


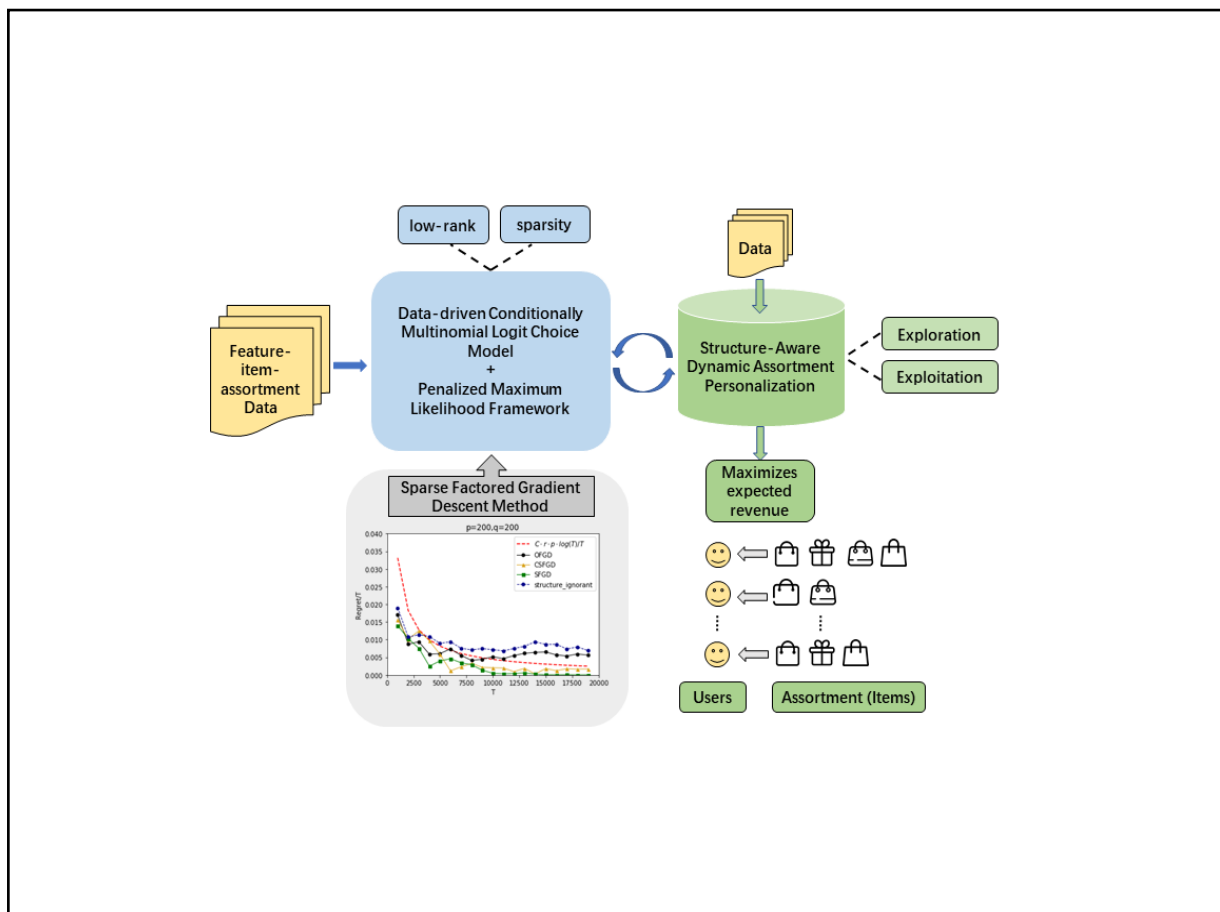
Sparse assortment personalization in high dimensions

Jingyu Shao, Ruipeng Dong , and Zemin Zheng

International Institute of Finance, School of Management, University of Science and Technology of China, Hefei 230026, China

Correspondence: Ruipeng Dong, E-mail: drp@mail.ustc.edu.cn

Graphical abstract



A penalized likelihood approach is proposed, in which the low-rank and sparsity structure are considered simultaneously. New algorithm sparse factored gradient descent (SFGD) is proposed to estimate the parameter matrix.


Public summary

- The data-driven conditional multinomial logit choice model with customer features has a good performance in assortment personalization problem when a low-rank structure of parameter matrix is considered.
- Our proposed method considers both low-rank and sparsity structure, which can further reduce model complexity and improve estimation and prediction accuracy.
- New algorithm sparse factored gradient descent (SFGD) is proposed to estimate the parameter matrix, which enjoys high interpretability and efficient performance in computing.

Sparse assortment personalization in high dimensions

Jingyu Shao, Ruipeng Dong , and Zemin Zheng

International Institute of Finance, School of Management, University of Science and Technology of China, Hefei 230026, China

 Correspondence: Ruipeng Dong, E-mail: drp@mail.ustc.edu.cn



Cite This: *JUSTC*, 2022, 52(3): 5 (11pp)



Read Online

Abstract: The data-driven conditional multinomial logit choice model with customer features performs well in the assortment personalization problem when the low-rank structure of the parameter matrix is considered. However, despite recent theoretical and algorithmic advances, parameter estimation in the choice model still poses a challenging task, especially when there are more predictors than observations. For this reason, we suggest a penalized likelihood approach based on a feature matrix to recover the sparse structure from populations and products toward the assortment. Our proposed method considers simultaneously low-rank and sparsity structures, which can further reduce model complexity and improve its estimation and prediction accuracy. A new algorithm, sparse factorial gradient descent (SFGD), was proposed to estimate the parameter matrix, which has high interpretability and efficient computing performance. As a first-order method, the SFGD works well in high-dimensional scenarios because of the absence of the Hessian matrix. Simulation studies show that the SFGD algorithm outperforms state-of-the-art methods in terms of estimation, sparsity recovery, and average regret. We also demonstrate the effectiveness of our proposed method using advertising behavior data analysis.

Keywords: assortment personalization; sparsity; penalized likelihood; factorial gradient descent; low-rank matrix approximation

CLC number: O212.1

Document code: A

2020 Mathematics Subject Classification: 62J12

1 Introduction

As an important part of revenue management, assortment planning has a wide range of applications in retail, advertising, and e-commerce. Personalization techniques are used to optimize the selection of products or services for certain customers. A key factor in optimizing assortment successfully is the ability to understand and predict the demand or customer preferences. Customer-specific data are available for companies in the scenario of many online applications. The feature information of customer data is of great significance in modelling the relationship between features and purchase decisions. In Refs.[1, 2], transactional data were used to estimate customer preferences. They rely on the discrete context of each customer type; that is, certain types of customers are discovered before the estimation. A practical algorithm for personalization under inventory constraints was proposed in Ref.[3]. The full feature data of customers are considered in Ref.[4], where covariate information was learned from the data.

Logit models are commonly used to better understand customer preferences and demands in practice. This is an advantage of interpretability and simplicity, which makes the logit model a popular choice. Such a framework is widely used in targeted advertising^[5], pricing^[6], and assortment personalization^[1,3,4]. The data-driven logit model framework, as a special type of generalized linear model, uses the information of customer features to estimate the coefficients, based on which assortment optimization is carried out. In big data applications, it is challenging to learn and infer dependence structures, because the responses and predictors in such a generalized

linear model (GLM) framework may be related through a few latent pathways or a subset of predictors. Furthermore, with the exponential growth in data volume, the curse of dimensionality and massive amounts of data make estimation and prediction more difficult to process. To successfully recover the sparse structure of predictors associated with the response, regularization methods such as lasso^[7], group lasso^[8], and group lasso for logistic regression^[9] are used.

In the multi-response scenario, the data-driven multinomial logit model tackles the associations between the predictors and responses via a sparse and low-rank representation of the coefficient matrix. Sparse reduced-rank regression has been extensively researched in the literature, which maintains the interpretability of the estimated matrix by eliminating irrelevant features, and the low-rank structure helps to reduce the number of free parameters of the model^[10-13]. Sparse reduced-rank regression has applications in social network community discovery^[14], subspace clustering^[15], and motion segmentation^[16]. In multitask learning and noisy matrix decomposition, there are abundant references that have considered the sparse reduced-rank representation; see Refs. [17-19] and also note that these references therein have estimated matrices with a low-rank plus sparse structure which is different from our work, as we focus on the estimation of a matrix that is jointly low-rank and sparse (see Refs. [10-12] for similar frameworks). Regarding the sparsity of the parameter matrix, Refs. [20,21] focused on the co-sparsity structure in the matrix. However, the sparsity in our assortment personalization problem aims to help select the features of customers, and row-wise sparsity is introduced. To the best of our know-

ledge, in the application of assortment personalization problem, the simultaneous sparse and low-rank structure in the coefficient matrix of the multinomial logit model have been rarely considered in the literature. To meet this requirement in our multinomial logit model, we choose the penalized likelihood framework.

To derive a sparse reduced-rank approximation of the parameter matrix, it is common to choose L_1 and nuclear norm regularizers. There are several methods for solving the penalized likelihood problem because of the convex relaxations to the sparsity and low rankness of a matrix. The resulting problem is convex and can be solved by the alternating direction method of multipliers (ADMM)^[22]; see Ref. [14]. Other methods include the sequential co-sparse unit-rank method^[20] and sparse eigenvalue decomposition^[23]. All the above sparse reduced-rank approaches have desirable theoretical properties. However, they cannot be directly used in penalized likelihood frameworks. For the GLM problem, the factored gradient descent method^[24] is commonly used in problems that can be posed as matrix factorization; see Ref. [25] for the precise convergence rate guarantees for a general convex function. Such a first-order method works in an alternative way and does not require the SVD of the parameter matrix at each step, which makes high-efficiency computation a possibility in solving the penalized likelihood problem.

The main contributions of this study are threefold. First, we provide the framework for the assortment personalization problem, which maximizes the expected revenue over a feasible assortment. We introduce customer features related to the utility model and present our data-driven conditional multinomial logit choice model. For the sparsity of the parameter matrix, we use a group lasso-type penalty to derive the row-wise sparsity, which is the same as the feature selection for customers. Thus, we make the estimation in the high-dimensional feature scenario available. Second, to solve the penalized maximum likelihood problem, we propose a first-order sparse factored gradient descent (SFGD) approach, in which both sparsity and low-rank structures are considered. Because of the low rank of the parameter matrix, the SVD can be used to reduce the number of parameters. We illustrate the details of the thresholding rule in SFGD and how it proceeds in the alternative updating of the two matrices derived from the decomposition. Moreover, we show the local convergence of SFGD and present a structure-aware dynamic assortment personalization procedure based on the SFGD method. Third, our simulation, which contains high-dimensional settings, shows that SFGD can consistently estimate the parameter matrix and accurately recover the support of the features. The average regret of different structure settings was compared with the growth of the time horizon, and the SFGD method with the sparse reduced-rank structure considered outperformed the sparsity structure-ignorant methods. We applied the proposed method to advertising behavior data, in which the features of both users and advertisements are considered. Furthermore, the SFGD based assortment personalization procedure exhibited the best precision.

2 Model specification

In this section, we present our modeling framework for data-

driven assortment personalization problems, in which customer features are considered. Throughout this paper, bold letters are used to denote the matrices and vectors. In this study, \mathbf{z}_i is the column vector of the i th row of \mathbf{Z} , and z_{ij} is the j element of the vector \mathbf{z}_i without special instructions. For any matrix $\mathbf{Z} = (z_{ij})$, denoted by $\|\mathbf{Z}\|_F = \sqrt{\sum_{i,j} z_{ij}^2}$, $\|\mathbf{Z}\|_{2,1} = \sum_i \|\mathbf{z}_i\|_2$ and $\|\mathbf{Z}\|_1 = \sum_{i,j} |z_{ij}|$ denotes the Frobenius norm, rows $l_{2,1}$ -norm and element-wise l_1 -norm. Furthermore, $\sigma_1(\mathbf{Z})$ is the largest singular value of \mathbf{Z} .

In the assortment personalization problem, the retailer records the observed transactional data in the past, which contains customer features, items (products) chosen by customers, and the assortment arrangement provided by the retailer. For time horizon T , the decision maker observes customer data in the past time $t = 1, \dots, T$. At time t , the decision maker obtains customer data \mathbf{x}_t with p features that include individual information, assortment $S_t \subset \{1, \dots, q\}$, and items $j_t \in \{1, \dots, q\}$, which were chosen by the t customer.

2.1 Data-driven conditionally multinomial logit choice model

In the data-driven assortment problem, we assume that the customer data matrix \mathbf{X} of size $T \times p$ is obtained directly from the past, also known as feature vectors. We assume that customers choose among the products according to some conditional probability $\mathbb{P}_{\boldsymbol{\theta}}(j|S)$ when the assortment is shown to the customer. Here, $\boldsymbol{\theta}$ is the parameter matrix that plays an important role in the conditional multinomial logit choice model. The choice of $\boldsymbol{\theta}$ will be presented later. For each item $j \in \{1, \dots, q\}$, let r_j be the associated revenue. Here, $r_0 = 0$ in revenue for the no-purchase option. Then, the decision-maker maximizes the expected revenue.

$$f(S) = \sum_{j \in S} r_j \mathbb{P}_{\boldsymbol{\theta}}(j|S) \quad (1)$$

over a feasible assortment $S \subset \{1, \dots, q\}$. The assortment personalization problem aims to find an assortment that maximizes the expected revenue.

$$\hat{S} = \underset{S \subset \{1, \dots, q\}}{\operatorname{argmax}} f(S)$$

To obtain a clear view of $\boldsymbol{\theta}$, we first introduce the utility of items. A popular way to model customer choice probability is to utilize the random utility model^[26]. We assume that a customer with the feature vector $\mathbf{x} \in \mathbb{R}^p$ has utility

$$U_j^x = V_j^x + \epsilon_j \quad (2)$$

for each product j , where V_j^x can be interpreted as the mean utility of product j for this customer and ϵ_j is a standard Gumbel random variable with a mean of zero. When a decision maker offers assortment $S \subset \{1, \dots, q\}$ to a customer with feature \mathbf{x} , the customer will choose the product in S with the highest U_j^x . The utility V_0^x of no-purchase option is to be zero. Here, we assume that the mean utility is given by the linear model $V_j^x = \langle \mathbf{x}, \boldsymbol{\theta}_j^* \rangle$, where $\boldsymbol{\theta}_j^* \in \mathbb{R}^p$ for $1 \leq j \leq q$. Hence, we obtain the mean utility matrix for all items $\mathbf{V}^x = \mathbf{X}\boldsymbol{\theta}^*$, where the underlying parameter matrix is $\boldsymbol{\theta}^* = (\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_q) \in$

$\mathbb{R}^{p \times q}$.

The data-driven conditional multinomial logit choice model is over time $t = 1, \dots, T$, and items $j = 1, \dots, q$. We introduce two random variables: customer I and item (choice) J . Using a well-known result from discrete choice theory^[27], given assortment $S \subset \{1, \dots, q\}$, we derive a personalized case of choice probability

$$\mathbb{P}_{\boldsymbol{\theta}^*}(J = j|S) = \frac{e^{V_j^x}}{1 + \sum_{j' \in S} e^{V_{j'}^x}} = \frac{\exp\{\langle \mathbf{x}, \boldsymbol{\theta}_j^* \rangle\}}{1 + \sum_{j' \in S} \exp\{\langle \mathbf{x}, \boldsymbol{\theta}_{j'}^* \rangle\}} \quad (3)$$

We choose the linear model of \mathbf{x} and $\boldsymbol{\theta}_j^*$ to represent V_j^x ; then, the choice J has the conditional distribution

$$\mathbb{P}_{\boldsymbol{\theta}^*}(J = j|I = \mathbf{x}_t; S) = \frac{1}{1 + \sum_{j' \in S} \exp(\mathbf{x}_t^T \boldsymbol{\theta}_{j'}^*)} \times \begin{cases} 1, & j = 0 \\ 0, & j \neq 0, j \notin S \\ \exp(\mathbf{x}_t^T \boldsymbol{\theta}_j^*), & j \neq 0, j \in S \end{cases} \quad (4)$$

where $J = 0$ indicates that no product has been purchased in assortment S . A no-purchase option is common in the choice model. In our data-driven framework, the decision maker can observe customer features $\mathbf{x}_t \in \mathbb{X}$, $t = 1, \dots, T$, where $\mathbb{X} \subset \mathbb{R}^p$ is a space of possible contexts. We also assume that \mathbf{x}_t is scaled to satisfy $\|\mathbf{x}_t\|_{\infty} \leq 1$, for $t = 1, \dots, T$.

2.2 Penalized maximum likelihood approach

We suppose that we have T observations (\mathbf{x}_t, j_t, S_t) for $t = 1, \dots, T$, where S_t comes from the set of subsets of $\{1, \dots, q\}$ of size K , and j_t are i. i. d., according to model (4). Based on the specific form of $\mathbb{P}_{\boldsymbol{\theta}^*}(J = j|I = \mathbf{x}_t; S)$ in (4), we define the loss function constructed from the negative log-likelihood as

$$\mathcal{L}(\mathbf{X}; \boldsymbol{\theta}) = \frac{1}{T} \sum_{t=1}^T \log \left(1 + \sum_{j \in S_t} e^{\mathbf{x}_t^T \boldsymbol{\theta}_j} (I_{(j_t=0)} + I_{(j_t \neq 0)} e^{\mathbf{x}_t^T \boldsymbol{\theta}_{j_t}})^{-1} \right) \quad (5)$$

Similar to classical methods, we often assume that the underlying parameter matrix $\boldsymbol{\theta}^*$ has a certain special structure, such as the low-rank structure of $\boldsymbol{\theta}^*$ and the sparsity of $\boldsymbol{\theta}^*$ ^[12]. In the customer choice model, it is reasonable to assume that for customer \mathbf{x} , only a few features have a significant impact on the utility of choosing different items. Because sparsity depends on the items, we introduce the row-wise sparsity of $\boldsymbol{\theta}^*$. To recover the sparse structure of the parameter matrix, the regularization method can be helpful for variable selection as well as sparsity recovery. In sparse reduced-rank learning, we tend to recover the sparsity and low-rank structures simultaneously. Choosing a large number of features is also a procedure for variable selection in our generalized multi-response regression problem. When large numbers of predictor variables (i.e., features) are available, some may not be helpful for both the estimation and prediction. Therefore, it is important to perform feature selection using the shrinkage method.

Inspired by the regularization method in regression, we chose a grouped lasso-type^[8,12] penalty to avoid overfitting and improve interpretability. Another widely used method is to

derive the sparsity in the matrix using an element-wise lasso penalty (see Refs. [28,29]). However, note that element-wise sparsity does not imply the row-wise sparsity that we expect to have, and the model will be unable to select the features of data \mathbf{X} . In our problem, setting the entire row of $\boldsymbol{\theta}$ to zero corresponds to excluding a feature from the customer data. Therefore, we introduced the $l_{2,1}$ norm of $\boldsymbol{\theta}$ rather than an element-wise l_1 norm. Let $1 \leq \tilde{r} \leq \min\{p, q\}$; then, we have the form of our problem as

$$\begin{aligned} & \text{minimize } Q(\mathbf{X}; \boldsymbol{\theta}) = \mathcal{L}(\mathbf{X}; \boldsymbol{\theta}) + \lambda \|\boldsymbol{\theta}\|_{2,1} \\ & \text{s.t. rank}(\boldsymbol{\theta}) \leq \tilde{r} \end{aligned} \quad (6)$$

In a full-rank problem, $\tilde{r} = \min\{p, q\}$ must be chosen. However, if the problem has a low-rank structure or if we want to enforce a low rank, then we use a proper choice of smaller \tilde{r} , reducing storage and computational work; see a similar motivation in Ref. [1]. If $\boldsymbol{\theta}^*$ has rank r , we may factor $\boldsymbol{\theta}^*$ to find the vectors $\mathbf{u}_i = (u_{i1}, \dots, u_{ir})^T$ and $\mathbf{v}_j = (v_{j1}, \dots, v_{jr})^T$ for $i = 1, \dots, p$ and $j = 1, \dots, q$ such that $\boldsymbol{\theta}_{ij}^*$ is approximately equal to $\sum_{l=1}^r u_{il} v_{jl}$. We denote $\mathbf{U} = (\mathbf{u}_i)_{p \times r}$ and $\mathbf{V} = (\mathbf{v}_j)_{q \times r}$; then, the right factors \mathbf{V} can be considered latent item weights, and the left factors \mathbf{U} as latent features^[1]. We now derive an appealing sparse SVD representation of $\mathbf{X}\boldsymbol{\theta}$ as $\frac{1}{\sqrt{T}} \mathbf{X}\boldsymbol{\theta} =$

$\left(\frac{1}{\sqrt{T}} \mathbf{X} \mathbf{U}_0 \mathbf{D}_0 \right) \mathbf{V}_0^T = \frac{1}{\sqrt{T}} \mathbf{X} \mathbf{U} \mathbf{V}^T$ where $\mathbf{U} = \mathbf{U}_0 \mathbf{D}_0 \in \mathbb{R}^{p \times r}$, $\mathbf{V} = \mathbf{V}_0 \in \mathbb{R}^{q \times r}$, $\mathbf{D}_0 = \text{diag}\{d_1^0, \dots, d_r^0\}$. This encourages us to use \mathbf{U} and \mathbf{V} to factorize the matrix $\boldsymbol{\theta}$; that is, $\boldsymbol{\theta} = \mathbf{U} \mathbf{V}^T$. Thus, it is feasible to introduce the first-order method over \mathbf{U} and \mathbf{V} to derive the estimation of $\boldsymbol{\theta}$. It is worth mentioning that our goal is to accurately provide a low-rank and row-sparse estimate of $\boldsymbol{\theta}$ rather than the estimates of \mathbf{U} and \mathbf{V} independently. This factorial form of $\boldsymbol{\theta}$ allows our method to approximate $\boldsymbol{\theta}$ alternately. A similar framework can be found in Refs. [1,25]. The tuning parameters λ , which are chosen based on an information criterion, are discussed later. We can shrink $\|\boldsymbol{\theta}_i\|_2$ to zero by setting the i th row of \mathbf{U} to zero and then derive the row-wise sparsity on \mathbf{U} and $\boldsymbol{\theta}$ accordingly. In our generalized multiresponse regression problem, all the items have probabilities to be chosen, which motivates us to introduce row-wise instead of column-wise sparsity.

We define our estimator $\hat{\boldsymbol{\theta}}$ for $\boldsymbol{\theta}^*$ as the solution to the maximum likelihood problem with the low-rank assumption $\text{rank}(\boldsymbol{\theta}^*) \ll \min\{p, q\}$. Because problem (6) is convex, we can apply a variety of convex methods. In the next section, we will attempt to use a first-order algorithm on the non-convex and factored forms.

3 Algorithm

With the convexity of $Q(\mathbf{X}; \boldsymbol{\theta})$, many fast optimization approaches, such as the alternating direction method of multipliers, accelerated projected gradient descent, and factored gradient descent, can perform well. The commonly used method for estimating a parameter matrix with a low-rank structure is the factored gradient descent method^[24]. In this section, we introduce a data-driven sparse factored gradient

descent (SFGD) algorithm to approximate θ^* using a low-rank and sparse structure. The SFGD is an interactive method in which the row-wise sparsity of U is considered and the updates of U and V overlap. In the update of U , we used a subgradient approach that cooperates with the gradient descent method.

3.1 Sparse factored gradient descent method

In a scenario of high dimensions, the computation of the Hessian matrix can be difficult or even not feasible. In this section, we introduce a first-order algorithm for computing $\hat{\theta}$, which works on the factored form of the low-rank and sparse constraint likelihood optimization problem (6). First, we consider the problem without regularization.

$$\begin{aligned} & \text{minimize } \mathcal{L}(X; UV^T) \\ & \text{s.t. } U \in \mathbb{R}^{p \times r}, V \in \mathbb{R}^{q \times r}, r \leq \tilde{r} \end{aligned} \quad (7)$$

It is clear that the algorithm reduces the computational cost because this model has only $r \times (p + q)$ optimization parameters, rather than $p \times q$. Moreover, our SFGD algorithm works in an alternative manner; that is, we optimize the factors U and V of the parameter matrix $\theta = UV^T$ rather than producing SVD at each step.

From the convexity of $\mathcal{L}(X; \theta)$ with respect to θ , it is feasible to use the gradient-descent method. Inspired by the factored form of our problem, we introduce the factored gradient descent procedure, which is a data-driven nonconvex method and a fundamental part of SFGD. The SFGD algorithm first solves the unconstrained problem (7) using the alternate updating rule

$$\begin{aligned} U' &= U - \eta \nabla_U \mathcal{L}(X; UV^T) \\ V' &= V - \eta \nabla_V \mathcal{L}(X; UV^T) \end{aligned} \quad (8)$$

which is closely related to the alternating convex search (ACS) method, as in Refs. [20,30]. The main difference between our SFGD is that U and V overlap with each other with rank $r \geq 1$, rather than the unit-rank problem. We begin the line search with a step size of $\eta = 1$, after which the adaptive step size is repeatedly decreased by a shrinkage factor β until the objective decreases.

It is easy to compute the gradients of the objective in (7). According to the chain rule of the differentiable function, we have

$$\begin{aligned} \nabla_U \mathcal{L}(X; UV^T) &= X^T \nabla \mathcal{L}(X; UV^T) V, \\ \nabla_V \mathcal{L}(X; UV^T) &= \nabla \mathcal{L}(X; UV^T)^T X U. \end{aligned}$$

Here, we do not need to explicitly form $\nabla \mathcal{L}(X; UV^T)$ to compute gradients. Recall that u and v are the $r \times 1$ column vectors of rows of U and V ; then, we have the following form of gradients:

$$\begin{aligned} \nabla_U \mathcal{L}(X; UV^T) &= \frac{1}{T} \sum_{t=1}^T \left(\frac{\sum_{j \in S_t} e^{x_{tj}^T U v_j} x_{tj} v_j^T}{1 + \sum_{j \in S_t} e^{x_{tj}^T U v_j}} - x_{tj} v_{j_t}^T \right), \\ \nabla_V \mathcal{L}(X; UV^T) &= \frac{1}{T} \sum_{t=1}^T \left(\frac{\sum_{j \in S_t} e^{x_{tj}^T U v_j} e_j x_{tj}^T}{1 + \sum_{j \in S_t} e^{x_{tj}^T U v_j}} - e_{j_t} x_{tj}^T \right) U, \end{aligned}$$

which clarifies the gradient descent direction. See Appendix A for more details.

We now introduce the row-wise sparsity of θ . To solve this problem, in the (m) th step, we used the subgradient method to screen the rows of $U^{(m)}$, which aims to find sparsity when $\|u_i^{(m)}\|_2 = 0$. The j th element of x_t is denoted as x_{tj} . For any $i = 1, \dots, p$, we use the subgradient method with respect to $u_i^{(m)}$ and let the subgradient be zero, which leads to

$$\frac{1}{T} \sum_{t=1}^T \left(\frac{\sum_{j \in S_t} e^{x_{tj}^T U^{(m)} v_j} x_{tj} v_j^T}{1 + \sum_{j \in S_t} e^{x_{tj}^T U^{(m)} v_j}} - x_{tj} v_{j_t}^T \right) + \lambda \frac{u_i^{(m)}}{\|u_i^{(m)}\|_2} = 0 \quad (9)$$

Let $s_i = \frac{u_i^{(m)}}{\|u_i^{(m)}\|_2}$ if $\|u_i^{(m)}\|_2 \neq 0$. Further, s_i is an r vector satisfying $\|s_i\|_2 < 1$ if $\|u_i^{(m)}\|_2 = 0$; then, we have

$$s_i = -\frac{1}{\lambda T} \sum_{t=1}^T \left(\frac{\sum_{j \in S_t} e^{x_{tj}^T U^{(m)} v_j} x_{tj} v_j^T}{1 + \sum_{j \in S_t} e^{x_{tj}^T U^{(m)} v_j}} - x_{tj} v_{j_t}^T \right) \quad (10)$$

We present the SFGD details as follows

- (i) For the $(m+1)$ th repeat in gradient descent, we screen the row-wise sparsity before finally updating $U^{(m)}$.
- (ii) For $i = 1, \dots, p$, denote x_{tj} for the j th element of x_t , then compute s_i . Update the j th row of $U^{(m)}$ using the threshold rule

$$u_i^{(m+1)} = \frac{1}{\|s_i\|_2 - 1} (\|s_i\|_2 - 1)_+ u_i^{(m)},$$

where $(z)_+ = \max\{0, z\}$ for all $z \in \mathbb{R}$. Without loss of generality, if $\|s_i\|_2 - 1 = 0$, we let $u_i^{(m+1)} = u_i^{(m)}$.

- (iii) After screening for all rows of $U^{(m)}$, update $U^{(m)}$ derive $U^{(m+1)}$ and then enter the next iteration or stop.

Our SFGD method can be initialized using the technique from Ref. [25], which only requires gradients of $\mathcal{L}(X; \theta)$. By the SVD of $-\nabla \mathcal{L}(X; \mathbf{0})$, it entails $-\nabla \mathcal{L}(X; \mathbf{0}) = \tilde{U} \text{diag}(\tilde{\sigma}_1, \dots, \tilde{\sigma}_{\min(p,q)}) \tilde{V}^T$. We denote by \tilde{U}_r and \tilde{V}_r the first r columns of \tilde{U} and \tilde{V} , where $r \leq \tilde{r}$ is one of the tuning parameters. Let E_1 be a $p \times q$ matrix that has a value of one in the $(1, 1)$ element with other zeros. Then we initialize

$$U^0 = \omega^{-1/2} \text{diag}(\sqrt{\tilde{\sigma}_1}, \dots, \sqrt{\tilde{\sigma}_r}) \tilde{U}_r,$$

$$V^0 = \omega^{-1/2} \text{diag}(\sqrt{\tilde{\sigma}_1}, \dots, \sqrt{\tilde{\sigma}_r}) \tilde{V}_r,$$

where $\omega = \|\nabla \mathcal{L}(X; \mathbf{0}) - (\nabla \mathcal{L}(X; E_1) + \lambda E_1)\|_F$. Besides, the termination of our algorithm is met when the decrease in the ob-

jective function value is smaller than the tolerance τ .

The selection of the tuning parameter λ was based on the information criterion which will be discussed later. Considering the overshooting problem in the line search process, the step size shrinkage factor β adjusts η to ensure that the local optimal is not missed. The details of SFGD are presented in Algorithm 3.1.

Algorithm 3.1 Sparse factored gradient descent (SFGD)

Input Feature, item and assortment data $\{(\mathbf{x}_t, j_t, S_t)\}_{t=1}^T$; dimensions of $\boldsymbol{\theta}$: p, q , tuning parameters (λ, r) , step size shrinkage factor β , and tolerance τ . $\mathbf{U} \leftarrow \mathbf{U}^0, \mathbf{V} \leftarrow \mathbf{V}^0, f' \leftarrow \infty$.

```

1 repeat
2    $\eta \leftarrow 1, f \leftarrow f', \Delta \mathbf{U} \leftarrow -\lambda \mathbf{U}, \Delta \mathbf{V} \leftarrow -\lambda \mathbf{V}$ 
3   for  $t = 1, \dots, T$  do
4     for  $j \in S_t$  do
5        $w_j \leftarrow e^{\mathbf{x}_t^T \mathbf{v}_j}, W \leftarrow W + w_j$ 
6     end for
7      $\Delta \mathbf{U} \leftarrow \Delta \mathbf{U} - \frac{1}{N}(\mathbf{x}_t \mathbf{v}_j^T - \frac{1}{W} \sum_{j \in S_t} w_j \mathbf{x}_t \mathbf{v}_j^T)$ 
8      $\Delta \mathbf{V} \leftarrow \Delta \mathbf{V} - \frac{1}{N}(\mathbf{e}_j \mathbf{x}_t^T - \frac{1}{W} \sum_{j \in S_t} w_j \mathbf{e}_j \mathbf{x}_t^T) \mathbf{U}$ 
9   end for
10  repeat
11     $\mathbf{U}' \leftarrow \mathbf{U} + \eta \Delta \mathbf{U}, \mathbf{V}' \leftarrow \mathbf{V} + \eta \Delta \mathbf{V}$ 
12     $f' \leftarrow \mathcal{L}(\mathbf{X}; \mathbf{U}' \mathbf{V}'^T) + \lambda \|\mathbf{U}' \mathbf{V}'^T\|_{2,1}$ 
13     $\eta \leftarrow \beta \eta$ 
14  until  $f' \leq f$ 
15  for  $i = 1, \dots, p$  do
16     $\mathbf{u}'_i = \frac{1}{\|\mathbf{s}_i\|_2 - 1}(\|\mathbf{s}_i\|_2 - 1) \mathbf{u}'_i$  with  $\mathbf{s}_i$  defined in (10)
17  end for
18   $\mathbf{U} \leftarrow \mathbf{U}', \mathbf{V} \leftarrow \mathbf{V}'$ 
19 until  $\frac{f - f'}{f'} \leq \tau$ 
Output  $\hat{\boldsymbol{\theta}} = \mathbf{U} \mathbf{V}^T$ 

```

3.2 Local convergence of SFGD

Now, we provide the convergence performance of the SFGD algorithm as follows: Appendix B provides the proof.

Theorem 3.1. Let $\boldsymbol{\theta} = \mathbf{U} \mathbf{V}^T$ and $\boldsymbol{\theta}' = \mathbf{U}' \mathbf{V}'^T$ denote the input and output of an iteration of SFGD. Then there exist a constant $M > 0$, if the step size $\eta \leq \frac{1}{3M(\sigma_1(\mathbf{U}^T \mathbf{U}) + \sigma_1(\mathbf{V}^T \mathbf{V}))}$, then

$$Q(\mathbf{X}; \boldsymbol{\theta}') \leq Q(\mathbf{X}; \boldsymbol{\theta}) - \frac{5}{6} \eta \|\nabla \mathcal{L}(\mathbf{X}; \boldsymbol{\theta})\|^T \mathbf{U} \mathbf{U}^T \quad (11)$$

There exists an optimum $\tilde{\boldsymbol{\theta}} = \tilde{\mathbf{U}} \tilde{\mathbf{V}}^T$ such that $Q(\mathbf{X}; \boldsymbol{\theta})$ converges to $Q(\mathbf{X}; \tilde{\boldsymbol{\theta}})$.

The theorem states that the step size η adjusted by the shrinkage factor is always below the upper bound. Updating by such a step size ensures that $Q(\mathbf{X}; \boldsymbol{\theta})$ decreases towards the optimum value.

3.3 The structure-aware dynamic assortment personalization problem

In the scenario of the dynamic assortment personalization

problem, we learn from the past until the time horizon T , first by affording random assortments S_t and recording the observations (\mathbf{x}_t, j_t, S_t) , which is the exploration procedure. In the next step, we implement the SFGD algorithm to estimate $\boldsymbol{\theta}$ using both low-rank and sparse structures. With an increase in T , the in-sample prediction of the assortment can be derived by maximizing expected revenue (1). For the given p, q , and r , there is a critical value as a function $C(T)$ that depends on T , such as $C(T) = Cr(p+q)\log(T)$ in Ref. [1]. When T meets $C(T)$, the problem becomes exploitation, which is the out-of-sample prediction of the assortment. We denote by \mathcal{A} the collection of observations, and $C(T)$ slowly varies with respect to T . We then present the details of our dynamic assortment personalization problem in Algorithm 3.2.

After the exploration step, it yields the structure-aware estimate $\hat{\boldsymbol{\theta}}$, and based on $\hat{\boldsymbol{\theta}}$, we derive the conditional distribution using (4) with respect to the new data \mathbf{x}_t . Now, we can see that our structure-aware dynamic assortment personalization approach serves every incoming individual at time t rather than several types of customers, as in Ref. [1].

Algorithm 3.2 Structure-aware dynamic assortment personalization

```

Input  $C(T), \lambda$ 
1  $\mathcal{A} \leftarrow \emptyset$ 
2 for  $T = 1, 2, \dots$  do
3    $t \leftarrow T$ 
4   if  $T \leq C(T)$  then
5     Exploration:
6     choose  $S_t$  uniformly at random from  $\{1, \dots, q\}$ 
7     of size  $K$ ,
8     observe  $(\mathbf{x}_t, j_t, S_t)$  and  $\mathcal{A} \leftarrow \mathcal{A} \cup (\mathbf{x}_t, j_t, S_t)$ ,
9      $\mathcal{L}(\mathbf{X}; \boldsymbol{\theta}) = \frac{1}{T} \sum_{(\mathbf{x}_t, j_t, S_t) \in \mathcal{A}} \log \frac{1 + \sum_{j \in S_t} e^{\mathbf{x}_t^T \boldsymbol{\theta}_j}}{I_{(j_t=0)} + I_{(j_t \neq 0)} e^{\mathbf{x}_t^T \boldsymbol{\theta}_{j_t}}}$ ,
10     $\hat{\boldsymbol{\theta}} \in \{\boldsymbol{\theta} : \argmax Q(\mathbf{X}; \boldsymbol{\theta}), \text{ s.t. rank}(\boldsymbol{\theta}) \leq \tilde{r}\}$ 
11  else
12    Exploitation:
13     $S_t \in \{S : \argmax_{S \subseteq \{1, \dots, q\}} \sum_{j \in S} r_j \mathbb{P}_{\hat{\boldsymbol{\theta}}}(j|S)\}$ 
14  end if
15 Output  $S_1, \dots, S_T$ 

```

4 Simulation studies

In this section, we describe the implementation of the simulation to demonstrate the advantages of the proposed approach. We use the generalized information criterion (GIC)^[31] for high-dimensional penalized likelihood settings to select the tuning parameter λ and rank r by minimizing

$$\text{GIC}_{a_T}(\lambda, r) = \frac{1}{T} \{\mathcal{L}(\mathbf{X}; \hat{\boldsymbol{\theta}}_{\lambda, r}) + a_T |\alpha_\lambda|\},$$

where $\alpha_\lambda \subset \{1, \dots, p\}$ is the row-wise support for estimate $\hat{\boldsymbol{\theta}}$. Denote by α_0 the true row-wise support of $\hat{\boldsymbol{\theta}}$. Then, there exists a λ_0 such that $\alpha_{\lambda_0} = \alpha_0$. In addition, a_T is a positive sequence that depends only on T . We choose a modified BIC-type a_T such that $a_T = C_T \log(T)$ with a diverging C_T sequence,

as in Ref. [32]. In this study, we used this strategy by letting $C_T = c \log(\log(T + p + q))$, where c is a positive constant. In the following analysis, we select λ and r by minimizing $\text{GIC}_{ar}(\lambda, r)$.

4.1 Estimation accuracy

First, we generate true Θ^* as follows: First, we generate the $p \times q$ matrix Θ_0 from the elemental standard normal, take the SVD of Θ_0 as $\Theta_0 = U \text{diag}(\sigma_1, \sigma_2, \dots) V^T$, reserve the first r singular values, and derive $\Theta_1 = U \text{diag}(\sigma_1, \dots, \sigma_r, 0, \dots, 0) V^T$. Then, $\Theta_2 = \Theta_1 / \text{sd}(\text{vec}(\Theta_1))$. Finally, we derive row-wise sparsity by randomly choosing $S \subseteq \{1, \dots, p\}$ and $|S| = s$ as the corresponding non-sparse rows of Θ_2 with other row zeros, which yields Θ^* . Let customer data X be drawn from the normal distribution $N(0, \Sigma)$, where $\Sigma = (\sigma_{ij})_{p \times p}$ with $\sigma_{ij} = 0.5^{|i-j|}$, assortments $S_t, t = 1, \dots, T$ be uniformly drawn from $\{1, \dots, q\}$ with subset size $K = 10$, and then derive j_t according to the conditional distribution, as demonstrated in (4). We considered different settings of true rank $r = 2, 3, 5$ with $\tilde{r} = 2r$, $s = 10$, $\tau = 10^{-10}$, $\eta = 0.05$, and $c = 5$. To tune the parameter selection, we minimize $\text{GIC}_{ar}(\lambda, r)$ for every fixed value of $r \leq \tilde{r}$, and then we finish the tuning by choosing (λ, r) that minimize $\text{GIC}_{ar}(\lambda, r)$ globally. In a real-world application, GIC will in turn help approximate the upper bound \tilde{r} of the low-rank constraint because when the rank reaches a certain value and after, the GIC value will hardly change with different λ .

In the method comparison, we considered the ordinary factored gradient descent (OFGD) method, which solves problem (7) using the alternative updating rule (8). OFGD recovers only the low-rank structure of Θ . Moreover, we introduce the maximum likelihood estimation (MLE) method with the structure-free Θ ; that is, both sparse and low-rank structures of Θ are ignored, and the rank of Θ is chosen as $\min\{p, q\}$.

The error of estimation is measured by root mean squared error (RMSE)

$$\text{RMSE}(\Theta) = \frac{1}{\sqrt{pq}} \|\Theta - \Theta^*\|_F.$$

To evaluate the utility error, as declared in Eq. (2), we introduce $\text{Er}(X\Theta)$, defined by

$$\text{Er}(X\Theta) = \frac{1}{\sqrt{Tq}} \|\Sigma^{\frac{1}{2}}(\Theta - \Theta^*)\|_F.$$

Moreover, we choose two indicators: the false positive $\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}$ and false negative rate $\text{FNR} = \frac{\text{FN}}{\text{FN} + \text{TP}}$ to evaluate the results of sparsity recovery, in which for Θ_i , if $\Theta_i^* = 0$, but $\hat{\Theta}_i \neq 0$, then i goes into the counter of FP, and FN, TN, TP are calculated by analogy. We also introduce the number of iterations k , which sums the total updating times of the alternate updating rule (8) in the SFGD with a proper choice of step size η .

In Table 1, we compare the performance of OFGD, SFGD, and MLE in 100 replications and report the results in a high-dimensional setting with $p > T$. As reported in Table 1, under the setting $c = 5$ in the tuning of λ , the structure-aware SFGD method outperforms all the other methods in terms of the error of both estimation and utility. Moreover, as expected,

the OFGD method, which only considers the low-rank structure, performs better than the structure-ignorant MLE method. The SFGD has the ability in sparsity recovery because FPR and FNR are well controlled. In a high-dimensional setting when $p > T$, the SFGD maintains the performance of estimation accuracy as well as sparse recovery. For different settings of true rank r , we find that SFGD still enjoys the lowest RMSE and has good control of the FNR, which indicates the robustness of our methods for different structures of Θ . The CPU time and number of iterations k are also reported in Table 1, from which we determine the efficiency of our sparse reduced rank method SFGD in both low- and high-dimensional settings.

4.2 Regret for low-rank and sparse structure

Next, we consider the dynamic assortment personalization problem. We compared the average regret^[33] of the three methods. One alternative is the structure-ignorant algorithm, in which we fit a single MNL model by MLE to the entire population without the low-rank and sparse structure on Θ . We first define the average regret as follows:

Definition 4.1. Given an instance (p, q, Θ^*) , the average regret of algorithm π at time T is

$$\text{AveRegret}(T; \pi) = \mathbb{E}^{\pi, \Theta^*} \left[\frac{1}{T} \sum_{t=1}^T r_t \right] - \mathbb{E}^{\pi} \left[\frac{1}{T} \sum_{t=1}^T r_t \right].$$

In the simulation of the average regret, the feature matrix X and underlying Θ^* are generated based on the previous simulation of the estimation accuracy. We now construct the true revenue for each product as follows:

(i) K out of q items have revenue parameters $r_i = 1$.

(ii) For the other $(q - K)$ items, both revenues are uniformly distributed in $[0.05, 0.1]$.

For comparison, we also introduce the average regret $O\left(r \max(p, q) \frac{\log(T)}{T}\right)$ of Ref. [1] as a baseline that considers customer types rather than feature data.

In Fig. 1, we report all results for different settings of p, q , and T after 100 replications. From Fig. 1, we can see that SFGD has a lower average regret than both OFGD and MLE, which means that both low-rank and sparse structures reduce the regret level. Moreover, with the growth of types p and items q , the consideration of both low-rank and sparse structures is closer to the baseline.

We found that the SFGD method stabilized at a mean regret level that was much lower than that of the OFGD and MLE methods. Before reaching the minimum regret level, with an increase in the time horizon T , the average regret will decrease for SFGD, whereas for OFGD and MLE, the regret will not decrease with a larger T . Furthermore, in all settings, the OFGD method that only uses the low-rank structure achieves a better performance than the structure-ignorant MLE. Therefore, our results confirm the necessity of sparse recovery and the effectiveness of our proposed algorithm for the dynamic-assessment personalization problem.

The structure-aware method, which contains low-rank and sparsity structures, is of great significance when handling large-scale customer feature data, especially in high-dimensional scenarios $p \geq T$. We also observe the effective variable selection capability of the grouped lasso-type shrinkage method on Θ , which is computed iteratively using our pro-

Table 1. Results in methods OFGD, SFGD, and MLE with different r, p, q, T settings, 100 replications (standard deviations are shown in parentheses).

r	Method	RMSE	$\text{Er}(X\theta)$	FPR %	FNR %	CPU time	k
2			$p = 50, q = 25;$	$T = 400$			
	OFGD	2.1656(0.2698)	1.6229(0.3595)	100(0)	0(0)	13.63(0.22)	139(9)
	SFGD	0.6438(0.0288)	0.2938(0.0181)	3.42(1.24)	0(0)	15.70(1.17)	123(11)
	MLE	6.1391(0.4325)	4.7971(0.3621)	100(0)	0(0)	43.82(3.35)	471(8)
			$p = 100, q = 100;$	$T = 200$			
	OFGD	1.9767(0.1276)	2.8220(0.4002)	100(0)	0(0)	12.18(0.22)	161(12)
	SFGD	0.4858(0.0736)	0.5721(0.0542)	1.98(0.46)	0(0)	14.31(1.45)	155(10)
	MLE	10.4705(0.5249)	13.8533(0.6313)	100(0)	0(0)	76.82(5.25)	509(11)
			$p = 300, q = 100;$	$T = 200$			
3	OFGD	1.8704(0.0902)	4.6054(0.3512)	100(0)	0(0)	20.35(4.13)	228(10)
	SFGD	0.3866(0.0873)	0.8175(0.0463)	2.17(0.31)	1.35(0.47)	44.78(7.48)	222(12)
	MLE	13.2192(0.6139)	16.8519(0.8791)	100(0)	0(0)	133.37(9.21)	716(10)
			$p = 50, q = 25;$	$T = 400$			
	OFGD	2.5649(0.2405)	2.4220(0.2886)	100(0)	0(0)	18.73(0.64)	141(8)
	SFGD	0.4418(0.0640)	0.2427(0.0571)	2.50(0.58)	0(0)	18.08(2.12)	131(12)
	MLE	6.1815(0.4265)	4.7051(0.3445)	100(0)	0(0)	48.55(3.68)	496(11)
			$p = 100, q = 100;$	$T = 200$			
	OFGD	2.3732(0.4183)	3.4952(0.7210)	100(0)	0(0)	13.75(0.14)	150(13)
5	SFGD	0.6663(0.0275)	0.6322(0.0393)	2.58(0.64)	0(0)	14.35(1.35)	143(10)
	MLE	10.3445(0.4535)	13.4604(0.6637)	100(0)	0(0)	72.76(5.44)	515(11)
			$p = 300, q = 100;$	$T = 200$			
	OFGD	2.2845(0.6149)	5.3920(0.6399)	100(0)	0(0)	23.30(4.09)	225(11)
	SFGD	0.6154(0.0964)	0.9119(0.0742)	2.05(0.29)	1.34(0.48)	41.92(6.22)	219(8)
	MLE	13.6123(0.6734)	16.9354(0.7346)	100(0)	0(0)	136.75(9.61)	724(9)
			$p = 50, q = 25;$	$T = 400$			
	OFGD	2.7600(0.0411)	2.0115(0.0933)	100(0)	0(0)	19.41(0.65)	164(12)
	SFGD	0.9718(0.0607)	0.6560(0.0709)	1.02(0.32)	0(0)	21.96(4.62)	157(11)
5	MLE	6.1952(0.5854)	4.7553(0.3933)	100(0)	0(0)	49.01(3.59)	503(13)
			$p = 100, q = 100;$	$T = 200$			
	OFGD	2.5851(0.0226)	3.3536(0.0705)	100(0)	0(0)	20.24(0.34)	169(7)
	SFGD	0.8363(0.0293)	0.9026(0.0688)	1.31(0.50)	0(0)	22.59(2.65)	161(11)
	MLE	10.4693(0.5535)	14.0481(0.7380)	100(0)	0(0)	73.83(3.87)	511(9)
			$p = 300, q = 100;$	$T = 200$			
	OFGD	2.9147(0.1480)	6.3237(0.4272)	100(0)	0(0)	21.42(4.85)	237(12)
	SFGD	0.7256(0.0941)	0.9880(0.0612)	2.13(0.35)	1.33(0.41)	48.75(8.43)	227(10)
	MLE	13.8753(0.6278)	17.9641(0.9625)	100(0)	0(0)	141.64(8.54)	741(11)

posed SFGD method. Furthermore, with the growth of horizon T , SFGD always enjoys the lowest average regret among different algorithms.

5 Application to advertising behavior data

This section analyzes the advertising behavior data collected

on seven consecutive days, which contain the features of users, available at Kaggle ^①. Target advertising^[5,34] is a key problem in advertising computation. Increasing the accuracy of personal advertising is crucial for improving the effectiveness of precision marketing. We will analyze the advertising behavior dataset, which contains both the information of users and advertisements. The advertising dataset has 10000

① <https://www.kaggle.com/>

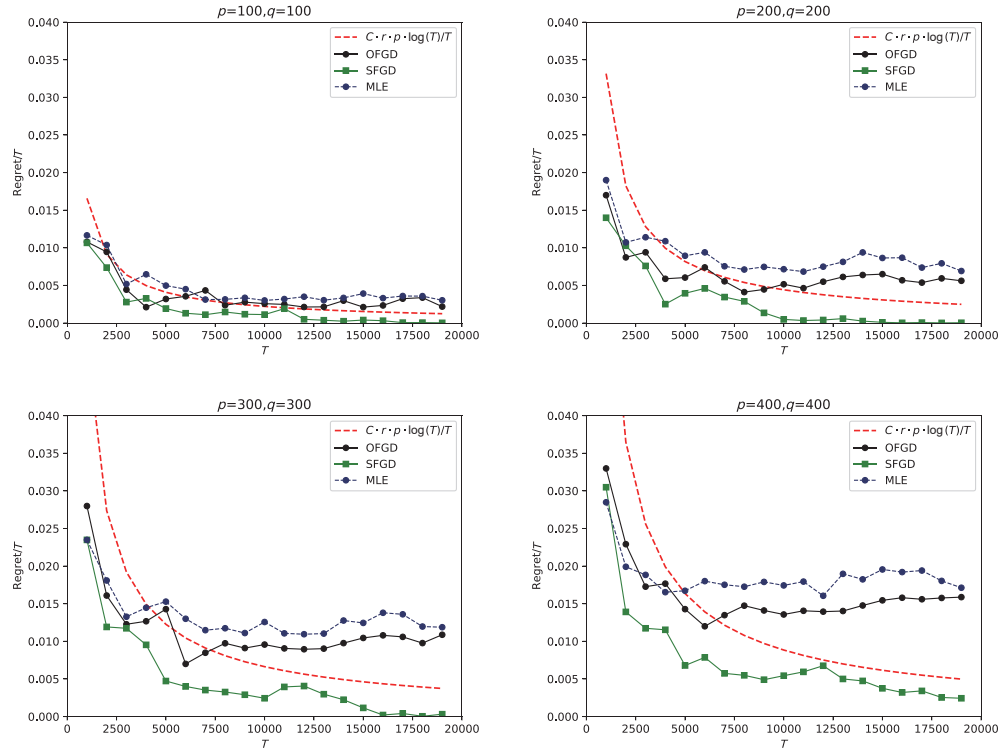


Fig. 1. Comparison of average regret between our proposed methods OFGD, SFGD, and MLE. The time horizon T ranges from 1000 to 20000. The constant of the baseline is chosen as $C = 0.12$.

records of advertisers' information offered, attributes of the advertisements, attributes of the users, and advertisements clicked by users. We consider 30 features of users that have interaction effects with the click behavior such as user age; city rank: level of the resident city of a user; device name: phone model used by a user; career; gender; net type: network status when a behavior occurs; residence: resident province of a user; App storage size; release time; app rating score; device price; active time by mobile phone; and membership lifecycle. To facilitate the computation of discrete variables in Euclidean space, we introduce one-hot encoding and finally obtain $p = 501$ features. There are $q = 113$ advertisers' types of ads clicked by users, denoted by $j_i \in \{1, \dots, q\}$.

We begin by splitting our data into a training set of 70% records and a test set of 30% records. We fit and evaluate our model 100 times over each setting of the training set sizes $T \in \{200, 700, 1700, 3000, 5000, 7000\}$ using the following steps. First, we randomly selected T users from the training pool and selected the tuning parameters λ and r using our GIC method. Noting that the value of the GIC function rarely changes when $r > 5$, we set $\tilde{r} = 5$. We then fit the model to this training set of size T . Finally, we tested its performance on a fully held-out test set with a size of 3000.

According to the structure-aware dynamic assortment personalization procedure in Algorithm 3.2, in the exploration stage, we fit the model and provide a sparse reduced-rank representation of θ . Without additional knowledge, we treat the rewards of all the items(ads) equally by letting $r_1 = r_2 = \dots = r_q$. Then, in the exploitation stage, we assign a size of $K = 10$ to every user in the test set by maximizing the expected revenue. To evaluate our model, we use precision,

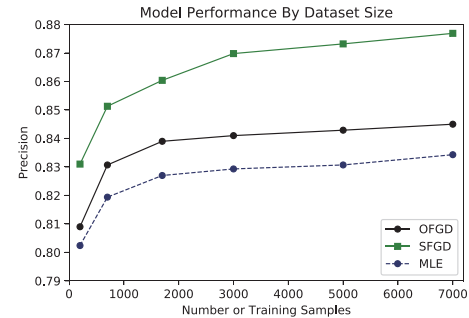


Fig. 2. Model performance by dataset size.

which is the percentage of users' click behavior of advertisers' types j_i successfully covered by the predicted assortment S_i . We benchmark our SFGD method with OFGD and MLE as in the simulations; the results are shown in Fig. 2.

From Fig. 2, we observe that the precision of assortment personalization by SFGD increases from 83.1% to 87.7% as the size of the training set increases from 200 to 7000. It is worth mentioning that our SFGD approach reaches 83.1% precision under a high-dimensional scenario when $T = 200$, $p = 501$, in which features such as age, city rank, career, device price, and consumer purchase are selected in a block-wise manner. Moreover, the advantages of the structure-aware SFGD method will become increasingly evident with increasing sample size. The performance differential seems to grow larger when comparing SFGD with sparsity-ignorant OFGD and structure-ignorant MLE.

6 Discussion

This study focused on the assortment personalization

problem using a data-driven conditional multinomial logit choice model, in which the sparse and low-rank settings of the parameter matrix are considered. Then, we present the SFGD method for our penalized maximum likelihood problem (i.e., a negative likelihood loss function plus certain penalties), leading to computational efficiency. Moreover, we prove that the SFGD exhibits the local convergence property, and the simulations show that SFGD achieves good estimation accuracy and feature selection ability with massive and high-dimensional data. A real-world application of advertising behavior data is presented, in which we demonstrate the excellent performance of our assortment personalization procedure.

One interesting direction for future research is the non-asymptotic analysis of the multinomial logit penalized likelihood in a high-dimensional setting, which statistically describes the estimation accuracy. Another research direction is to extend the SFGD method to a co-sparse framework that considers both the row-wise and column-wise sparsity of the parameter matrix.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (72071187, 11671374, 71731010, 71921001) and the Fundamental Research Funds for the Central Universities (WK3470000017, WK2040000027).

Conflict of interest

The authors declare that they have no conflict of interest.

Biographies

Jingyu Shao is currently a graduate student under the tutelage of Professor Zemin Zheng at the University of Science and Technology of China. His research interests focus on statistical learning and variable selection.

Ruipeng Dong currently is a postdoctoral researcher at the Chinese University of Hong Kong, Shenzhen. He received his PhD degree in Statistics from the University of Science and Technology of China. His research interests focus on statistical learning and variable selection.

References

- [1] Kallus N, Udell M. Dynamic assortment personalization in high dimensions. *Operations Research*, **2020**, 68 (4): 1020–1037.
- [2] Bernstein F, Kök A G, Xie L. Dynamic assortment customization with limited inventories. *Manufacturing & Service Operations Management*, **2015**, 17 (4): 538–553.
- [3] Golrezaei N, Nazerzadeh H, Rusmevichientong P. Real-time optimization of personalized assortments. *Management Science*, **2014**, 60 (6): 1532–1551.
- [4] Chen X, Owen Z, Pixton C, et al. A statistical learning approach to personalization in revenue management. *Management Science*, **2021**, 68 (3): 1923–1937.
- [5] Luo X, Andrews M, Fang Z, et al. Mobile targeting. *Management Science*, **2014**, 60 (7): 1738–1756.
- [6] Xue Z, Wang Z, Ettl M. Pricing personalized bundles: A new approach and an empirical study. *Manufacturing & Service Operations Management*, **2016**, 18 (1): 51–68.
- [7] Tibshirani R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **1996**, 58 (1): 267–288.
- [8] Yuan M, Lin Y. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **2006**, 68 (1): 49–67.
- [9] Meier L, Van De Geer S, Bühlmann P. The group lasso for logistic regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **2008**, 70 (1): 53–71.
- [10] Negahban S, Wainwright M J. Estimation of (near) low-rank matrices with noise and high-dimensional scaling. *The Annals of Statistics*, **2011**, 39 (2): 1069–1097.
- [11] Chen K, Chan K S, Stenseth N C. Reduced rank stochastic regression with a sparse singular value decomposition. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **2012**, 74 (2): 203–221.
- [12] Chen L, Huang J Z. Sparse reduced-rank regression for simultaneous dimension reduction and variable selection. *Journal of the American Statistical Association*, **2012**, 107 (500): 1533–1545.
- [13] Chen K, Dong H, Chan K S. Reduced rank regression via adaptive nuclear norm penalization. *Biometrika*, **2013**, 100 (4): 901–920.
- [14] Zhou K, Zha H, Song L. Learning social infectivity in sparse low-rank networks using multi-dimensional hawkes processes. In: Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics. PMLR, 2013: 641–649.
- [15] Wang Y X, Xu H, Leng C. Provable subspace clustering: When LRR meets SSC. In: Proceedings of the 26th International Conference on Neural Information Processing Systems: Volume 1. Red Hook, NY: Curran Associates Inc, 2013: 64–72.
- [16] Feng J, Lin Z, Xu H, et al. Robust subspace segmentation with block-diagonal prior. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2014: 3818–3825.
- [17] Chen J, Liu J, Ye J. Learning incoherent sparse and low-rank patterns from multiple tasks. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, **2012**, 5 (4): 1–31.
- [18] Agarwal A, Negahban S, Wainwright M J. Noisy matrix decomposition via convex relaxation: Optimal rates in high dimensions. *The Annals of Statistics*, **2012**, 40 (2): 1171–1197.
- [19] Chen J, Zhou J, Ye J. Integrating low-rank and group-sparse structures for robust multi-task learning. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: Association for Computing Machinery, 2011: 42–50.
- [20] Mishra A, Dey D K, Chen K. Sequential co-sparse factor regression. *Journal of Computational and Graphical Statistics*, **2017**, 26 (4): 814–825.
- [21] Mishra A, Dey D K, Chen Y, et al. Generalized co-sparse factor regression. *Computational Statistics & Data Analysis*, **2021**, 157: 107127.
- [22] Boyd S, Parikh N, Chu E. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, **2011**, 3 (1): 1–122.
- [23] Zheng Z, Bahadori M T, Liu Y, et al. Scalable interpretable multi-response regression via SEED. *Journal of Machine Learning Research*, **2019**, 20 (107): 1–34.
- [24] Jain P, Netrapalli P, Sanghavi S. Low-rank matrix completion using alternating minimization. In: Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing. New York: Association for Computing Machinery, 2013: 665–674.
- [25] Bhojanapalli S, Kyrillidis A, Sanghavi S. Dropping convexity for faster semi-definite optimization. In: 29th Annual Conference on Learning Theory. PMLR, 2016, 49: 530–582.
- [26] Golrezaei N, Nazerzadeh H, Rusmevichientong P. Real-time optimization of personalized assortments. *Management Science*, **2014**, 60 (6): 1532–1551.
- [27] Train K E. Discrete Choice Methods with Simulation. Cambridge, UK: Cambridge University Press, 2009.
- [28] Song Z, Woodruff D P, Zhong P. Low rank approximation with entrywise l_1 -norm error. In: Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing. New York: Association for Computing Machinery, 2017: 688–701.
- [29] Klopp O, Lounici K, Tsybakov A B. Robust matrix completion.

- Probability Theory and Related Fields*, **2017**, 169 (1): 523–564.
- [30] Gorski J, Pfeuffer F, Klamroth K. Biconvex sets and optimization with biconvex functions: A survey and extensions. *Mathematical Methods of Operations Research*, **2007**, 66 (3): 373–407.
- [31] Fan Y, Tang C Y. Tuning parameter selection in high dimensional penalized likelihood. *Journal of the Royal Statistical Society: Series B: (Statistical Methodology)*, **2013**, 75 (3): 531–552.
- [32] Wang H, Li B, Leng C. Shrinkage tuning parameter selection with a diverging number of parameters. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **2009**, 71 (3): 671–683.
- [33] Chen X, Krishnamurthy A, Wang Y. Robust dynamic assortment optimization in the presence of outlier customers. <https://arxiv.org/abs/1910.04183>.
- [34] Boerman S C, Kruikemeier S, Zuiderveen Borgesius F J. Online behavioral advertising: A literature review and research agenda. *Journal of Advertising*, **2017**, 46 (3): 363–376.
- [35] Mirsky L. A trace inequality of John von Neumann. *Monatshefte für Mathematik*, **1975**, 79 (4): 303–306.

Appendix A Details on sparse factored gradient descent

First, we let \mathbf{e}_l be the l th unit vector with the l th element 1 and other zeros, and $\mathbf{e}_l \in \mathbb{R}^T, \mathbf{e}_j \in \mathbb{R}^q$. Then we rewrite the loss

$$\mathcal{L}(\mathbf{X}; \boldsymbol{\theta}) = \frac{1}{T} \sum_{t=1}^T \left(\log \left(1 + \sum_{j \in S_t} \mathbf{e}_t^T \mathbf{X} \boldsymbol{\theta} \mathbf{e}_j \right) - \mathbf{e}_t^T \mathbf{X} \boldsymbol{\theta} \mathbf{e}_{j_t} \right) \quad (\text{A1})$$

and this leads to the gradient and Hessian with respect to $\boldsymbol{\theta}$ as

$$\begin{aligned} \nabla \mathcal{L}(\mathbf{X}; \boldsymbol{\theta}) &= \frac{1}{T} \sum_{t=1}^T \mathbf{X}^T \left(\frac{\sum_{j \in S_t} \mathbf{e}_t^T \mathbf{X} \boldsymbol{\theta} \mathbf{e}_j \mathbf{e}_j^T}{1 + \sum_{j \in S_t} \mathbf{e}_t^T \mathbf{X} \boldsymbol{\theta} \mathbf{e}_j} - \mathbf{e}_t \mathbf{e}_{j_t}^T \right) = \\ &= \frac{1}{T} \sum_{t=1}^T \left(\frac{\sum_{j \in S_t} \mathbf{e}_t^T \boldsymbol{\theta} \mathbf{e}_j \mathbf{x}_j \mathbf{e}_j^T}{1 + \sum_{j \in S_t} \mathbf{e}_t^T \boldsymbol{\theta} \mathbf{e}_j} - \mathbf{x}_t \mathbf{e}_{j_t}^T \right), \\ \nabla^2 \mathcal{L}(\mathbf{X}; \boldsymbol{\theta}) &= \frac{1}{T} \sum_{t=1}^T \left(\frac{\sum_{j \in S_t} \mathbf{e}_t^T \boldsymbol{\theta} \mathbf{e}_j (\mathbf{x}_j \mathbf{e}_j^T)^{\otimes 2}}{1 + \sum_{j \in S_t} \mathbf{e}_t^T \boldsymbol{\theta} \mathbf{e}_j} - \frac{(\sum_{j \in S_t} \mathbf{e}_t^T \boldsymbol{\theta} \mathbf{e}_j \mathbf{x}_j \mathbf{e}_j^T)^{\otimes 2}}{(1 + \sum_{j \in S_t} \mathbf{e}_t^T \boldsymbol{\theta} \mathbf{e}_j)^2} \right) \end{aligned} \quad (\text{A2})$$

where $\mathbf{Z}^{\otimes 2} = \mathbf{Z} \otimes \mathbf{Z}$ is the symmetric linear operator on matrices, and $\nabla \mathcal{L}(\mathbf{X}; \boldsymbol{\theta}), \nabla^2 \mathcal{L}(\mathbf{X}; \boldsymbol{\theta})$ are of sizes $p \times q$ and $pq \times pq$. Because we have the chain rule as

$$\begin{aligned} \nabla_U \mathcal{L}(\mathbf{X}; \mathbf{U} \mathbf{V}^T) &= \nabla_{\mathbf{U} \mathbf{V}^T} \mathcal{L}(\mathbf{X}; \mathbf{U} \mathbf{V}^T) \mathbf{V}, \\ \nabla_V \mathcal{L}(\mathbf{X}; \mathbf{U} \mathbf{V}^T) &= \nabla_{\mathbf{U} \mathbf{V}^T} \mathcal{L}(\mathbf{X}; \mathbf{U} \mathbf{V}^T)^T \mathbf{U}, \end{aligned}$$

then use the result above, we obtain

$$\begin{aligned} \nabla_U \mathcal{L}(\mathbf{X}; \mathbf{U} \mathbf{V}^T) &= \frac{1}{T} \sum_{t=1}^T \mathbf{X}^T \left(\frac{\sum_{j \in S_t} \mathbf{e}_t^T \mathbf{X} \mathbf{U} \mathbf{V}^T \mathbf{e}_j \mathbf{e}_j^T}{1 + \sum_{j \in S_t} \mathbf{e}_t^T \mathbf{X} \mathbf{U} \mathbf{V}^T \mathbf{e}_j} - \mathbf{e}_t \mathbf{e}_{j_t}^T \right) \mathbf{V} = \\ &= \frac{1}{T} \sum_{t=1}^T \left(\frac{\sum_{j \in S_t} \mathbf{e}_t^T \mathbf{U} \mathbf{v}_j \mathbf{x}_j \mathbf{v}_j^T}{1 + \sum_{j \in S_t} \mathbf{e}_t^T \mathbf{U} \mathbf{v}_j} - \mathbf{x}_t \mathbf{v}_{j_t}^T \right) \end{aligned} \quad (\text{A3})$$

$$\begin{aligned} \nabla_V \mathcal{L}(\mathbf{X}; \mathbf{U} \mathbf{V}^T) &= \frac{1}{T} \sum_{t=1}^T \left(\frac{\sum_{j \in S_t} \mathbf{e}_t^T \mathbf{X} \mathbf{U} \mathbf{V}^T \mathbf{e}_j \mathbf{e}_j^T}{1 + \sum_{j \in S_t} \mathbf{e}_t^T \mathbf{X} \mathbf{U} \mathbf{V}^T \mathbf{e}_j} - \mathbf{e}_t \mathbf{e}_{j_t}^T \right) \mathbf{X} \mathbf{U} = \\ &= \frac{1}{T} \sum_{t=1}^T \left(\frac{\sum_{j \in S_t} \mathbf{e}_t^T \mathbf{U} \mathbf{v}_j \mathbf{e}_j \mathbf{x}_t^T}{1 + \sum_{j \in S_t} \mathbf{e}_t^T \mathbf{U} \mathbf{v}_j} - \mathbf{e}_t \mathbf{x}_t^T \right) \mathbf{U} \end{aligned} \quad (\text{A4})$$

which clarifies the gradient descent direction in Algorithm 3.1.

Appendix B Proof of Theorem 3.1

Proof. For simplicity, when \mathbf{X} is fixed, we use $\mathcal{Q}(\boldsymbol{\theta})$ and $\mathcal{L}(\boldsymbol{\theta})$ instead of $\mathcal{Q}(\mathbf{X}; \boldsymbol{\theta})$ and $\mathcal{L}(\mathbf{X}; \boldsymbol{\theta})$, respectively. According to the overlap of \mathbf{U} and \mathbf{V} , one of \mathbf{U} and \mathbf{V} updates with the other fixed. We now divide the problem into \mathbf{U} -step and \mathbf{V} -step, and start from the \mathbf{V} -step.

The \mathbf{V} -step. We let $\boldsymbol{\theta}^V = \mathbf{U} \mathbf{V}^T$. In the \mathbf{V} step, by the chain rule, the updating of \mathbf{V} satisfies

$$\mathbf{V}' = \mathbf{V} - \eta \nabla_V \mathcal{Q}(\boldsymbol{\theta}) = \mathbf{V} - \eta [\nabla \mathcal{L}(\boldsymbol{\theta})]^T \mathbf{U}.$$

Then $\boldsymbol{\theta}^V = \mathbf{U} \mathbf{V}^T = \mathbf{U} [\mathbf{V} - \eta \mathbf{U} \nabla_V \mathcal{L}(\boldsymbol{\theta})]^T = \boldsymbol{\theta} - \eta \mathbf{U} \mathbf{U}^T \nabla \mathcal{L}(\boldsymbol{\theta})$. From the smoothness of $\mathcal{L}(\boldsymbol{\theta})$, we have

$$\begin{aligned} \mathcal{L}(\boldsymbol{\theta}^V) &\leq \mathcal{L}(\boldsymbol{\theta}) + \langle \nabla \mathcal{L}(\boldsymbol{\theta}), \boldsymbol{\theta}^V - \boldsymbol{\theta} \rangle + \frac{M_1}{2} \|\boldsymbol{\theta}^V - \boldsymbol{\theta}\|_F^2 = \\ &= \mathcal{L}(\boldsymbol{\theta}) + \text{Tr}(\nabla \mathcal{L}(\boldsymbol{\theta})(\boldsymbol{\theta}^V - \boldsymbol{\theta})^T) + \\ &+ \frac{M_1}{2} \text{Tr}((\boldsymbol{\theta}^V - \boldsymbol{\theta})(\boldsymbol{\theta}^V - \boldsymbol{\theta})^T), \end{aligned}$$

where $M_1 > 0$ denotes a constant. By the property of a trace, it entails the following:

$$\begin{aligned} \text{Tr}(\nabla \mathcal{L}(\boldsymbol{\theta})(\boldsymbol{\theta}^V - \boldsymbol{\theta})^T) &= -\eta \text{Tr}(\nabla \mathcal{L}(\boldsymbol{\theta})[\nabla \mathcal{L}(\boldsymbol{\theta})]^T \mathbf{U} \mathbf{U}^T) = \\ &= -\eta \text{Tr}([\nabla \mathcal{L}(\boldsymbol{\theta})]^T \mathbf{U} \mathbf{U}^T \nabla \mathcal{L}(\boldsymbol{\theta})) = \\ &= -\eta \|\nabla \mathcal{L}(\boldsymbol{\theta})\|_F^2. \end{aligned}$$

Furthermore, by the Von Neumann's trace inequality^[35], we have

$$\text{Tr}((\boldsymbol{\theta}^V - \boldsymbol{\theta})(\boldsymbol{\theta}^V - \boldsymbol{\theta})^T) = \eta^2 \text{Tr}([\nabla \mathcal{L}(\boldsymbol{\theta})]^T \mathbf{U} \mathbf{U}^T \mathbf{U} \mathbf{U}^T \nabla \mathcal{L}(\boldsymbol{\theta})) \leq \eta^2 \|\nabla \mathcal{L}(\boldsymbol{\theta})\|_F^2 \cdot \sigma_1(\mathbf{U}^T \mathbf{U}).$$

If the step size satisfies $\eta \leq \frac{1}{3M_1(\sigma_1(\mathbf{U}^T \mathbf{U}) + \sigma_1(\mathbf{V}^T \mathbf{V}))} < \frac{1}{3M_1\sigma_1(\mathbf{U}^T \mathbf{U})}$, then we derive

$$\begin{aligned} \mathcal{L}(\boldsymbol{\theta}^V) &\leq \mathcal{L}(\boldsymbol{\theta}) - \eta \|\nabla \mathcal{L}(\boldsymbol{\theta})\|_F^2 + \frac{M_1 \eta^2}{2} \|\nabla \mathcal{L}(\boldsymbol{\theta})\|_F^2 \sigma_1(\mathbf{U}^T \mathbf{U}) \leq \\ &= \mathcal{L}(\boldsymbol{\theta}) - \eta \|\nabla \mathcal{L}(\boldsymbol{\theta})\|_F^2 + \frac{\eta}{6} \|\nabla \mathcal{L}(\boldsymbol{\theta})\|_F^2 \leq \\ &= \mathcal{L}(\boldsymbol{\theta}) - \frac{5}{6} \eta \|\nabla \mathcal{L}(\boldsymbol{\theta})\|_F^2 \end{aligned} \quad (\text{B1})$$

The \mathbf{U} -step. Let $\boldsymbol{\theta}^U = \mathbf{U}' \mathbf{V}^T$. According to the updating rule, we have

$$\mathbf{U}' = \mathbf{U} - \eta \nabla_U \mathcal{Q}(\boldsymbol{\theta}) = \mathbf{U} - \eta \nabla \mathcal{Q}(\boldsymbol{\theta}) \mathbf{V} \mathbf{V}^T.$$

Here we don't need the explicit expression of $\nabla \mathcal{Q}(\boldsymbol{\theta})$, and from the smoothness of function $\mathcal{Q}(\boldsymbol{\theta})$, there is a constant $M_2 > 0$; if $\eta \leq \frac{1}{3M_2(\sigma_1(\mathbf{U}^T \mathbf{U}) + \sigma_1(\mathbf{V}^T \mathbf{V}))} < \frac{1}{3M_2\sigma_1(\mathbf{V}^T \mathbf{V})}$, then we derive

$$\begin{aligned}
 Q(\boldsymbol{\theta}^U) &\leq Q(\boldsymbol{\theta}) + \langle \nabla Q(\boldsymbol{\theta}), \boldsymbol{\theta}^U - \boldsymbol{\theta} \rangle + \frac{M_2}{2} \|\boldsymbol{\theta}^U - \boldsymbol{\theta}\|_F^2 = \\
 &Q(\boldsymbol{\theta}) - \eta \text{Tr}(\nabla Q(\boldsymbol{\theta}) \mathbf{V} \mathbf{V}^T [\nabla Q(\boldsymbol{\theta})]^T) + \\
 &\frac{M_2 \eta^2}{2} \text{Tr}(\mathbf{V} \mathbf{V}^T [\nabla Q(\boldsymbol{\theta})]^T \nabla Q(\boldsymbol{\theta}) \mathbf{V} \mathbf{V}^T) \leq \\
 &Q(\boldsymbol{\theta}) - \eta \|\nabla Q(\boldsymbol{\theta}) \mathbf{V}\|_F^2 + \frac{M_2 \eta^2}{2} \|\nabla Q(\boldsymbol{\theta}) \mathbf{V}\|_F^2 \sigma_1(\mathbf{V}^T \mathbf{V}) \leq \\
 &Q(\boldsymbol{\theta}) - \frac{5}{6} \eta \|\nabla Q(\boldsymbol{\theta}) \mathbf{V}\|_F^2
 \end{aligned} \tag{B2}$$

where the second inequality is Von Neumann's.

Let $M = \max\{M_1, M_2\}$; then, when $\eta \leq \frac{1}{3M(\sigma_1(\mathbf{U}^T \mathbf{U}) + \sigma_1(\mathbf{V}^T \mathbf{V}))}$ is satisfied, (B1) and (B2) hold simultaneously. In $\boldsymbol{\theta}' = \mathbf{U}' \mathbf{V}'^T$, from the results above, we can treat \mathbf{V}' as the fixed part first; by (B2), it yields

$$Q(\boldsymbol{\theta}') \leq Q(\boldsymbol{\theta}^V) - \frac{5}{6} \eta \|\nabla Q(\boldsymbol{\theta}^V) \mathbf{V}\|_F^2.$$

Next, \mathbf{U} is fixed and it enters the \mathbf{V} -step. In addition, $\nabla Q(\boldsymbol{\theta}) = \nabla \mathcal{L}(\boldsymbol{\theta})$ holds in this step; thus, by (B1), we have $Q(\boldsymbol{\theta}^V) \leq Q(\boldsymbol{\theta}) - \frac{5}{6} \eta \|[\nabla \mathcal{L}(\boldsymbol{\theta})]^T \mathbf{U}\|_F^2$, then

$$\begin{aligned}
 Q(\boldsymbol{\theta}') &\leq Q(\boldsymbol{\theta}) - \frac{5}{6} \eta \|[\nabla \mathcal{L}(\boldsymbol{\theta})]^T \mathbf{U}\|_F^2 - \frac{5}{6} \eta \|\nabla Q(\boldsymbol{\theta}^V) \mathbf{V}\|_F^2 \leq \\
 &Q(\boldsymbol{\theta}) - \frac{5}{6} \eta \|[\nabla \mathcal{L}(\boldsymbol{\theta})]^T \mathbf{U}\|_F^2.
 \end{aligned}$$

Because $\mathcal{L}(\boldsymbol{\theta})$ is constructed by negative likelihood, it entails

$$Q(\boldsymbol{\theta}) \geq \mathcal{L}(\boldsymbol{\theta}) = -\frac{1}{T} \sum_{t=1}^T \log(\mathbb{P}_{\boldsymbol{\theta}}(J = j_t | I = \mathbf{x}_t; S_t)) \geq -1,$$

where $\mathbb{P}_{\boldsymbol{\theta}}(J = j_t | I = \mathbf{x}_t; S_t)$ denotes the conditional probability defined in Eq. (4). $Q(\boldsymbol{\theta})$ has a lower bound, and the value of $Q(\boldsymbol{\theta})$ descends iteratively through SFGD, which shows that there is a local optimum $\tilde{\boldsymbol{\theta}}$ such that $Q(\boldsymbol{\theta})$ converges to $Q(\tilde{\boldsymbol{\theta}})$ through simple analysis.