0.319)	2.665(0.356)	2.405(0.319)	2.642(0.380)
	MAE	1.598(0.190)	1.774(0.217)	1.611(0.191)	1.755(0.232)
	PICP	–	–	–	–

via sampling the factor matrix of time from the time posterior distribution, denoted as $\text{BPTF}_{\text{bayes}}$. Following the forecasting technique of Araujo et al. (2019), we also consider BPTF incorporating basic exponential smoothing (Holt's method, Holt, 2004) and double exponential smoothing (Holt-Winters method, Holt, 2004; Winters, 1960). That is, we first use BPTF to estimate the factor matrices of user, item, context and time in the training data, and then forecast the factor matrix of time at given time points of the testing data via basic exponential smoothing or double exponential smoothing, denoted as $\text{BPTF}_{\text{basic}}$ and $\text{BPTF}_{\text{double}}$, respectively. All methods are replicated by 100 simulation runs.

Table 1 provides the estimation results of all methods. We observe that the proposed method has better performance when the working correlation structure is the same as the true correlation structure. When the true correlation structure is independence, the DTRSin has smaller RMSE and MAE than the DTRSar, with more than 2.17% improvement. Similarly, when the true correlation structure is AR-1, the DTRSar outperforms the DTRSin. Moreover, the PICPs of the DTRS method are close to 0.95, which implies that the proposed method provides accurate prediction intervals, whereas the competing methods are not able to provide such prediction intervals. For the performance of forecasting, we observe that the DTRSin and DTRSar outperform other methods across all settings. Specifically, both DTRS methods improve the RMSE and MAE of the REM by more than 40%, and

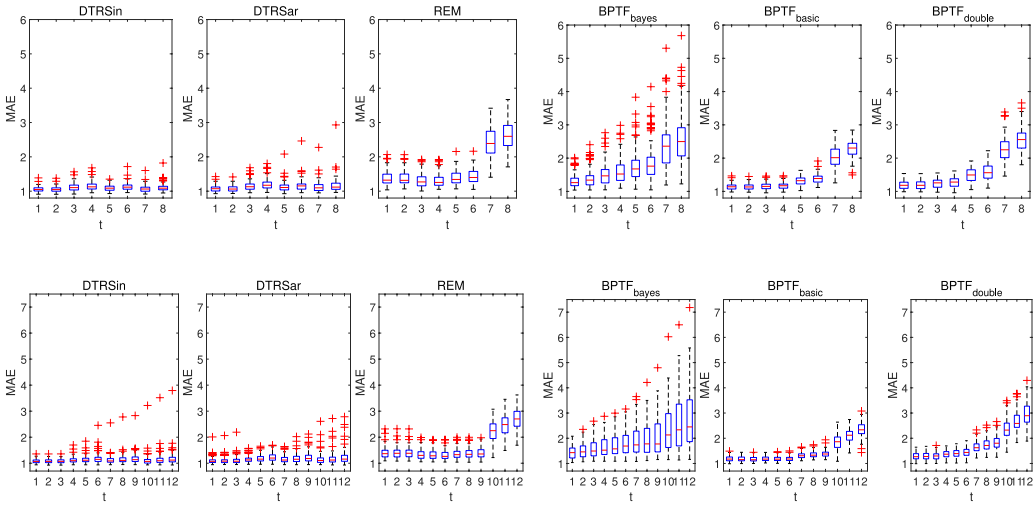

Figure 3: Box plots of the MAE for forecasting values with 8 and 12 time points and the true independent correlation.

those of the $\text{BPTF}_{\text{bayes}}$, $\text{BPTF}_{\text{basic}}$ and $\text{BPTF}_{\text{double}}$ by more than 60%, 24%, and 41%, respectively. In this setting, the $\text{BPTF}_{\text{basic}}$ performs better than the $\text{BPTF}_{\text{double}}$. This is probably because the basic exponential smoothing method is more applicable in forecasting time series with no clear trend or seasonal pattern, whereas the double exponential smoothing method performs better when a trend is present (Holt, 2004). Although the $\text{BPTF}_{\text{basic}}$ and $\text{BPTF}_{\text{double}}$ perform better than the $\text{BPTF}_{\text{bayes}}$, the proposed method is still able to beat the best of the BPTF variations. This indicates that the proposed method provides more accurate forecasting compared to other methods.

To illustrate the specific performance for forecasting at each time point, we calculate the MAE at each time point and provide box plots for the MAE in Figures 3-4. We observe that the performance of the proposed method is relatively robust against time in all settings. The MAEs of both DTRS methods at any time point are the lowest. The proposed method outperforms other methods, especially for long-term forecasting, indicating that it can handle both short-term and long-term forecasting accurately.

5.2 Long Forecasting Time Period

In this simulation study, we evaluate the performance of the proposed method with a vast set of users and items and time period. Moreover, we also report the average computational time (ComTime) in seconds for each method based on 50 repetitions. All experiments are implemented using Window 10 with 1.99 GHz Intel Core i7 Processor and 16 GB memory.

We consider a third-order tensor function of time with user, context and item modes. We set the numbers of users, contexts and items to $n_1 = 1000$, $n_2 = 30$, $n_3 = 10000$, respectively. The number of time points is $T = T_1 + T_2$ with the number of the training time points $T_1 = 80$ and the number of the testing time points $T_2 = 8, 12, 24, 32$. Other setting is similar to the setting in Section 5.1. We consider that the error $\boldsymbol{\varepsilon}_{i_1 i_2 i_3} = (\varepsilon_{i_1 i_2 i_3}(t_1), \ldots, \varepsilon_{i_1 i_2 i_3}(t_T))^\top$

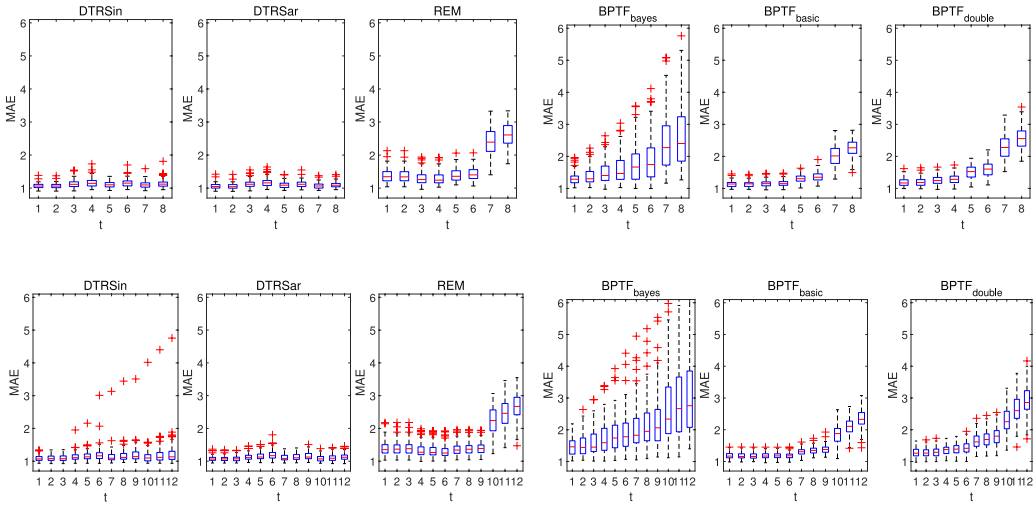

Figure 4: Box plots of the MAE for forecasting values with 8 and 12 time points and the true AR-1 correlation.

follows a multivariate normal distribution with mean 0 and the independent structure. We perform the proposed method and the compering methods as in Section 5.1. The simulation results are summarized in Table 2.

Table 2 shows that the proposed method outperforms other methods across all settings. Specifically, both DTRS methods improve the RMSE and MAE of the REM by more than 17%, and those of the BPTF$_{bayes}$, BPTF$_{basic}$ and BPTF$_{double}$ by more than 39%, 31% and 39%, respectively. Moreover, the proposed method is robust against longer forecasting time periods comparing with other methods. Specifically, we observe that the REM is more efficient computationally than the proposed method but the accuracy of the REM is lower than the proposed method. This is probably due to that the REM treats time as a mode of tensor, with more parameters involved as the time increases, and leads to more errors of estimation. Due to the demand of Gibbs sampling, the BPTF$_{bayes}$, BPTF$_{basic}$ and BPTF$_{double}$ require longer computational time and larger menory storage. In additional, as the time increases, the BPTF$_{bayes}$ needs to involve more parameters in time factors which leads to low computational efficiency and low accuracy in forecasting. Although the BPTF$_{basic}$ and BPTF$_{double}$ do not have increasing number of parameters as the time increases due to the exponential smoothing method, the Gibbs sampling is still time-consuming. Reducing the number of Gibbs samples may lead to less computational time but with less accurate predictions. For our method, the DTRSar requires to estimate the covariance matrix, and the DTRSar requires more computational time than the DTRSin. Nevertheless, the DTRSar has higher accuracy than other competing methods in terms of RMSE and MAE. Overall, the proposed method performs well in term of computational efficiency and forecasting accuracy.

Table 2: Average RMSE and MAE of all approaches for different forecasting time horizons. The PICP and ComTime are the average coverage probability of the 95% prediction interval and average computational time in seconds, respectively. The RMSE, MAE and PICP are provided with standard error based on 50 simulations in each parenthesis.

Method		$T_2 = 8$	$T_2 = 12$	$T_2 = 24$	$T_2 = 32$
DTRSin	RMSE	**1.549**(0.244)	**1.574**(0.325)	**1.633**(0.407)	**1.682**(0.458)
	MAE	1.088(0.148)	1.107(0.173)	1.147(0.218)	1.178(0.247)
	PICP	0.930(0.024)	0.932(0.026)	0.935(0.029)	0.941(0.026)
	ComTime	634.5s	634.7s	635.4s	635.9s
DTRSar	RMSE	1.574(0.343)	1.597(0.377)	1.663(0.510)	1.727(0.675)
	MAE	1.106(0.185)	1.121(0.208)	1.164(0.274)	1.202(0.343)
	PICP	0.929(0.020)	0.931(0.023)	0.933(0.031)	0.938(0.028)
	ComTime	766.5s	766.8s	767.6s	768.1s
REM	RMSE	1.855(0.410)	2.127(0.560)	2.268(0.469)	2.041(0.640)
	MAE	1.321(0.245)	1.459(0.303)	1.528(0.280)	1.432(0.355)
	PICP	–	–	–	–
	ComTime	159.1s	168.5s	184.5s	163.5s
$\text{BPTF}_{\text{bayes}}$	RMSE	2.331(0.772)	2.332(0.753)	2.323(0.680)	2.618(0.617)
	MAE	1.634(0.453)	1.629(0.439)	1.618(0.387)	1.793(0.369)
	PICP	–	–	–	–
	ComTime	772.0s	777.2s	792.5s	802.4s
$\text{BPTF}_{\text{basic}}$	RMSE	2.104(0.658)	2.106(0.671)	2.198(0.717)	2.641(0.791)
	MAE	1.503(0.448)	1.503(0.454)	1.558(0.471)	1.789(0.495)
	PICP	–	–	–	–
	ComTime	756.1s	761.0s	775.5s	784.1s
$\text{BPTF}_{\text{double}}$	RMSE	2.203(0.704)	2.224(0.734)	2.367(0.834)	2.803(0.977)
	MAE	1.562(0.470)	1.572(0.485)	1.651(0.530)	1.879(0.581)
	PICP	–	–	–	–
	ComTime	738.8s	743.8s	759.5s	769.3s

Table 3: Average RMSE and MAE of the proposed method DTRSin under $0\%, 10\%$ and 30% cluster misspecification rate (Mis. rate). The PICP is the average coverage probability of the 95% prediction interval. The RMSE, MAE and PICP are provided with standard error based on 50 simulations in each parenthesis.

Method	Mis. rate		$T_2 = 8$	$T_2 = 12$	$T_2 = 24$	$T_2 = 32$
DTRSin	0%	RMSE	1.549(0.244)	1.574(0.325)	1.633(0.407)	1.682(0.458)
		MAE	1.088(0.148)	1.107(0.173)	1.147(0.218)	1.178(0.247)
		PICP	0.930(0.024)	0.932(0.026)	0.935(0.029)	0.941(0.026)
	10%	RMSE	1.594(0.350)	1.612(0.357)	1.658(0.395)	1.703(0.433)
		MAE	1.109(0.184)	1.124(0.194)	1.159(0.222)	1.190(0.247)
		PICP	0.935(0.019)	0.938(0.020)	0.942(0.024)	0.947(0.022)
	30%	RMSE	1.604(0.386)	1.617(0.387)	1.697(0.502)	1.771(0.609)
		MAE	1.120(0.221)	1.129(0.221)	1.179(0.278)	1.226(0.330)
		PICP	0.935(0.021)	0.939(0.021)	0.943(0.024)	0.947(0.023)
REM	0%	RMSE	1.855(0.410)	2.127(0.560)	2.268(0.469)	2.041(0.640)
		MAE	1.321(0.245)	1.459(0.303)	1.528(0.280)	1.432(0.355)
	10%	RMSE	2.443(0.405)	2.390(0.384)	2.401(0.574)	2.383(0.432)
		MAE	1.661(0.243)	1.619(0.211)	1.612(0.344)	1.617(0.259)
	30%	RMSE	2.443(0.437)	2.337(0.314)	2.360(0.392)	2.268(0.382)
		MAE	1.676(0.294)	1.597(0.193)	1.595(0.245)	1.536(0.219)

5.3 Robustness under Cluster Misspecification

In this simulation study, we study the robustness of the proposed method when the clusters are misspecified.

We follow the same data-generating process as in Section 5.2, but allow the cluster assignment to be misspecified. Specifically, we misassign users, contexts and items to adjacent clusters with $0\%, 10\%$ and 30% chance. The Adjacent clusters are defined as the clusters with the closest group effects. This definition of adjacent clusters reflects the real-data situation (e.g., a facial tissue might be misassigned as paper towels, but less likely to be misassigned as yogurt). We also compare the proposed method with the REM method which also consider the subgroup information.

The simulation results based on 50 replications are summarized in Table 3. Table 3 shows that in general the proposed method is robust against the misspecification of cluster. In comparison with the results when 0% of the cluster members are misclassified, the proposed method is more robust than the REM method in all settings under 10% and 30% cluster misspecification rate. For example, the proposed method under the 10% misspecification rate is 2.9% worse than the proposed method without misspecification in terms of the RMSE under $T_2 = 8$; and is 3.6% worse than one without misspecification under $T_2 = 8$. However, the REM method under the 10% and 30% misspecification rates is 31.7% worse than the REM method without misspecification in terms of the RMSE under $T_2 = 8$.

6. Empirical Examples

6.1 IRI Marketing Data

In this section, we focus on sales data at drug stores from the IRI Marketing Data (Bronnenberg et al., 2008) to illustrate the performance of the proposed method. The original IRI data is an immense collection of consumer panel data and store sales at grocery stores, drug stores and mass-market stores over the years 2001-2011. The store sales data contain weekly product sales volumes, pricing, and promotion data for all items from 31 product categories sold in 50 U.S. markets. These markets are geographic units defined typically as an agglomeration of counties, usually covering a major metropolitan areas (e.g., Chicago, IL) but sometimes covering just part of a region (e.g., New England). A detailed description of an early version of the data is available in Bronnenberg et al. (2008).

To illustrate the proposed method, we choose sales data at drug stores collected from 2001 to 2011, where there are sales volume records, recorded times, promotion strategies, 43,631 product IDs, and 471 drug store IDs. These drug stores are from 50 markets across the United States. The products include items sold from these stores during the 11-year period, and are from 31 product categories, including hot dogs, household cleaners, margarine/butter blends, mayonnaise, milk, coffee, cigarettes, photography supplies, paper towels, frozen pizza, toilet tissue, yogurt, beer/ale/alcoholic cider, blades, cold cereal, carbonated beverages, diapers, deodorant, facial tissue, frozen dinners/entrees, laundry detergent, peanut butter, razors, mustard and ketchup, sugar substitutes, spaghetti/Italian sauce, soup, shampoo, salty snacks, toothpaste, and toothbrush. Moreover, various advertising and promotions strategies are imposed on these products to attract consumers. The promotions strategies have 30 types which 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 goal of our study is to predict the future sales volumes of each product from each store given each promotion strategy based on historical sales data. Through this prediction procedure, we are able to estimate future purchases, evaluate the influence of promotion strategy for product sales, and potentially recommend the most profitable products to store managers, so the company can make wiser decisions on marketing strategies and inventory planning. For the IRI marketing data, a personalized suggestion refers to the recommendation of potentially profitable products to store managers. Statistically this can be viewed as predicting future sales volumes of each product from each store based on historical sales data. There are abundant literature on product recommender systems including, but not limited to, Giering (2008), Xiong et al. (2010) and Yu et al. (2016). In these works, similar to the proposed IRI data analysis, recommendations of products to stores are also considered. For considering the trend of product sales, we aggregate the weekly data into monthly data according to the record time information so that the data contain more than 79.2 million sales records for 132 months from the beginning of 2001 to the end of 2011. For the proposed method, we classify stores, products, observed time points and promotion strategies into subgroups based on their markets, product categories, month of the year and whether a price reduction is applied, respectively.

Table 4 shows the summary statistics of the data. According to the proposed method, the data can be reframed into monthly third-order tensors by store, product and promotion.

Table 4: Summary statistics of the monthly IRI marketing data.

	The number of types	The number of subgroups
Store	471	50
Promotion	30	2
Product	43,631	31
Month	132	12
Sales record	79,243,289	

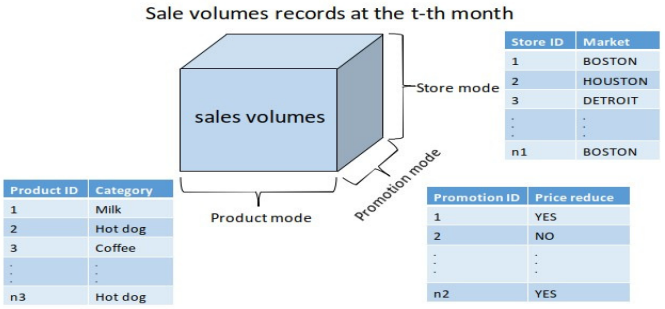

Figure 5: An illustration for monthly sale tensors at the tth month.

Figure 5 provides an illustration of the reframed sale records to clarify the proposed method. According to the given structure of monthly third-order tensors, the total number of sale records could be up to $471 \times 30 \times 43631 \times 132 \approx 8.1$ billion. Although the observed records are more than 79.2 million, there are still a large number of store-promotion-product-month combinations which are associated with unknown sales volumes. The sales data have a 99.9% missing rate and are highly sparse, which renders a particular challenge for recommender systems and forecasting.

For comparison, we implement and report the performances of the proposed methods with different working correlation matrices and the competing methods as in Section 5. For all methods, we select the number of latent factors r ranging from 3 to 30. For the REM method and the proposed methods, we select a tuning parameter λ from 1 to 29. For the BPTF methods, we use the default values of the remaining parameters. For selecting the above parameters, we set the data from the beginning of 2001 to the end of 2009 (i.e., the first 108 months) as the training set and the data from the beginning of 2010 to the end of 2010 as the validation set, and then tune these parameters through minimizing the root mean square error on the validation set. We randomly sample 80% of the data from 2001 to 2010 as a training set and 20% of the data from the entire year of 2011 as the testing set. The random sampling is replicated 50 times.

Table 5 shows the forecasting results produced by each method. The results of the DTRSar are similar to ones of the DTRSin and are omitted. From Table 5, we observe that the DTRSin method achieves coverage probabilities of the prediction interval close to 95%, implying that the proposed method can estimate the prediction interval accurately, whereas the competing methods cannot provide such prediction intervals. In addition, the

Table 5: The RMSE and MAE of the forecasting sale volumes in 2011 from five methods. The PICP is the average coverage probability of the 95% prediction interval. The RMSE, MAE and PICP are provided with standard error based on 50 experiments in each parenthesis. The RRMSE and RMAE show the relative improvement ratios of the DTRSin method over others in terms of the RMSE and MAE.

Method	RMSE	RRMSE	MAE	RMAE	PICP
DTRSin	**11.284**(0.536)	–	**3.790**(0.058)	–	0.967(0.001)
REM	13.425(1.458)	18.97%	4.072(0.261)	7.44%	–
$\text{BPTF}_{\text{bayes}}$	15.792(1.746)	39.95%	4.276(0.171)	12.82%	–
$\text{BPTF}_{\text{basic}}$	12.736(0.385)	12.87%	3.838(0.058)	1.27%	–
$\text{BPTF}_{\text{double}}$	12.732(0.388)	12.83%	3.835(0.057)	1.19%	–

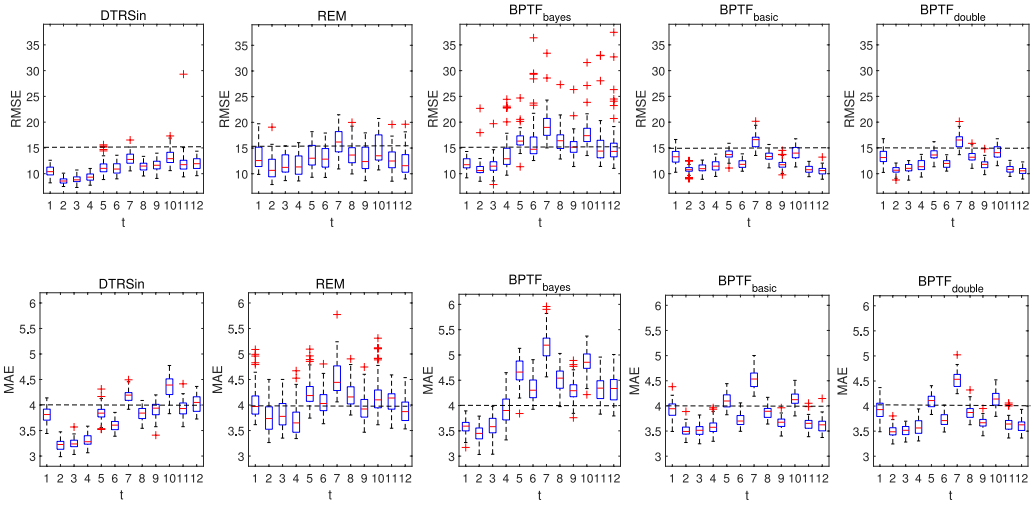

Figure 6: Box plots of the RMSE and MAE for forecasting values at 12 time points.

DTRSin method has the lowest RMSE and MAE. For example, DTRSin improves on the RMSE and MAE of the $\text{BPTF}_{\text{bayes}}$ by 39.95% and 12.82%, and of the REM by 18.97% and 7.44%, respectively. The $\text{BPTF}_{\text{basic}}$ and $\text{BPTF}_{\text{double}}$ perform better than the $\text{BPTF}_{\text{bayes}}$, while the $\text{BPTF}_{\text{double}}$ performs better than the $\text{BPTF}_{\text{basic}}$. However, the proposed method still outperforms the $\text{BPTF}_{\text{basic}}$ and $\text{BPTF}_{\text{double}}$.

To illustrate the specific performance for forecasting at each time point, we calculate the RMSE and MAE at each time point and provide box plots for the RMSE and MAE in Figure 6. We observe that the performances of the DTRSin, $\text{BPTF}_{\text{basic}}$ and $\text{BPTF}_{\text{double}}$ are more robust than those of the REM and $\text{BPTF}_{\text{bayes}}$. However, the RMSEs of the DTRSin method at each time point are still lower than those of the $\text{BPTF}_{\text{basic}}$ and $\text{BPTF}_{\text{double}}$ methods. The proposed method outperforms other methods for forecasting at each time point, and can deal with long-term forecasting accurately.

6.2 Last.fm Data

In this section, we analyze the Lastfm-1K dataset collected by Last.fm API (Celma, 2010) to evaluate the performance of the proposed method. The dataset is available at `http://ocelma.net/MusicRecommendationDataset/lastfm-1K.html` and has been widely used in music recommendation experiments. The Lastfm data includes the listening history of 992 users and songs played daily, recorded by quadruples with user, timestamp, artist and song information, where the users' profiles contain gender, age, country and signup, and artists contain 107,528 artists with ID and 69,420 without ID. A detailed description of the Lastfm-1K dataset is available at `http://ocelma.net/MusicRecommendationDataset/lastfm-1K.html`.

We extract the user-artist-song playcount tensor-valued function with each monthly time point based on the quadruples records. The goal of our study is to predict the future playcount of each song given each artist for each user. Through this prediction procedure, we are able to estimate future listening habit for each user, so that to recommend interesting songs to each user. To evaluate the performance of the proposed method, we consider a sub-dataset with 100 users randomly, where there are 7,490 artists, 32,287 songs and 53 months from February of 2005 to June of 2009. We classify users, song, time stamp and artists into subgroups based on users' gender, a song's artist, month of the year, and whether the artist has ID, respectively. Although the observed records have been 356,786, there are still a large number of user-artist-song-month combinations which are associated with unknown playcount. The data have a high missing rate and are highly sparse.

Similar to Section 6.1, we implement and report the performances of the proposed method and the competing methods. We randomly sample 80% of the data from February of 2005 to May of 2008 (i.e., the first 40 months) as a training set and 20% of the data from June of 2008 to June of 2009 (i.e., the last 13 months) as the testing set. The random sampling is replicated 50 times. Table 6 shows the forecasting results produced by each method and the computational time.

Table 6 indicates that the DTRSin method achieves coverage probabilities of the prediction interval close to 95%, which supports that the proposed method can obtain accurate prediction interval estimator, whereas the competing methods cannot provide such prediction interval. Table 6 also shows that the proposed method achieves the best performance. Specifically, the proposed method improves the compering methods by at least 8.02% with respect to the RMSE, and by at least 10.05% with respect to the MAE, and still achieves the smallest standard error with a reasonable computational time.

7. Discussion

In this article, we propose a new dynamic tensor recommender system which incorporates time information through a tensor-valued function. A unique contribution of our method is that it can estimate recommendation accurately given any time point in continuous time intervals. Technically, the proposed method builds a time-value tensor decomposition model and borrows group information from existing time points of the same group for higher forecasting accuracy. Moreover, the proposed method utilizes the polynomial spline method and the weighted least squared method to incorporate time-dependency and intra-cluster correlation into the DRS. The spline extrapolation enables our method to achieve both

Table 6: The RMSE and MAE of the forecasting the playcount of songs from five methods. The PICP and ComTime are the average coverage probability of the 95% prediction interval and the average computational time in seconds. The RMSE, MAE and PICP are provided with standard error based on 50 experiments in each parenthesis. The RRMSE and RMAE show the relative improvement ratios of the DTRSin method over others in terms of the RMSE and MAE.

Method	RMSE	RRMSE	MAE	RMAE	PICP	ComTime
DTRSin	**12.113**(1.836)	–	**1.940**(0.058)	–	0.931(0.012)	92.8s
REM	16.085(2.375)	32.79%	3.228(0.717)	66.39%	–	197.0s
$\text{BPTF}_{\text{bayes}}$	13.084(1.966)	8.02%	2.139(0.080)	10.26%	–	84.3s
$\text{BPTF}_{\text{basic}}$	13.117(2.014)	8.29%	2.137(0.082)	10.15%	–	86.2s
$\text{BPTF}_{\text{double}}$	13.116(2.015)	8.28%	2.135(0.083)	10.05%	–	82.5s

short-term and long-term forecasts accurately, as confirmed in the numerical studies. In addition, the proposed method is able to provide pointwise prediction intervals based on the established asymptotic property, while existing recommender systems are not equipped with prediction intervals. In theory, we demonstrate that the proposed decomposition achieves asymptotic consistency on prediction and the spline coefficient estimators have asymptotic normality. In addition, we show the numerical advantages of our method compared to existing methods, including the analysis of IRI marketing data.

Acknowledgments

We would like to thank the action editors and referees for insightful comments and suggestions which improve the article significantly. We would like to acknowledge support for this project from the National Science Foundation Grants (DMS1821198, DMS1613190 and DMS1952402), National Natural Science Foundation of P.R. China (11731011, 11671349 and 12001479), and Natural Science Foundation of Yunnan Province of China (2019FD068). We would like to thank IRI for making the data available. All estimates and analysis in this paper based on data provided by IRI are by the authors and not by the IRI.

Appendix A.

In this appendix we prove the theorems from Section 4.

Lemma 4 *Predicted values given by (2) with fixed spline bases are invariant with respect to scaling and permutation indeterminacies.*

■

Proof According to (2), the tensor function $\mathcal{Y}(t)$ is represented as:

$$\mathcal{Y}(t) \approx \sum_{j=1}^{r} \hat{h}_j(t)\mathbf{p}^1_{\cdot j} \circ \mathbf{p}^2_{\cdot j} \circ \cdots \circ \mathbf{p}^d_{\cdot j} + \hat{g}(t)\mathbf{q}^1 \circ \mathbf{q}^2 \circ \cdots \circ \mathbf{q}^d.$$

Given fixed spline bases $\{B_{ji}(\cdot)\}_{i=1}^{M}$, $h_j(t)$ is uniquely confirmed by coefficients $\boldsymbol{\alpha}_j$ for $j = 1, \ldots, r$, and $g(t)$ is similar. Thus, the determinacy of coefficients is equivalent to the determinacy of functional factors, so we discuss the indeterminacy of functional factors. Scaling indeterminacy refers to non-uniqueness with respect to a scale change of $\mathbf{p}^k_{\cdot j}$ and $\mathbf{q}^k$, and of each function $h_j(t)$ and $g(t)$ for $k = 1, \ldots, d; j = 1, \ldots, r$. That is, for π_{kl} and δ_l, $k = 1, \ldots, d$, $l = 1, \ldots, r+1$, we have $\widetilde{\mathbf{p}}^k_{\cdot j} = \pi_{kj}\mathbf{p}^k_{\cdot j}$, $\widetilde{h}_j(t) = \delta_j h_j(t)$, $j = 1, \ldots, r$, $\widetilde{\mathbf{q}}^k = \pi_{kr+1}\mathbf{q}^k$, and $\widetilde{g}(t) = \delta_{r+1}g(t)$ such that $\delta_l \prod_{k=1}^{d} \pi_{kl} = 1$ for $l = 1, \ldots, r+1$. Thus, we know that $\sum_{j=1}^{r} h_j(t)\mathbf{p}^1_{\cdot j} \circ \mathbf{p}^2_{\cdot j} \circ \cdots \circ \mathbf{p}^d_{\cdot j} + g(t)\mathbf{q}^1 \circ \mathbf{q}^2 \circ \cdots \circ \mathbf{q}^d = \sum_{j=1}^{r} \widetilde{h}_j(t)\widetilde{\mathbf{p}}^1_{\cdot j} \circ \widetilde{\mathbf{p}}^2_{\cdot j} \circ \cdots \circ \widetilde{\mathbf{p}}^d_{\cdot j} + \widetilde{g}(t)\widetilde{\mathbf{q}}^1 \circ \widetilde{\mathbf{q}}^2 \circ \cdots \circ \widetilde{\mathbf{q}}^d$. Let $\mathbf{h}(t) = \{h_1(t), h_2(t), \ldots, h_r(t)\}^\top$. Permutation indeterminacy refers to an arbitrary $r \times r$ permutation matrix $\boldsymbol{\Pi}$ such that $\sum_{j=1}^{r} h_j(t)\mathbf{p}^1_{\cdot j} \circ \mathbf{p}^2_{\cdot j} \circ \cdots \circ \mathbf{p}^d_{\cdot j} + g(t)\mathbf{q}^1 \circ \mathbf{q}^2 \circ \cdots \circ \mathbf{q}^d \doteq [[\mathbf{h}(t); \mathbf{P}^1, \mathbf{P}^2, \cdots, \mathbf{P}^d]] + g(t)\mathbf{q}^1 \circ \mathbf{q}^2 \circ \cdots \circ \mathbf{q}^d = [[\boldsymbol{\Pi}\mathbf{h}(t); \mathbf{P}^1\boldsymbol{\Pi}, \mathbf{P}^2\boldsymbol{\Pi}, \cdots, \mathbf{P}^d\boldsymbol{\Pi}]] + g(t)\mathbf{q}^1 \circ \mathbf{q}^2 \circ \cdots \circ \mathbf{q}^d$. These imply the invariance of equation (2) with respect to scaling and permutation indeterminacies. The proof of the Lemma is completed. ■

Proof of Proposition 1.

Based on the definition of Kruskal rank, the Kruskal rank of the $(r+1) \times 1$ matrix $(h_1(t), h_2(t), \ldots, h_r(t), g(t))^\top$ is one. Based on Theorem 3 in Sidiropoulos and Bro (2000), the sufficient condition of identifiability of $(\mathbf{P}^k, \mathbf{q}^k)$ up to permutation and scaling of columns is $\sum_{k=1}^{d} K_k + 1 \geq 2(r+1) + (d+1) - 1$, that is, $\sum_{k=1}^{d} K_k \geq 2r + d + 1$. Under the sufficient condition, if there exist two minimizers $(\widetilde{\mathbf{P}}, \widetilde{\mathbf{q}}, \widetilde{\boldsymbol{\alpha}}, \widetilde{\boldsymbol{\beta}})$ and $(\widehat{\mathbf{P}}, \widehat{\mathbf{q}}, \widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\beta}})$ of $L(\cdot|\mathbb{Y})$, then $(\widetilde{\mathbf{P}}, \widetilde{\mathbf{q}}, \widetilde{\boldsymbol{\alpha}}, \widetilde{\boldsymbol{\beta}})$ and $(\widehat{\mathbf{P}}, \widehat{\mathbf{q}}, \widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\beta}})$ are identical with the exception of scaling and permutation. By Lemma 4, the $\widehat{\mathbf{y}}_{i_1 i_2 \cdots i_d}$'s provided by $(\widetilde{\mathbf{P}}, \widetilde{\mathbf{q}}, \widetilde{\boldsymbol{\alpha}}, \widetilde{\boldsymbol{\beta}})$ and $(\widehat{\mathbf{P}}, \widehat{\mathbf{q}}, \widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\beta}})$ are identical. Thus, $L(\widetilde{\mathbf{P}}, \widetilde{\mathbf{q}}, \widetilde{\boldsymbol{\alpha}}, \widetilde{\boldsymbol{\beta}}|\mathbb{Y}) = L(\widehat{\mathbf{P}}, \widehat{\mathbf{q}}, \widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\beta}}|\mathbb{Y})$ implies that

$$\begin{aligned} & \textstyle\sum_{k=1}^{d}(\sum_{j=1}^{r} \|\widetilde{\mathbf{p}}^k_{\cdot j}\|_2^2 + \|\widetilde{\mathbf{q}}^k\|_2^2) + \sum_{j=1}^{r} \|\widetilde{\boldsymbol{\alpha}}_j\|^2 + \sum_{e=1}^{m_{d+1}} \|\widetilde{\boldsymbol{\beta}}_e\|^2 \\ = \; & \textstyle\sum_{k=1}^{d}(\sum_{j=1}^{r} \|\widehat{\mathbf{p}}^k_{\cdot j}\|_2^2 + \|\widehat{\mathbf{q}}^k\|_2^2) + \sum_{j=1}^{r} \|\widehat{\boldsymbol{\alpha}}_j\|^2 + \sum_{e=1}^{m_{d+1}} \|\widehat{\boldsymbol{\beta}}_e\|^2. \end{aligned} \tag{12}$$

Suppose there exist some $k_1, k_2 = 1, \ldots, d$, $k_1 \neq k_2$, such that $\widehat{\mathbf{p}}^{k_1}_{\cdot j} = \nu_j \widetilde{\mathbf{p}}^{k_1}_{\cdot j}$, $\widehat{\mathbf{p}}^{k_2}_{\cdot j} = \widetilde{\mathbf{p}}^{k_2}_{\cdot j}/\nu_j$, $\widehat{\mathbf{q}}^{k_1} = \tau\widetilde{\mathbf{q}}^{k_1}$, $\widehat{\mathbf{q}}^{k_2} = \widetilde{\mathbf{q}}^{k_2}/\tau$ for positive constants τ, ν_j, $j = 1, \ldots, r$. We have

$$\sum_{j=1}^{r}(\|\widehat{\mathbf{p}}^{k_1}_{\cdot j}\|_2^2 + \|\widehat{\mathbf{p}}^{k_2}_{\cdot j}\|_2^2) + \|\widehat{\mathbf{q}}^{k_1}\|_2^2 + \|\widehat{\mathbf{q}}^{k_2}\|_2^2 = \sum_{j=1}^{r}(\nu_j^2\|\widetilde{\mathbf{p}}^{k_1}_{\cdot j}\|_2^2 + \|\widetilde{\mathbf{p}}^{k_2}_{\cdot j}\|_2^2/\nu_j^2) + \tau^2\|\widetilde{\mathbf{q}}^{k_1}\|_2^2 + \|\widetilde{\mathbf{q}}^{k_2}\|_2^2/\tau^2.$$

Then (12) implies that $\tau = 1$ and ν_j almost surely, $j = 1, \ldots, r$. Similarly, suppose there exist some $k_1 = 1, \ldots, d$, such that $\widehat{\mathbf{p}}_{\cdot j}^{k_1} = \nu_j \widetilde{\mathbf{p}}_{\cdot j}^{k_1}$, $\widehat{\boldsymbol{\alpha}}_j = \widetilde{\boldsymbol{\alpha}}_j/\nu_j$, $\widehat{\mathbf{q}}^{k_1} = \tau \widetilde{\mathbf{q}}^{k_1}$, $\widehat{\boldsymbol{\beta}}_e = \widetilde{\boldsymbol{\beta}}_e/\tau$ for positive constants τ, ν_j, $j = 1, \ldots, r; e = 1, \ldots, m_{d+1}$. We have

$$\begin{aligned} & \textstyle\sum_{j=1}^r (\|\widehat{\mathbf{p}}_{\cdot j}^{k_1}\|_2^2 + \|\widehat{\boldsymbol{\alpha}}_j\|^2) + \|\widehat{\mathbf{q}}^{k_1}\|_2^2 + \sum_{e=1}^{m_{d+1}} \|\widehat{\boldsymbol{\beta}}_e\|^2 \\ = \; & \textstyle\sum_{j=1}^r (\nu_j^2 \|\widetilde{\mathbf{p}}_{\cdot j}^{k_1}\|_2^2 + \|\widetilde{\boldsymbol{\alpha}}_j\|^2/\nu_j^2) + \tau^2 \|\widetilde{\mathbf{q}}^{k_1}\|_2^2 + \sum_{e=1}^{m_{d+1}} \|\widetilde{\boldsymbol{\beta}}_e\|^2/\tau^2. \end{aligned}$$

Then (12) implies that $\tau = 1$ and ν_j almost surely, $j = 1, \ldots, r$. Thus, $(\widetilde{\mathbf{P}}, \widetilde{\mathbf{q}}, \widetilde{\boldsymbol{\alpha}}, \widetilde{\boldsymbol{\beta}})$ and $(\widehat{\mathbf{P}}, \widehat{\mathbf{q}}, \widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\beta}})$ are identical almost surely with the exception of permutation. ■

Proof of Theorem 2.

The estimator $\widehat{\mathbf{u}}_{i_1 \cdots i_d}$ is obtained by solving $\partial_{(i_1 \cdots i_d)} L(\mathcal{U}|\mathbb{Y}) = \mathbf{0}$, where $\partial_{(i_1 \cdots i_d)}$ represents the first derivatives with respect to the vector $\mathbf{u}_{i_1 \cdots i_d}$. By Taylor expansion, we have

$$\begin{aligned} & \widehat{\mathbf{u}}_{i_1 \cdots i_d} - \mathbf{u}_{0 i_1 \cdots i_d} \\ = \; & \{\partial^2_{(i_1 \cdots i_d)} L(\mathcal{U}|\mathbb{Y})|_{\mathbf{u}^*_{i_1 \cdots i_d}}\}^{-1} \partial_{(i_1 \cdots i_d)} L(\mathcal{U}|\mathbb{Y})|_{\mathbf{u}_{0 i_1 \cdots i_d}} \\ = \; & \{\mathbf{F}_{i_1 i_2 \ldots i_d}^\top \boldsymbol{\Sigma}_{i_1 i_2 \ldots i_d}^{-1} \mathbf{F}_{i_1 i_2 \ldots i_d} + \lambda \partial^2_{(i_1 \cdots i_d)} J(\mathcal{U})|_{\mathbf{u}^*_{i_1 \cdots i_d}}\}^{-1} \\ & \{\mathbf{F}_{i_1 i_2 \ldots i_d}^\top \boldsymbol{\Sigma}_{i_1 i_2 \ldots i_d}^{-1} \boldsymbol{\varepsilon}_{i_1 i_2 \ldots i_d} - \lambda \partial_{(i_1 \cdots i_d)} J(\mathcal{U})|_{\mathbf{u}_{0 i_1 \cdots i_d}}\} \\ = \; & I_1^{-1} I_2 \end{aligned}$$

where $\mathbf{u}^*_{i_1 \cdots i_d}$ is between $\widehat{\mathbf{u}}_{i_1 \cdots i_d}$ and $\mathbf{u}_{0 i_1 \cdots i_d}$, and $\partial^2_{(i_1 \cdots i_d)}$ represents the second derivatives with respect to the vector $\mathbf{u}_{i_1 \cdots i_d}$. Since $\lambda = o_p(1)$ and $J(\mathcal{U})$ have bounded first and second derivatives at true parameter $\mathcal{U}_0$, $\lambda \partial^2_{(i_1 \cdots i_d)} J(\mathcal{U})|_{\mathbf{u}^*_{i_1 \cdots i_d}} = o_p(1)$ and $\lambda \partial_{(i_1 \cdots i_d)} J(\mathcal{U})|_{\mathbf{u}_{0 i_1 \cdots i_d}} = o_p(1)$. Under conditions (C1), (C3) and (C4), we have

$$\begin{aligned} \|I_1\|_F &= \|\mathbf{F}_{i_1 i_2 \ldots i_d}^\top (\boldsymbol{\Sigma}_{i_1 i_2 \ldots i_d}^{-1} \boldsymbol{\Sigma}_{i_1 i_2 \ldots i_d}^0)(\boldsymbol{\Sigma}_{i_1 i_2 \ldots i_d}^0)^{-1} \mathbf{F}_{i_1 i_2 \ldots i_d} + \lambda \partial^2_{(i_1 \cdots i_d)} J(\mathcal{U})|_{\mathbf{u}^*_{i_1 \cdots i_d}}\|_F \\ &\geq \|c_1 \sigma_2^{-2} \mathbf{F}_{i_1 i_2 \ldots i_d}^\top \mathbf{F}_{i_1 i_2 \ldots i_d} + \lambda \partial^2_{(i_1 \cdots i_d)} J(\mathcal{U})|_{\mathbf{u}^*_{i_1 \cdots i_d}}\|_F \\ &\geq \|c_1 \sigma_2^{-2} \min_{t \in \mathbb{T}_{i_1 \ldots i_d}} \{|\mathbb{T}_{i_1 \ldots i_d}|\} \{\tfrac{1}{|\mathbb{T}_{i_1 \ldots i_d}|} \textstyle\sum_{t \in \mathbb{T}_{i_1 \ldots i_d}} \mathbf{f}(t) \mathbf{f}(t)^\top\} \\ &\quad + \lambda \partial^2_{(i_1 \cdots i_d)} J(\mathcal{U})|_{\mathbf{u}^*_{i_1 \cdots i_d}}\|_F \\ &\gtrsim T_{\min}, \\ \|I_2\|_F &\leq \|c_2 \sigma_1^{-2} \textstyle\sum_{t \in \mathbb{T}_{i_1 \ldots i_d}} \mathbf{f}(t) \varepsilon_{i_1 \ldots i_d t} - \lambda \partial_{(i_1 \cdots i_d)} J(\mathcal{U})|_{\mathbf{u}_{0 i_1 \cdots i_d}}\|_F \\ &\leq \|C c_2 \sigma_1^{-2} |\mathbb{T}_{i_1 \ldots i_d}| \{\tfrac{1}{|\mathbb{T}_{i_1 \ldots i_d}|} \textstyle\sum_{t \in \mathbb{T}_{i_1 \ldots i_d}} \varepsilon_{i_1 \ldots i_d t}\} - \lambda \partial_{(i_1 \cdots i_d)} J(\mathcal{U})|_{\mathbf{u}_{0 i_1 \cdots i_d}}\|_F \\ &\lesssim |\mathbb{T}_{i_1 \ldots i_d}|^{1/2} \lesssim T_{\max}^{1/2}, \end{aligned}$$

where $a \lesssim b$ and $b \gtrsim a$ mean a/b is bounded, $T_{\min} = \min_{t \in \mathbb{T}_{i_1 \ldots i_d}} \{|\mathbb{T}_{i_1 \ldots i_d}|\}$ and $T_{\max} = \max_{t \in \mathbb{T}_{i_1 \ldots i_d}} \{|\mathbb{T}_{i_1 \ldots i_d}|\}$. Thus, under condition (C5), we have $\|\widehat{\mathbf{u}}_{i_1 \cdots i_d} - \mathbf{u}_{0 i_1 \cdots i_d}\|_2 \lesssim T_{\max}^{1/2}/T_{\min}$

$\lesssim N^{\tau/2-\upsilon}$. Under condition (C1) and (C5), we have

$$\begin{aligned}
& \tfrac{1}{N}\textstyle\sum_{(i_1,\cdots,i_d)\in\Omega}\|\mathbf{F}_{i_1\cdots i_d}(\widehat{\mathbf{u}}_{i_1\cdots i_d}-\mathbf{u}_{0i_1\cdots i_d})\|_2^2 \\
= & \tfrac{1}{N}\textstyle\sum_{(i_1,\cdots,i_d)\in\Omega}(\widehat{\mathbf{u}}_{i_1\cdots i_d}-\mathbf{u}_{0i_1\cdots i_d})^\top\sum_{t\in\mathbb{T}_{i_1\cdots i_d}}\mathbf{f}(t)\mathbf{f}(t)^\top(\widehat{\mathbf{u}}_{i_1\cdots i_d}-\mathbf{u}_{0i_1\cdots i_d}) \\
\leq & C\max_{t\in\mathbb{T}_{i_1\cdots i_d}}\{|\mathbb{T}_{i_1\cdots i_d}|\}\tfrac{1}{N}\textstyle\sum_{(i_1,\cdots,i_d)\in\Omega}(\widehat{\mathbf{u}}_{i_1\cdots i_d}-\mathbf{u}_{0i_1\cdots i_d})^\top(\widehat{\mathbf{u}}_{i_1\cdots i_d}-\mathbf{u}_{0i_1\cdots i_d}) \\
\lesssim & \tfrac{T_{\max}^2}{NT_{\min}^2}\lesssim N^{-1+2(\tau-\upsilon)}.
\end{aligned}$$

The proof of Theorem 2 is completed. ■

We can currently use a convenient basis system in our technical arguments and the results also hold true for other basis choices of the same function space. The B-spline and truncated polynomial basis functions span the same set of spline functions (de Boor, 2001), thus we use B-splines as the convenient basis system in our proofs. The B-splines have the following properties (de Boor, 2001): $B_k(t)\geq 0$, $\sum_{k=1}^M B_k(t)=1$, $t\in\mathbb{T}$, $\frac{C_1}{M}\sum_{k=1}^M\phi_k^2 dt\leq\int_{\mathbb{T}}(\sum_{k=1}^M\phi_k B_k(t))^2\leq\frac{C_2}{M}\sum_{k=1}^M\phi_k^2$, C_1 and C_2 are constant and $\phi_k\in\mathbb{R}$.

Lemma 5 *Under Conditions (C1)-(C5), we have*

$$\begin{aligned}
&\tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}(\widehat{\mathbf{W}}_{i_1\ldots i_d}-\mathbf{W}_{i_1\ldots i_d})^\top\mathbf{\Sigma}_{i_1\ldots i_d}^{-1}(\widehat{\mathbf{W}}_{i_1\ldots i_d}-\mathbf{W}_{i_1\ldots i_d})=O_p(N^{-1+2(\tau-\upsilon)}),\\
&\tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}(\widehat{\mathbf{W}}_{i_1\ldots i_d}-\mathbf{W}_{i_1\ldots i_d})^\top\mathbf{\Sigma}_{i_1\ldots i_d}^{-1}\mathbf{W}_{i_1\ldots i_d}=O_p(N^{-1+3\tau/2-\upsilon}),\\
&\tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}(\widehat{\mathbf{W}}_{i_1\ldots i_d}-\mathbf{W}_{i_1\ldots i_d})^\top\mathbf{\Sigma}_{i_1\ldots i_d}^{-1}\boldsymbol{\varepsilon}_{i_1\ldots i_d}=O_p(N^{-1+\tau-\upsilon}).
\end{aligned}$$

■

Proof By Theorem 2 and the non-negative bounded properties of the B-spline basis functions (de Boor, 2001), we have for $j,l=1,\ldots,r$; $e,k=1,\ldots,m_{d+1}$,

$$\begin{aligned}
& \tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}(\widehat{\mathbf{X}}_{i_1\ldots i_dj}-\mathbf{X}_{i_1\ldots i_dj})^\top\mathbf{\Sigma}_{i_1\ldots i_d}^{-1}(\widehat{\mathbf{X}}_{i_1\ldots i_dl}-\mathbf{X}_{i_1\ldots i_dl}) \\
= & \tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}(\widehat{u}_{i_1\ldots i_dj}-u_{i_1\ldots i_dj})(\widehat{u}_{i_1\ldots i_dl}-u_{i_1\ldots i_dl})\mathbf{B}_{i_1\ldots i_dj}^\top\mathbf{\Sigma}_{i_1\ldots i_d}^{-1}\mathbf{B}_{i_1\ldots i_dl} \\
\leq & c_2\sigma_1^{-2}T_{\max}\tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}(\widehat{u}_{i_1\ldots i_dj}-u_{i_1\ldots i_dj})(\widehat{u}_{i_1\ldots i_dl}-u_{i_1\ldots i_dl}) \\
& \cdot\{\tfrac{1}{|\mathbb{T}_{i_1\ldots i_d}|}\textstyle\sum_{t\in\mathbb{T}_{i_1\ldots i_d}}\mathbf{B}_j(t)^\top\mathbf{B}_l(t)\} \\
= & O_p(N^{-1+2(\tau-\upsilon)}),
\end{aligned}$$

$$\begin{aligned}
& \tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}(\widehat{\mathbf{X}}_{i_1\ldots i_dj}-\mathbf{X}_{i_1\ldots i_dj})^\top\mathbf{\Sigma}_{i_1\ldots i_d}^{-1}(\widehat{\mathbf{Z}}_{i_1\ldots i_de}-\mathbf{Z}_{i_1\ldots i_de}) \\
\leq & c_2\sigma_1^{-2}\tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}(\widehat{u}_{i_1\ldots i_dj}-u_{i_1\ldots i_dj})(\widehat{u}_{i_1\ldots i_d(r+1)}-u_{i_1\ldots i_d(r+1)})\mathbf{B}_{i_1\ldots i_dj}^\top\mathbf{A}_{i_1\ldots i_de} \\
= & O_p(N^{-1+2(\tau-\upsilon)}),
\end{aligned}$$

and

$$\begin{aligned}
& \tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}(\widehat{\mathbf{Z}}_{i_1\ldots i_dk}-\mathbf{Z}_{i_1\ldots i_dk})^\top\mathbf{\Sigma}_{i_1\ldots i_d}^{-1}(\widehat{\mathbf{Z}}_{i_1\ldots i_de}-\mathbf{Z}_{i_1\ldots i_de}) \\
\leq & c_2\sigma_1^{-2}\tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}(\widehat{u}_{i_1\ldots i_d(r+1)}-u_{i_1\ldots i_d(r+1)})^2\mathbf{A}_{i_1\ldots i_dk}^\top\mathbf{A}_{i_1\ldots i_de}=O_p(N^{-1+2(\tau-\upsilon)}).
\end{aligned}$$

By definition, we have

$$\frac{1}{N}\sum_{(i_1,\ldots,i_d)\in\Omega}(\widehat{\mathbf{W}}_{i_1\ldots i_d}-\mathbf{W}_{i_1\ldots i_d})^\top\mathbf{\Sigma}_{i_1\ldots i_d}^{-1}(\widehat{\mathbf{W}}_{i_1\ldots i_d}-\mathbf{W}_{i_1\ldots i_d})=O_p(N^{-1+2(\tau-\upsilon)}).$$

Under conditions (C3)-(C5), we have $j,l=1,\ldots,r$; $e,k=1,\ldots,m_{d+1}$,

$$\begin{aligned}
&\tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}(\widehat{\mathbf{X}}_{i_1\ldots i_d j}-\mathbf{X}_{i_1\ldots i_d j})^\top\mathbf{\Sigma}_{i_1\ldots i_d}^{-1}\mathbf{X}_{i_1\ldots i_d l}\\
\leq\;& c_2\sigma_1^{-2}\tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}(\widehat{u}_{i_1\ldots i_d j}-u_{i_1\ldots i_d j})u_{i_1\ldots i_d l}\mathbf{B}_{i_1\ldots i_d j}^\top\mathbf{B}_{i_1\ldots i_d l}\\
\leq\;& c_2\sigma_1^{-2}T_{\max}\tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}(\widehat{u}_{i_1\ldots i_d j}-u_{i_1\ldots i_d j})u_{i_1\ldots i_d l}\{\frac{1}{|\mathbb{T}_{i_1\ldots i_d}|}\sum_{t\in\mathbb{T}_{i_1\ldots i_d}}\mathbf{B}_j(t)^\top\mathbf{B}_l(t)\}\\
=\;& O_p(N^{-1+3\tau/2-\upsilon}),\\
&\tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}(\widehat{\mathbf{X}}_{i_1\ldots i_d j}-\mathbf{X}_{i_1\ldots i_d j})^\top\mathbf{\Sigma}_{i_1\ldots i_d}^{-1}\boldsymbol{\varepsilon}_{i_1\ldots i_d}\\
\leq\;& c_2\sigma_1^{-2}\tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}(\widehat{u}_{i_1\ldots i_d j}-u_{i_1\ldots i_d j})\mathbf{B}_{i_1\ldots i_d j}^\top\boldsymbol{\varepsilon}_{i_1\ldots i_d}\\
\leq\;& CT_{\max}^{1/2}\tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}(\widehat{u}_{i_1\ldots i_d j}-u_{i_1\ldots i_d j})\{\frac{1}{|\mathbb{T}_{i_1\ldots i_d}|^{1/2}}\sum_{t\in\mathbb{T}_{i_1\ldots i_d}}\varepsilon_{i_1\ldots i_d t}\}\\
=\;& O_p(N^{-1+\tau-\upsilon}),\\
&\tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}(\widehat{\mathbf{X}}_{i_1\ldots i_d j}-\mathbf{X}_{i_1\ldots i_d j})^\top\mathbf{\Sigma}_{i_1\ldots i_d}^{-1}\mathbf{Z}_{i_1\ldots i_d e}=O_p(N^{-1+3\tau/2-\upsilon}),\\
&\tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}(\widehat{\mathbf{Z}}_{i_1\ldots i_d e}-\mathbf{Z}_{i_1\ldots i_d e})^\top\mathbf{\Sigma}_{i_1\ldots i_d}^{-1}\mathbf{X}_{i_1\ldots i_d j}=O_p(N^{-1+3\tau/2-\upsilon}),\\
&\tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}(\widehat{\mathbf{Z}}_{i_1\ldots i_d s}-\mathbf{Z}_{i_1\ldots i_d s})^\top\mathbf{\Sigma}_{i_1\ldots i_d}^{-1}\mathbf{Z}_{i_1\ldots i_d e}=O_p(N^{-1+3\tau/2-\upsilon}),\\
&\tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}(\widehat{\mathbf{Z}}_{i_1\ldots i_d e}-\mathbf{Z}_{i_1\ldots i_d e})^\top\mathbf{\Sigma}_{i_1\ldots i_d}^{-1}\boldsymbol{\varepsilon}_{i_1\ldots i_d}=O_p(N^{-1+\tau-\upsilon}).
\end{aligned}$$

By definition, we can obtain

$$\frac{1}{N}\sum_{(i_1,\ldots,i_d)\in\Omega}(\widehat{\mathbf{W}}_{i_1\ldots i_d}-\mathbf{W}_{i_1\ldots i_d})^\top\mathbf{\Sigma}_{i_1\ldots i_d}^{-1}\mathbf{W}_{i_1\ldots i_d}=O_p(N^{-1+3\tau/2-\upsilon}),$$

and

$$\frac{1}{N}\sum_{(i_1,\ldots,i_d)\in\Omega}(\widehat{\mathbf{W}}_{i_1\ldots i_d}-\mathbf{W}_{i_1\ldots i_d})^\top\mathbf{\Sigma}_{i_1\ldots i_d}^{-1}\boldsymbol{\varepsilon}_{i_1\ldots i_d}=O_p(N^{-1+\tau-\upsilon}).$$

■

Proof of Theorem 3

Based on the criterion function (3), we can obtain the estimator of the coefficient as follows:

$$\widehat{\boldsymbol{\gamma}}=(\sum_{(i_1,i_2,\ldots,i_d)\in\Omega}\widehat{\mathbf{W}}_{i_1i_2\ldots i_d}^\top\mathbf{\Sigma}_{i_1i_2\ldots i_d}^{-1}\widehat{\mathbf{W}}_{i_1i_2\ldots i_d}+\lambda\mathbf{I})^{-1}\sum_{(i_1,i_2,\ldots,i_d)\in\Omega}\widehat{\mathbf{W}}_{i_1i_2\ldots i_d}^\top\mathbf{\Sigma}_{i_1i_2\ldots i_d}^{-1}\mathbf{y}_{i_1i_2\ldots i_d}.$$

Thus, we have

$$\begin{aligned}
\widehat{\boldsymbol{\gamma}}-\boldsymbol{\gamma}_0 \;=\;& \Big(\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}\widehat{\mathbf{W}}_{i_1\ldots i_d}^\top\mathbf{\Sigma}_{i_1i_2\ldots i_d}^{-1}\widehat{\mathbf{W}}_{i_1\ldots i_d}/N+\lambda\mathbf{I}/N\Big)^{-1}\\
&\Big\{\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}\widehat{\mathbf{W}}_{i_1\ldots i_d}^\top\mathbf{\Sigma}_{i_1i_2\ldots i_d}^{-1}(\mathbf{y}_{i_1\ldots i_d}-\widehat{\mathbf{W}}_{i_1\ldots i_d}\boldsymbol{\gamma}_0)/N-\lambda\boldsymbol{\gamma}_0/N\Big\}.
\end{aligned}$$

By Lemma 5, we have

$$\begin{aligned}
& \tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega} \widehat{\mathbf{W}}_{i_1\ldots i_d}^{\top}\boldsymbol{\Sigma}_{i_1 i_2\ldots i_d}^{-1}\widehat{\mathbf{W}}_{i_1\ldots i_d} \\
= \; & \tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega} \mathbf{W}_{i_1\ldots i_d}^{\top}\boldsymbol{\Sigma}_{i_1 i_2\ldots i_d}^{-1}\mathbf{W}_{i_1\ldots i_d} \\
& +\tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}(\widehat{\mathbf{W}}_{i_1\ldots i_d}-\mathbf{W}_{i_1\ldots i_d})^{\top}\boldsymbol{\Sigma}_{i_1 i_2\ldots i_d}^{-1}(\widehat{\mathbf{W}}_{i_1\ldots i_d}-\mathbf{W}_{i_1\ldots i_d}) \\
& +\tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}(\widehat{\mathbf{W}}_{i_1\ldots i_d}-\mathbf{W}_{i_1\ldots i_d})^{\top}\boldsymbol{\Sigma}_{i_1 i_2\ldots i_d}^{-1}\mathbf{W}_{i_1\ldots i_d} \\
& +\tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega} \mathbf{W}_{i_1\ldots i_d}^{\top}\boldsymbol{\Sigma}_{i_1 i_2\ldots i_d}^{-1}(\widehat{\mathbf{W}}_{i_1\ldots i_d}-\mathbf{W}_{i_1\ldots i_d}) \\
= \; & \tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega} \mathbf{W}_{i_1\ldots i_d}^{\top}\boldsymbol{\Sigma}_{i_1 i_2\ldots i_d}^{-1}\mathbf{W}_{i_1\ldots i_d}+O_p(N^{-1+3\tau/2-\upsilon})
\end{aligned}$$

and

$$\begin{aligned}
& \tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega} \widehat{\mathbf{W}}_{i_1\ldots i_d}^{\top}\boldsymbol{\Sigma}_{i_1 i_2\ldots i_d}^{-1}(\mathbf{y}_{i_1\ldots i_d}-\widehat{\mathbf{W}}_{i_1\ldots i_d}\boldsymbol{\gamma}_0) \\
= \; & \tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega} \mathbf{W}_{i_1\ldots i_d}^{\top}\boldsymbol{\Sigma}_{i_1 i_2\ldots i_d}^{-1}(\mathbf{y}_{i_1\ldots i_d}-\mathbf{W}_{i_1\ldots i_d}\boldsymbol{\gamma}_0) \\
& -\tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}(\widehat{\mathbf{W}}_{i_1\ldots i_d}-\mathbf{W}_{i_1\ldots i_d})^{\top}\boldsymbol{\Sigma}_{i_1 i_2\ldots i_d}^{-1}(\widehat{\mathbf{W}}_{i_1\ldots i_d}-\mathbf{W}_{i_1\ldots i_d})\boldsymbol{\gamma}_0 \\
& +\tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}(\widehat{\mathbf{W}}_{i_1\ldots i_d}-\mathbf{W}_{i_1\ldots i_d})^{\top}\boldsymbol{\Sigma}_{i_1 i_2\ldots i_d}^{-1}(\mathbf{y}_{i_1\ldots i_d}-\mathbf{W}_{i_1\ldots i_d}\boldsymbol{\gamma}_0) \\
& -\tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega} \mathbf{W}_{i_1\ldots i_d}^{\top}\boldsymbol{\Sigma}_{i_1 i_2\ldots i_d}^{-1}(\widehat{\mathbf{W}}_{i_1\ldots i_d}-\mathbf{W}_{i_1\ldots i_d})\boldsymbol{\gamma}_0 \\
= \; & \tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega} \mathbf{W}_{i_1\ldots i_d}^{\top}\boldsymbol{\Sigma}_{i_1 i_2\ldots i_d}^{-1}(\mathbf{y}_{i_1\ldots i_d}-\mathbf{W}_{i_1\ldots i_d}\boldsymbol{\gamma}_0) \\
& +\tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}(\widehat{\mathbf{W}}_{i_1\ldots i_d}-\mathbf{W}_{i_1\ldots i_d})^{\top}\boldsymbol{\Sigma}_{i_1 i_2\ldots i_d}^{-1}(\mathbf{y}_{i_1\ldots i_d}-\mathbf{W}_{i_1\ldots i_d}\boldsymbol{\gamma}_0)+O_p(N^{-1+3\tau/2-\upsilon}).
\end{aligned}$$

According to de Boor (2001), $\sup_{t\in\mathbb{T}}|h_j(t)-\widetilde{h}_j(t)|\le Ca_N^{-\xi}$ and $\sup_{t\in\mathbb{T}}|g_e(t)-\widetilde{g}_e(t)|\le Ca_N^{-\xi}$ for each $j=1,\ldots,r$ and each $e=1,\ldots,m_{d+1}$, where $\widetilde{h}_j(t)=\boldsymbol{\alpha}_{0j}^{\top}\mathbf{B}_j(t)$ and $\widetilde{g}_e(t)=\boldsymbol{\beta}_{0e}^{\top}\mathbf{A}_e(t)$. Let $\widetilde{\mathbf{y}}_{i_1\ldots i_d}=\mathbf{W}_{i_1\ldots i_d}\boldsymbol{\gamma}_0=\sum_{j=1}^{r}u_{i_1\ldots i_d j}\widetilde{\mathbf{h}}_{i_1\ldots i_d j}+u_{i_1\ldots i_d(r+1)}\widetilde{\boldsymbol{g}}_{i_1\ldots i_d}$, $\overline{\mathbf{y}}_{i_1\ldots i_d}=\sum_{j=1}^{r}u_{i_1\ldots i_d j}\mathbf{h}_{i_1\ldots i_d j}+u_{i_1\ldots i_d(r+1)}\boldsymbol{g}_{i_1\ldots i_d}$, where $\widetilde{\mathbf{h}}_{i_1\ldots i_d j}=\mathbf{B}_{i_1 i_2\ldots i_d j}\boldsymbol{\alpha}_{0j}$, $\widetilde{\boldsymbol{g}}_{i_1\ldots i_d}=\sum_{e=1}^{m_{d+1}}\mathbf{A}_{i_1 i_2\ldots i_d e}\boldsymbol{\beta}_{0e}$, and $\mathbf{h}_{i_1\ldots i_d j}$ and $\boldsymbol{g}_{i_1\ldots i_d}$ consist of $h_j(t)$ and $g(t)$ for all $t\in\mathbb{T}_{i_1\ldots i_d}$, respectively. That is, $\mathbf{y}_{i_1\ldots i_d}=\overline{\mathbf{y}}_{i_1\ldots i_d}+\boldsymbol{\varepsilon}_{i_1\ldots i_d}$. Similar to the proof of lemma 5, we can obtain that

$$\frac{1}{N}\sum_{(i_1,\ldots,i_d)\in\Omega}\mathbf{W}_{i_1\ldots i_d}^{\top}\boldsymbol{\Sigma}_{i_1 i_2\ldots i_d}^{-1}\{\sum_{j=1}^{r}u_{i_1\ldots i_d j}(\mathbf{h}_{i_1\ldots i_d j}-\widetilde{\mathbf{h}}_{i_1\ldots i_d j})\}=O_p(a_N^{-\xi}N^{\tau-1}),$$

and

$$\frac{1}{N}\sum_{(i_1,\ldots,i_d)\in\Omega}\mathbf{W}_{i_1\ldots i_d}^{\top}\boldsymbol{\Sigma}_{i_1 i_2\ldots i_d}^{-1}(\boldsymbol{g}_{i_1\ldots i_d}-\widetilde{\boldsymbol{g}}_{i_1\ldots i_d})u_{i_1\ldots i_d(r+1)}=O_p(a_N^{-\xi}N^{\tau-1}).$$

Therefore, we have

$$\begin{aligned}
&\tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}\mathbf{W}_{i_1\ldots i_d}^\top\boldsymbol{\Sigma}_{i_1i_2\ldots i_d}^{-1}(\mathbf{y}_{i_1\ldots i_d}-\mathbf{W}_{i_1\ldots i_d}\boldsymbol{\gamma}_0)\\
=\;&\tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}\mathbf{W}_{i_1\ldots i_d}^\top\boldsymbol{\Sigma}_{i_1i_2\ldots i_d}^{-1}(\mathbf{y}_{i_1\ldots i_d}-\overline{\mathbf{y}}_{i_1\ldots i_d})\\
&+\tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}\mathbf{W}_{i_1\ldots i_d}^\top\boldsymbol{\Sigma}_{i_1i_2\ldots i_d}^{-1}(\overline{\mathbf{y}}_{i_1\ldots i_d}-\widetilde{\mathbf{y}}_{i_1\ldots i_d})\\
=\;&\tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}\mathbf{W}_{i_1\ldots i_d}^\top\boldsymbol{\Sigma}_{i_1i_2\ldots i_d}^{-1}\boldsymbol{\varepsilon}_{i_1\ldots i_d}\\
&+\tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}\mathbf{W}_{i_1\ldots i_d}^\top\boldsymbol{\Sigma}_{i_1i_2\ldots i_d}^{-1}\{\sum_{j=1}^r u_{i_1\ldots i_dj}(\mathbf{h}_{i_1\ldots i_dj}-\widetilde{\mathbf{h}}_{i_1\ldots i_dj})\}\\
&+\tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}\mathbf{W}_{i_1\ldots i_d}^\top\boldsymbol{\Sigma}_{i_1i_2\ldots i_d}^{-1}(\boldsymbol{g}_{i_1\ldots i_d}-\widetilde{\boldsymbol{g}}_{i_1\ldots i_d})u_{i_1\ldots i_d(r+1)}\\
=\;&\tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}\mathbf{W}_{i_1\ldots i_d}^\top\boldsymbol{\Sigma}_{i_1i_2\ldots i_d}^{-1}\boldsymbol{\varepsilon}_{i_1\ldots i_d}+o_p(a_N^{-\xi}N^{\tau-1}).
\end{aligned}$$

Similarly, we can obtain

$$\begin{aligned}
&\tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}(\widehat{\mathbf{W}}_{i_1\ldots i_d}-\mathbf{W}_{i_1\ldots i_d})^\top\boldsymbol{\Sigma}_{i_1i_2\ldots i_d}^{-1}(\mathbf{y}_{i_1\ldots i_d}-\mathbf{W}_{i_1\ldots i_d}\boldsymbol{\gamma}_0)\\
=\;&\tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}(\widehat{\mathbf{W}}_{i_1\ldots i_d}-\mathbf{W}_{i_1\ldots i_d})^\top\boldsymbol{\Sigma}_{i_1i_2\ldots i_d}^{-1}\boldsymbol{\varepsilon}_{i_1\ldots i_d}\\
&+\tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}(\widehat{\mathbf{W}}_{i_1\ldots i_d}-\mathbf{W}_{i_1\ldots i_d})^\top\boldsymbol{\Sigma}_{i_1i_2\ldots i_d}^{-1}\{\sum_{j=1}^r u_{i_1\ldots i_dj}(\mathbf{h}_{i_1\ldots i_dj}-\widetilde{\mathbf{h}}_{i_1\ldots i_dj})\}\\
&+\tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}(\widehat{\mathbf{W}}_{i_1\ldots i_d}-\mathbf{W}_{i_1\ldots i_d})^\top\boldsymbol{\Sigma}_{i_1i_2\ldots i_d}^{-1}(\boldsymbol{g}_{i_1\ldots i_d}-\widetilde{\boldsymbol{g}}_{i_1\ldots i_d})u_{i_1\ldots i_d(r+1)}\\
=\;&O_p(N^{-1+\tau-\upsilon})+O_p(a_N^{-\xi}N^{-1+3\tau/2-\upsilon}).
\end{aligned}$$

Thus, we have

$$\begin{aligned}
&\tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}\widehat{\mathbf{W}}_{i_1\ldots i_d}^\top\boldsymbol{\Sigma}_{i_1i_2\ldots i_d}^{-1}(\mathbf{y}_{i_1\ldots i_d}-\widehat{\mathbf{W}}_{i_1\ldots i_d}\boldsymbol{\gamma}_0)\\
=\;&\tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}\mathbf{W}_{i_1\ldots i_d}^\top\boldsymbol{\Sigma}_{i_1i_2\ldots i_d}^{-1}\boldsymbol{\varepsilon}_{i_1\ldots i_d}+o_p(N^{-1+\tau-\upsilon})+o_p(a_N^{-\xi}N^{\tau-1}).
\end{aligned}$$

Since $\lambda=o_p(1)$, we have $N^{-1}\lambda\mathbf{I}=o_p(N^{-1})$ and $N^{-1}\lambda\boldsymbol{\gamma}_0=o_p(N^{-1})$. Then, we have

$$\begin{aligned}
&\tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}\widehat{\mathbf{W}}_{i_1\ldots i_d}^\top\boldsymbol{\Sigma}_{i_1i_2\ldots i_d}^{-1}\widehat{\mathbf{W}}_{i_1\ldots i_d}+\tfrac{1}{N}\lambda\mathbf{I}\\
=\;&\tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}\mathbf{W}_{i_1\ldots i_d}^\top\boldsymbol{\Sigma}_{i_1i_2\ldots i_d}^{-1}\mathbf{W}_{i_1\ldots i_d}+o_p(N^{-1+3\tau/2-\upsilon}),
\end{aligned}$$

and

$$\begin{aligned}
&\tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}\widehat{\mathbf{W}}_{i_1\ldots i_d}^\top\boldsymbol{\Sigma}_{i_1i_2\ldots i_d}^{-1}(\mathbf{y}_{i_1\ldots i_d}-\widehat{\mathbf{W}}_{i_1\ldots i_d}\boldsymbol{\gamma}_0)-\tfrac{1}{N}\lambda\boldsymbol{\gamma}_0\\
=\;&\tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}\mathbf{W}_{i_1\ldots i_d}^\top\boldsymbol{\Sigma}_{i_1i_2\ldots i_d}^{-1}\boldsymbol{\varepsilon}_{i_1\ldots i_d}+o_p(N^{-1+\tau-\upsilon})+o_p(a_N^{-\xi}N^{\tau-1}).
\end{aligned}$$

For any vector $\mathbf{c}$ whose components are not all zero, we have

$$\begin{aligned}
&\mathbf{c}^\top(\widehat{\boldsymbol{\gamma}}-\boldsymbol{\gamma}_0)\\
=\;&\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}\mathbf{c}^\top(\sum_{(i_1,\ldots,i_d)\in\Omega}\mathbf{W}_{i_1\ldots i_d}^\top\boldsymbol{\Sigma}_{i_1i_2\ldots i_d}^{-1}\mathbf{W}_{i_1\ldots i_d})^{-1}\mathbf{W}_{i_1\ldots i_d}^\top\boldsymbol{\Sigma}_{i_1i_2\ldots i_d}^{-1}\boldsymbol{\varepsilon}_{i_1\ldots i_d}+o_p(1)\\
=\;&\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}a_{i_1\ldots i_d}\zeta_{i_1\ldots i_d}+o_p(1),
\end{aligned}$$

where $\zeta_{i_1\ldots i_d}$ are independent with mean zero and variance one given $\{\mathbf{W}_{i_1\ldots i_d},(i_1,\ldots,i_d)\in\Omega\}$, and

$$\begin{aligned}
a_{i_1\ldots i_d}^2\;=\;&\mathbf{c}^\top(\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}\mathbf{W}_{i_1\ldots i_d}^\top\boldsymbol{\Sigma}_{i_1i_2\ldots i_d}^{-1}\mathbf{W}_{i_1\ldots i_d})^{-1}\mathbf{W}_{i_1\ldots i_d}^\top\boldsymbol{\Sigma}_{i_1i_2\ldots i_d}^{-1}\boldsymbol{\Sigma}_{i_1\ldots i_d}^{0}\boldsymbol{\Sigma}_{i_1i_2\ldots i_d}^{-1}\mathbf{W}_{i_1\ldots i_d}\\
&(\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega}\mathbf{W}_{i_1\ldots i_d}^\top\boldsymbol{\Sigma}_{i_1i_2\ldots i_d}^{-1}\mathbf{W}_{i_1\ldots i_d})^{-1}\mathbf{c}.
\end{aligned}$$

It follows easily by checking the Lindeberg condition that if

$$\frac{\max_{(i_1,\ldots,i_d)\in\Omega} a^2_{i_1\ldots i_d}}{\sum_{(i_1,\ldots,i_d)\in\Omega} a^2_{i_1\ldots i_d}} \to_p 0, \tag{13}$$

then $\sum_{(i_1,\ldots,i_d)\in\Omega} a_{i_1\ldots i_d}\zeta_{i_1\ldots i_d}/\sqrt{\sum_{(i_1,\ldots,i_d)\in\Omega} a^2_{i_1\ldots i_d}}$ is asymptotically $N(0,1)$. We only need to show that (13) holds.

Based on the properties of the B-splines and conditions (C3)-(C4), we have, for any $\boldsymbol{\phi} = (\boldsymbol{\phi}_1^\top, \ldots, \boldsymbol{\phi}_{r+m_{d+1}}^\top)^\top$ with $\boldsymbol{\phi}_j \in \mathbb{R}^M$,

$$\begin{aligned}
& \boldsymbol{\phi}^\top \mathbf{W}^\top_{i_1\ldots i_d} \boldsymbol{\Sigma}^{-1}_{i_1 i_2\ldots i_d} \boldsymbol{\Sigma}^{0}_{i_1\ldots i_d} \boldsymbol{\Sigma}^{-1}_{i_1 i_2\ldots i_d} \mathbf{W}_{i_1\ldots i_d} \boldsymbol{\phi} \\
\le\ & c_2^2 \sigma_1^{-2} \textstyle\sum_{t\in\mathbb{T}_{i_1\ldots i_d}} (\boldsymbol{\phi}^\top \mathbf{w}_{i_1\ldots i_d t})^2 \\
=\ & c_2^2 \sigma_1^{-2} \textstyle\sum_{t\in\mathbb{T}_{i_1\ldots i_d}} \{\sum_{j=1}^r u_{i_1 i_2\ldots i_d j} \boldsymbol{\phi}_j^\top \mathbf{B}_j(t) + u_{i_1 i_2\ldots i_d (r+1)} \sum_{e=1}^{m_{d+1}} \boldsymbol{\phi}_{r+e}^\top \mathbf{A}_e(t) I(t\in s_e)\}^2 \\
\le\ & c c_2^2 \sigma_1^{-2} \textstyle\sum_{t\in\mathbb{T}_{i_1\ldots i_d}} [(\sum_{j=1}^r u^2_{i_1 i_2\ldots i_d j}) \sum_{j=1}^r \{\boldsymbol{\phi}_j^\top \mathbf{B}_j(t)\}^2 + u^2_{i_1 i_2\ldots i_d (r+1)} \sum_{e=1}^{m_{d+1}} \{\boldsymbol{\phi}_{r+e}^\top \mathbf{A}_e(t)\}^2] \\
\lesssim\ & (\textstyle\sum_{j=1}^r u^2_{i_1 i_2\ldots i_d j}) \sum_{j=1}^r \sum_{t\in\mathbb{T}_{i_1\ldots i_d}} \{\boldsymbol{\phi}_j^\top \mathbf{B}_j(t)\}^2 + u^2_{i_1 i_2\ldots i_d (r+1)} \sum_{e=1}^{m_{d+1}} \sum_{t\in\mathbb{T}_{i_1\ldots i_d}} \{\boldsymbol{\phi}_{r+e}^\top \mathbf{A}_e(t)\}^2 \\
\lesssim\ & (\textstyle\sum_{j=1}^r u^2_{i_1 i_2\ldots i_d j}) \sum_{j=1}^r \frac{C_2}{M} |\boldsymbol{\phi}_j|^2 + u^2_{i_1 i_2\ldots i_d (r+1)} \sum_{e=1}^{m_{d+1}} \frac{C_2}{M} |\boldsymbol{\phi}_{r+e}|^2 \\
\lesssim\ & (\textstyle\sum_{j=1}^{r+1} u^2_{i_1 i_2\ldots i_d j}) \frac{C_2}{M} |\boldsymbol{\phi}|^2 \\
\lesssim\ & \tfrac{1}{M} |\boldsymbol{\phi}|^2.
\end{aligned}$$

By Lemmas A.1 and A.2 in Huang et al. (2004), we have

$$\begin{aligned}
& \boldsymbol{\phi}^\top (\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega} \mathbf{W}^\top_{i_1\ldots i_d} \boldsymbol{\Sigma}^{-1}_{i_1 i_2\ldots i_d} \boldsymbol{\Sigma}^{0}_{i_1\ldots i_d} \boldsymbol{\Sigma}^{-1}_{i_1 i_2\ldots i_d} \mathbf{W}_{i_1\ldots i_d}) \boldsymbol{\phi} \\
\ge\ & c_1^2 \sigma_2^{-2} \textstyle\sum_{(i_1,\ldots,i_d)\in\Omega} \boldsymbol{\phi}^\top \mathbf{W}^\top_{i_1\ldots i_d} \mathbf{W}_{i_1\ldots i_d} \boldsymbol{\phi} \\
\ge\ & c_1^2 \sigma_2^{-2} N T_{\min} [\tfrac{1}{N} \textstyle\sum_{(i_1,\ldots,i_d)\in\Omega} \frac{1}{|\mathbb{T}_{i_1\ldots i_d}|} \sum_{t\in\mathbb{T}_{i_1\ldots i_d}} \{\sum_{j=1}^r u_{i_1 i_2\ldots i_d j} \boldsymbol{\phi}_j^\top \mathbf{B}_j(t) \\
& + u_{i_1 i_2\ldots i_d (r+1)} \textstyle\sum_{e=1}^{m_{d+1}} \boldsymbol{\phi}_{r+e}^\top \mathbf{A}_e(t) I(t\in s_e)\}^2] \\
\ge\ & c_1^2 \sigma_2^{-2} N T_{\min} [\tfrac{1}{N} \textstyle\sum_{(i_1,\ldots,i_d)\in\Omega} \frac{1}{|\mathbb{T}_{i_1\ldots i_d}|} \sum_{t\in\mathbb{T}_{i_1\ldots i_d}} \{\sum_{j=1}^r u_{i_1 i_2\ldots i_d j} \boldsymbol{\phi}_j^\top \mathbf{B}_j(t) \\
& + \min_e \{u_{i_1 i_2\ldots i_d (r+1)} \boldsymbol{\phi}_{r+e}^\top \mathbf{A}_e(t)\}\}^2] \\
\gtrsim\ & N T_{\min} \tfrac{1}{M} |\boldsymbol{\phi}|^2.
\end{aligned}$$

Thus, we obtain that

$$\frac{\max_{(i_1,\ldots,i_d)\in\Omega} \boldsymbol{\phi}^\top \mathbf{W}^\top_{i_1\ldots i_d} \boldsymbol{\Sigma}^{-1}_{i_1 i_2\ldots i_d} \boldsymbol{\Sigma}^{0}_{i_1\ldots i_d} \boldsymbol{\Sigma}^{-1}_{i_1 i_2\ldots i_d} \mathbf{W}_{i_1\ldots i_d} \boldsymbol{\phi}}{\boldsymbol{\phi}^\top (\sum_{(i_1,\ldots,i_d)\in\Omega} \mathbf{W}^\top_{i_1\ldots i_d} \boldsymbol{\Sigma}^{-1}_{i_1 i_2\ldots i_d} \boldsymbol{\Sigma}^{0}_{i_1\ldots i_d} \boldsymbol{\Sigma}^{-1}_{i_1 i_2\ldots i_d} \mathbf{W}_{i_1\ldots i_d}) \boldsymbol{\phi}} \lesssim \frac{1}{N T_{\min}}.$$

Hence, (13) holds. Then we have $\mathbf{c}^\top(\widehat{\boldsymbol{\gamma}}-\boldsymbol{\gamma}_0) var\{\mathbf{c}^\top(\widehat{\boldsymbol{\gamma}}-\boldsymbol{\gamma}_0)\}^{-1/2} \xrightarrow{\mathrm{L}} N(0,1)$, where $var\{\mathbf{c}^\top(\widehat{\boldsymbol{\gamma}}-\boldsymbol{\gamma}_0)\} = \mathbf{c}^\top \boldsymbol{\Psi}^{-1} \boldsymbol{\Phi} \boldsymbol{\Psi}^{-1} \mathbf{c}$, in which $\boldsymbol{\Phi} = \sum_{(i_1,\ldots,i_d)\in\Omega} \mathbf{W}^\top_{i_1\ldots i_d} \boldsymbol{\Sigma}^{-1}_{i_1 i_2\ldots i_d} \boldsymbol{\Sigma}^{0}_{i_1\ldots i_d} \boldsymbol{\Sigma}^{-1}_{i_1 i_2\ldots i_d} \mathbf{W}_{i_1\ldots i_d}$, and $\boldsymbol{\Psi} = \sum_{(i_1,\ldots,i_d)\in\Omega} \mathbf{W}^\top_{i_1\ldots i_d} \boldsymbol{\Sigma}^{-1}_{i_1 i_2\ldots i_d} \mathbf{W}_{i_1\ldots i_d}$.

For any $\boldsymbol{\phi} = (\boldsymbol{\phi}_1^\top, \ldots, \boldsymbol{\phi}_{r+m_{d+1}}^\top)^\top$ with $\boldsymbol{\phi}_j \in \mathbb{R}^M$, under conditions (C3)-(C4), we have

$$\begin{aligned}
& \boldsymbol{\phi}^\top(\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega} \mathbf{W}_{i_1\ldots i_d}^\top \boldsymbol{\Sigma}_{i_1 i_2\ldots i_d}^{-1} \boldsymbol{\Sigma}_{i_1\ldots i_d}^{0} \boldsymbol{\Sigma}_{i_1 i_2\ldots i_d}^{-1} \mathbf{W}_{i_1\ldots i_d})\boldsymbol{\phi} \\
\leq\ & c_2^2\sigma_1^{-2} \textstyle\sum_{(i_1,\ldots,i_d)\in\Omega} \boldsymbol{\phi}^\top \mathbf{W}_{i_1\ldots i_d}^\top \mathbf{W}_{i_1\ldots i_d}\boldsymbol{\phi} \\
\leq\ & c_2^2\sigma_1^{-2} N T_{\max}[\tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega} \frac{1}{|\mathbb{T}_{i_1\ldots i_d}|} \sum_{t\in\mathbb{T}_{i_1\ldots i_d}} \{\sum_{j=1}^r u_{i_1 i_2\ldots i_d j} \boldsymbol{\phi}_j^\top \mathbf{B}_j(t) \\
& + u_{i_1 i_2\ldots i_d (r+1)} \textstyle\sum_{e=1}^{m_{d+1}} \boldsymbol{\phi}_{r+e}^\top \mathbf{A}_e(t) I(t\in s_e)\}^2] \\
\lesssim\ & N T_{\max} \tfrac{1}{M} |\boldsymbol{\phi}|^2,
\end{aligned}$$

and

$$\begin{aligned}
& \boldsymbol{\phi}^\top(\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega} \mathbf{W}_{i_1\ldots i_d}^\top \boldsymbol{\Sigma}_{i_1 i_2\ldots i_d}^{-1} \mathbf{W}_{i_1\ldots i_d})\boldsymbol{\phi} \\
\geq\ & c_1\sigma_2^{-2} N T_{\min}[\tfrac{1}{N}\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega} \frac{1}{|\mathbb{T}_{i_1\ldots i_d}|} \sum_{t\in\mathbb{T}_{i_1\ldots i_d}} (\boldsymbol{\phi}^\top \mathbf{w}_{i_1\ldots i_d t})^2] \\
\gtrsim\ & N T_{\min} \tfrac{1}{M} |\boldsymbol{\phi}|^2.
\end{aligned}$$

Since $M = a_N + \kappa + 1$, we have

$$\begin{aligned}
var\{\mathbf{c}^\top(\widehat{\boldsymbol{\gamma}} - \boldsymbol{\gamma}_0)\} &= \mathbf{c}^\top(\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega} \mathbf{W}_{i_1\ldots i_d}^\top \boldsymbol{\Sigma}_{i_1 i_2\ldots i_d}^{-1} \mathbf{W}_{i_1\ldots i_d})^{-1} \\
&\quad (\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega} \mathbf{W}_{i_1\ldots i_d}^\top \boldsymbol{\Sigma}_{i_1 i_2\ldots i_d}^{-1} \boldsymbol{\Sigma}_{i_1\ldots i_d}^{0} \boldsymbol{\Sigma}_{i_1 i_2\ldots i_d}^{-1} \mathbf{W}_{i_1\ldots i_d}) \\
&\quad (\textstyle\sum_{(i_1,\ldots,i_d)\in\Omega} \mathbf{W}_{i_1\ldots i_d}^\top \boldsymbol{\Sigma}_{i_1 i_2\ldots i_d}^{-1} \mathbf{W}_{i_1\ldots i_d})^{-1}\mathbf{c} \\
&= \mathbf{c}^\top \boldsymbol{\Psi}^{-1} \boldsymbol{\Phi} \boldsymbol{\Psi}^{-1} \mathbf{c} \\
&\lesssim \tfrac{M T_{\max}}{N T_{\min}^2} \lesssim \tfrac{a_N T_{\max}}{N T_{\min}^2} \lesssim a_N N^{-1+\tau-2\upsilon},
\end{aligned}$$

where

$$\boldsymbol{\Psi} = \sum_{(i_1,i_2,\ldots,i_d)\in\Omega} \mathbf{W}_{i_1 i_2\ldots i_d}^\top \boldsymbol{\Sigma}_{i_1 i_2\ldots i_d}^{-1} \mathbf{W}_{i_1 i_2\ldots i_d},$$

and

$$\boldsymbol{\Phi} = \sum_{(i_1,i_2,\ldots,i_d)\in\Omega} \mathbf{W}_{i_1 i_2\ldots i_d}^\top \boldsymbol{\Sigma}_{i_1 i_2\ldots i_d}^{-1} \boldsymbol{\Sigma}_{i_1 i_2\ldots i_d}^{0} \boldsymbol{\Sigma}_{i_1 i_2\ldots i_d}^{-1} \mathbf{W}_{i_1 i_2\ldots i_d}.$$

The proof of the theorem is complete. ■

References

Miguel Araujo, Pedro Ribeiro, Hyun Ah Song, and Christos Faloutsos. Tensorcast: Forecasting and mining with coupled tensors. *Knowledge and Information Systems*, 59(3): 497–522, 2019.

Xuan Bi, Annie Qu, and Xiaotong Shen. Multilayer tensor factorization with applications to recommender systems. *The Annals of Statistics*, 46(6B):3308–3333, 2018.

Bart J. Bronnenberg, Michael W. Kruger, and Carl F. Mela. Database paper—the iri marketing data set. *Marketing Science*, 27(4):745–748, 2008.

O. Celma. *Music Recommendation and Discovery in the Long Tail*. Springer, 2010.

Chris Chatfield. Calculating interval forecasts. *Journal of Business & Economic Statistics*, 11(2):121–135, 1993.

Bilian Chen, Simai He, Zhening Li, and Shuzhong Zhang. Maximum block improvement and polynomial optimization. *SIAM Journal on Optimization*, 22(1):87–107, 2012.

Gerda Claeskens, Tatyana Krivobokova, and Jean D. Opsomer. Asymptotic properties of penalized spline estimators. *Biometrika*, 96(3):529–544, 2009.

Carl de Boor. *A Practical Guide to Splines; rev. ed.* Springer, Berlin, 2001.

Vin de Silva and Lek-Heng Lim. Tensor rank and the ill-posedness of the best low-rank approximation problem. *SIAM Journal on Matrix Analysis and Applications*, 30(3):1084–1127, 2008.

Robin Devooght and Hugues Bersini. Long and short-term recommendations with recurrent neural networks. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*, UMAP '17, page 13–21, New York, NY, USA, 2017.

Van Than Dung and Tegoeh Tjahjowidodo. A direct method to solve optimal knots of b-spline curves: An application for non-uniform b-spline curves fitting. *PLOS ONE*, 12 (3):1–24, 2017.

Jianqing Fan and Wenyang Zhang. Statistical methods with varying coefficient models. *Statistics and Its Interface*, 1(1):179–195, 2008.

Evgeny Frolov and Ivan Oseledets. Tensor methods and recommender systems. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 7(3):e1201, 2017.

Michael Giering. Retail sales prediction and item recommendations using customer demographics at store level. *SIGKDD Explorations Newsletter*, 10(2):84–89, 2008.

San Gultekin and John Paisley. A collaborative kalman filter for time-evolving dyadic processes. In *Proceedings of the 2014 IEEE International Conference on Data Mining*, pages 140–149, Washington, DC, 2014.

Guibing Guo, Feida Zhu, Shilin Qu, and Xingwei Wang. Pccf: Periodic and continual temporal co-factorization for recommender systems. *Information Sciences*, 436-437:56–73, 2018.

Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22(1):5–53, 2004.

Charles C. Holt. Forecasting seasonals and trends by exponentially weighted moving averages. *International Journal of Forecasting*, 20(1):5–10, 2004.

Jianhua Z. Huang. Local asymptotics for polynomial spline regression. *The Annals of Statistics*, 31(5):1600–1635, 2003.

Jianhua Z. Huang and Haipeng Shen. Functional coefficient regression models for non-linear time series: A polynomial spline approach. *Scandinavian Journal of Statistics*, 31 (4):515–534, 2004.

Jianhua Z. Huang, Colin O. Wu, and Lan Zhou. Polynomial spline estimation and inference for varying coefficient models with longitudinal data. *Statistica Sinica*, 14(3):763–788, 2004.

Young-Jun Ko, Lucas Maystre, and Matthias Grossglauser. Collaborative recurrent neural networks for dynamic recommender systems. volume 63 of *Proceedings of Machine Learning Research*, pages 366–381, The University of Waikato, Hamilton, New Zealand, 16–18 Nov 2016.

Tamara G. Kolda and Brett W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

Yehuda Koren. Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 447–456, New York, NY, 2009.

Kung Yee Liang and Scott Zeger. Longitudinal data analysis using generalized linear models. *Biometrika*, 73(1):13–22, 1986.

Jinzhi Liao, Jiuyang Tang, Xiang Zhao, and Haichuan Shang. Improving poi recommendation via dynamic tensor completion. *Scientific Programming*, pages 1–11, 2018.

Dun Liu and Xiaoqing Ye. A matrix factorization based dynamic granularity recommendation with three-way decisions. *Knowledge-Based Systems*, 191:105243, 2020.

Amit Livne, Moshe Unger, Bracha Shapira, and Lior Rokach. Deep context-aware recommender system utilizing sequential latent context. *CoRR*, abs/1909.03999, 2019. URL `http://arxiv.org/abs/1909.03999`.

Xin Luo, Yunni Xia, and Qingsheng Zhu. Incremental collaborative filtering recommender based on regularized matrix factorization. *Knowledge-Based Systems*, 27:271–280, 2012.

Dimitrios Rafailidis and Alexandros Nanopoulos. Modeling the dynamics of user preferences in coupled tensor factorization. In *Proceedings of the 8th ACM Conference on Recommender Systems*, pages 321–324, New York, NY, 2014.

James Salter and Nick Antonopoulos. Cinemascreen recommender agent: combining collaborative and content-based filtering. *IEEE Intelligent Systems*, 21(1):35–41, 2006.

Nicholas D. Sidiropoulos and Rasmus Bro. On the uniqueness of multilinear decomposition of n-way arrays. *Journal of Chemometrics*, 14(3):229–239, 2000.

Jieun Son and Seoung Bum Kim. Content-based filtering for recommendation systems using multiattribute networks. *Expert Systems with Applications*, 89:404–412, 2017.

Dandan Song, Zhifan Li, Mingming Jiang, Lifei Qin, and Lejian Liao. A novel temporal and topic-aware recommender model. *World Wide Web*, 22(5):2105–2127, 2019.

W. Van Loock, G. Pipeleers, J. De Schutter, and J. Swevers. A convex optimization approach to curve fitting with b-splines. *IFAC Proceedings Volumes*, 44(1):2290–2295, 2011.

Jian Wei, Jianhua He, Kai Chen, Yi Zhou, and Zuoyin Tang. Collaborative filtering and deep learning based recommendation system for cold start items. *Expert Systems with Applications*, 69:29–39, 2017.

Peter R. Winters. Forecasting sales by exponentially weighted moving averages. *Management Science*, 6(3):324–342, 1960.

Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J. Smola, and How Jing. Recurrent recommender networks. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 495–503, New York, NY, 2017.

Xian Wu, Baoxu Shi, Yuxiao Dong, Chao Huang, and Nitesh V. Chawla. Neural tensor factorization for temporal interaction learning. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 537–545, New York, NY, 2019.

Liang Xiong, Xi Chen, Tzu-Kuo Huang, Jeff Schneider, and Jaime G Carbonell. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *Proceedings of the 2010 SIAM International Conference on Data Mining*, pages 211–222, 2010.

Lan Xue and Lijian Yang. Additive coefficient modeling via polynomial spline. *Statistica Sinica*, 16(4):1423–1446, 2006.

Hsiang-Fu Yu, Nikhil Rao, and Inderjit S Dhillon. Temporal regularized matrix factorization for high-dimensional time series prediction. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29, pages 847–855. Curran Associates, Inc., 2016.

Yuan Yuan, Nan Chen, and Shiyu Zhou. Adaptive b-spline knot selection using multi-resolution basis set. *IIE Transactions*, 45(12):1263–1277, 2013.

Chenyi Zhang, Ke Wang, Hongkun Yu, Jianling Sun, and Ee-Peng Lim. Latent factor transition for dynamic collaborative filtering. In *Proceedings of the 2014 SIAM International Conference on Data Mining*, pages 452–460, 2014.