https://pubsonline.informs.org/journal/ijoc

INFORMS JOURNAL ON COMPUTING
Vol. 35, No. 5, September–October 2023, pp. 1044–1060
ISSN 1091-9856 (print), ISSN 1526-5528 (online)

Simultaneous Dimension Reduction and Variable Selection for Multinomial Logistic Regression

Canhong Wen,[a] Zhenduo Li,[a] Ruipeng Dong,[a,*] Yijin Ni,[b] Wenliang Pan[c,*]

[a] International Institute of Finance, School of Management, University of Science and Technology of China, Hefei, Anhui 230026, China; [b] Industrial and System Engineering, Georgia Institute of Technology, 30318 Atlanta, Georgia; [c] Key Laboratory of Systems and Control, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China

*Corresponding author

Contact: wench@ustc.edu.cn, https://orcid.org/0000-0003-0220-9986 (CW); dwayne@mail.ustc.edu.cn (ZL); drp@mail.ustc.edu.cn, https://orcid.org/0000-0002-5073-4470 (RD); estelle@mail.ustc.edu.cn (YN); panwliang@amss.ac.cn, https://orcid.org/0000-0002-9821-6461 (WP)

Received: May 12, 2022
Revised: November 13, 2022; December 7, 2022; February 15, 2023; March 6, 2023
Accepted: March 7, 2023
Published Online in Articles in Advance: May 9, 2023

https://doi.org/10.1287/ijoc.2022.0132

Abstract. Multinomial logistic regression is a useful model for predicting the probabilities of multiclass outcomes. Because of the complexity and high dimensionality of some data, it is challenging to fit a valid model with high accuracy and interpretability. We propose a novel sparse reduced-rank multinomial logistic regression model to jointly select variables and reduce the dimension via a nonconvex row constraint. We develop a block-wise iterative algorithm with a majorizing surrogate function to efficiently solve the optimization problem. From an algorithmic aspect, we show that the output estimator enjoys consistency in estimation and sparsity recovery even in a high-dimensional setting. The finite sample performance of the proposed method is investigated via simulation studies and two real image data sets. The results show that our proposal has competitive performance in both estimation accuracy and computation time.

History: Accepted by Andrea Lodi, Area Editor for Design & Analysis of Algorithms–Discrete.
Funding: This work was supported by the National Natural Science Foundation of China [Grants 71991474, 12171449, 11801540, and 12071494] and the Natural Science Foundation of Anhui Province [Grant BJ2040170017].
Supplemental Material: The online appendix is available at https://doi.org/10.1287/ijoc.2022.0132.

Keywords: high-dimensional data • multinomial logistic regression • reduced-rank regression • variable selection

1. Introduction

Multinomial logistic regression is a widely used model depicting the relationship between a multiclass response and a set of independent variables, discrete or continuous. This model has been used as a powerful tool for analyzing dichotomous data in many areas, such as disease diagnosis, sociological research, natural language processing, and educational improvement. Related publications presenting application scenarios of multinomial models include performing risk analysis (Bayaga 2010), studying motivational effects (Lee et al. 2002), detecting anomaly intrusions (Wang 2005), constructing random utility models (Liu et al. 2020), and identifying machinery conditions (Pandya et al. 2014). For biostatisticians, modeling gene expression data for disease classifications is naturally formulated as a logistic regression equation (Vincent and Hansen 2014). In addition, multinomial logistic regression is applied in the task of image segmentation (Li et al. 2010) and recognition in the field of image processing.

Despite its popularity, the classical multinomial logistic regression model has limits in processing large-scale data because the maximum likelihood estimate tends to deteriorate as the dimension of predictors increases (Tutz et al. 2015). With high-dimensional settings frequently occurring in modern statistics, searching for extensions of classical models is inevitable.

In this paper, we consider the problem of analyzing data with a multiclass response and high-dimensional predictor variables. To address it, we consider performing dimensional reduction and variable selection simultaneously in the multinomial logistic regression model. Although a rank constraint is added to reduce the dimension of the coefficient matrix, a limit on the number of nonzero rows is imposed on that matrix to reduce the effective number of predictors. In addition, we offer an efficient algorithm and show that it has desirable properties of convergence and solution accuracy in theory and empirically.

1.1. Main Contributions

Our contribution in this paper is threefold. First, we introduce a sparse reduced-rank multinomial logistic regression (SRRMLR) model with the group ℓ_0-norm to jointly achieve dimension reduction and variable selection. Rather than adding a penalty on the row vectors, we constrain the number of nonzero rows, which is more intuitive and efficient. The proposed SRRMLR is related to two works in particular: Bunea et al. (2012) and She (2017). Although the studies in Bunea et al. (2012) and She (2017) use the squared error as the objective function, we use the negative likelihood function, a more complicated function. This makes the theoretical analysis and the computational algorithm much harder.

Second, we develop a block-wise iterative algorithm to solve the SRRMLR model based on the primal-dual active set (PDAS; Wen et al. 2020) algorithm, and the majorization-minimization (MM; Lange 2016) algorithm. The MM algorithm is applied to simplify the optimization procedure and reduce the computational cost, and the PDAS algorithm is modified to overcome the computational problem caused by the nonconvex group ℓ_0-norm. The output estimator is shown to be consistent in estimation and variable selection under some mild conditions; that is, the dimension of predictors can grow at an exponential rate of the sample size. It is worth noting that, unlike most previous works, our theoretical results are totally derived from an algorithmic aspect. In addition, we also show empirically that the proposed algorithm can generate solutions almost identical to the optimal one.

Last, we demonstrate in numerical experiments that the proposed algorithm is competitive with many previous and popular methods, including the multinomial lasso (Simon et al. 2013), ordinary multinomial logistic regression model via neural network (Vincent and Hansen 2014), and reduced-rank vector generalized linear model (Yee and Hastie 2003). In particular, we observe that in all synthetic data, our approach has substantially better performance with more accurate predictive power and more interpretive models, that is, fewer parameters in the final model. In the two benchmark data sets from the field of image recognition, our proposal outperforms the earlier approaches in terms of correctly identifying the true rank and yielding the minimum prediction error.

1.2. Literature Review

To avoid the curse of dimensionality and increase the interpretability of the obtained model, an intuitive approach is to add some sparsity constraints. Because its introduction, the concept of "sparsity" has been extensively used in many settings, including feature selection, factor analysis, and prediction accuracy improvement. It occurs as a regularization term for linear modeling, convex regression, and logistic regression. With an interpretable sparse multinomial classification model, we can see the groupwise connection between response and explanatory variables. A sparsity assumption about the observed data or hidden model is a fundamental premise for analysts to construct applicable models, as in the tree structure constructed in Mistry et al. (2020), dimension reduction method proposed for optimal pricing in Wang (2009), technique of searching for a sparse solution of a quadratic system in Jiang et al. (2020), building on a sparse convex model by Bertsimas and Mundru (2020), and ambulance redeployment programming problem studied by Maxwell et al. (2010). A sparse structure within the observed data are not only attractive for computing but also reveals simple connections between predictors and responses, increasing the interpretability in applications. Also, the low-rank pattern is natural with our constraint on the coefficient matrix. For example, in a lung disease study, multivariate linear regression can be applied to predict various pulmonary function test results by using segmented lung airway measurements from computed tomography scanned images (Choi et al. 2015, Chen 2016). As the airway variables of the same type and in the same segment are generally highly correlated (Chen et al. 2016), imposing constraint on the rank of the coefficient matrix enables us to further reduce the dimensionality and overcome the collinearity problem. It also identifies the association between pulmonary functions and lung airway measurements by several uncorrelated latent pathways and then further promotes the interpretability of models.

As the historically first approach, the reduced-rank vector generalized linear model related the responses and the predictors via a few hidden variables and some linear combinations of predictors, thus reducing the dimension (Yee and Hastie 2003). As an extension of reduced-rank regression (Izenman 1975), it worked by restricting the coefficient matrix to a low-rank matrix, indicating a simple structure within the actual model. Nevertheless, the hidden variables obtained by Yee and Hastie's method (Yee and Hastie 2003) are still linear combinations of all predictors, which might make it challenging to interpret the analytic results with so many predictors. Van der and Hinton (2008) proposed a nonparametric approach to dimension reduction named t-distributed Stochastic Neighbor Embedding, which is useful in visualization. Cheng et al. (2020) took a step forward by developing a supervised version. Although nonparametric methods lead to satisfying prediction accuracy, they do not provide an interpretable model, which is fundamentally useful in real applications.

Variable selection, which selects only a few relevant variables for the model, is an alternative approach to produce parsimonious models. A penalty or constraint on the coefficient matrix is added to the original optimization problem to reduce the effective elements. A naive approach is to constrain the nonzero rows in the coefficient matrix; yet this optimization problem is reduced to mixed integer programming that is impractical in application because of its computing

complexity. To address the computational problem, Meier et al. (2008) extended the group lasso (Yuan and Lin 2006) (a group version of lasso (Tibshirani 1996) where an ℓ_1 relaxation is used to constrain the nonzero rows) to logistic regression models with binary response variables. For multinomial logistic regression, Cawley et al. (2006) introduced a Bayesian ℓ_1 regularization to induce sparsity in the coefficient matrix by considering the Laplacian prior. From the algorithmic aspect, Krishnapuram et al. (2005) developed a fast algorithm for the ℓ_1-type penalty to find a sparse estimator for multinomial logistic models. Later on, a block-wise descent algorithm was proposed by Simon et al. (2013) to fit the penalized multinomial logistic regression model based on a group lasso penalty.

Despite its favorable properties, the aforementioned methods suffer from the problem of biased estimation and overselection because of the use of the ℓ_1-norm penalty. For multivariate continuous response, Chen and Huang (2012) addressed the bias problem by proposing an adaptive weighting strategy and showed its advantage over the group lasso penalty. Won et al. (2020) considered a group feature selection problem in networked data via the ℓ_0-norm regularization and developed a convex relaxation reformulation to cope with the computational challenge. However, there is no related work on multiclass response. In addition, the adaptive weighting version is a kind of convex relaxation of the group ℓ_0-norm, which counts the nonzero rows of the coefficient matrix. The methods based on group ℓ_0-norm have been shown to achieve the optimal rate for prediction in reduced-rank regression (Bunea et al. 2012, She 2017). However, the group ℓ_0-norm involves exhaustively searching over all possible combinations, which makes this method hard to compute even for moderate-size data. Recently, in an attempt to select the best subsets for univariate response, Wen et al. (2020) discovered that using the ℓ_0-norm performs substantially better than the lasso and other relaxation penalties. Bertsimas et al. (2016) developed a discrete extension of the first-order methods in convex optimization to obtain near-optimal solutions to the classic feature selection problem in linear regression.

1.3. Organization

Section 2 expounds our proposed methodology and presents the corresponding algorithm. In Section 3, we analyze the theoretical performance of the proposed algorithm. We demonstrate the competitive numerical performance of our method using simulation studies in Section 4 and show the effectiveness of our method via an application to the MNIST data set in Section 5. We discuss this paper in Section 6. Technical proofs of the main theoretical results and HAM10000 data analysis are provided in the online appendix.

2. Method and Algorithm

In this section, we present a sparse reduced-rank multinomial logistic regression (SRRMLR) model to simultaneously reduce dimension and select features for data with a multiclass response. Starting from the multinomial logistic model, a mathematical formulation for the SRRMLR is derived, which shapes our optimization procedure. Then we develop an alternative iteration algorithm for the SRRMLR. Through this section and the rest of the paper, the key notation summarized in Table 1 is used.

2.1. SRRMLR

Suppose we have observed data $\mathcal{D} = \{(\boldsymbol{x}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{x}_n, \boldsymbol{y}_n)\}$ of n observations, each with p features $\boldsymbol{x}_i \in \mathbb{R}^p$, and a q-class label $\boldsymbol{y}_i \in \mathbb{R}^q$. For clarity, let x_{ij}, y_{ik} represent the jth feature and the kth encoding label of the ith collected sample. The label uses a "1-of-q" rule; that is, if the ith observation belongs to the kth class, then we have $y_{ik} = 1$, $y_{il} = 0$ for all $l \neq k$.

The multinomial logistic model is an effective yet simple model for classification with multiclass data, which links the allocation probability of classes with a linear transformation of features through a score function. In particular, let $\boldsymbol{P}_{ik} = Pr(y_{ik} = 1|\boldsymbol{x}_i)$ denote the probability of the ith observation belonging to the kth class given $\boldsymbol{x}_i$, $i = 1, \ldots, n, k = 1, \ldots, q$. Without loss of generality, assume that the first class is chosen as the pivot. Then the multinomial logistic model works by separately regressing the other $(q-1)$ classes against the pivot class as follows:

$$\log \frac{\boldsymbol{P}_{ik}}{\boldsymbol{P}_{i1}} = \log \frac{Pr(y_{ik} = 1|\boldsymbol{x}_i)}{Pr(y_{i1} = 1|\boldsymbol{x}_i)} = \boldsymbol{c}_k^\top \boldsymbol{x}_i, \quad k = 2, \ldots, q, \tag{1}$$

where $\boldsymbol{c}_k$ is the coefficient for the kth class. Using the fact that all q of the probabilities must sum to one, we have

$$\boldsymbol{P}_{i1} = \frac{1}{1 + \sum_{l=2}^{q} \exp(\boldsymbol{c}_l^\top \boldsymbol{x}_i)}, \quad \boldsymbol{P}_{ik} = \frac{\exp(\boldsymbol{c}_k^\top \boldsymbol{x}_i)}{1 + \sum_{l=2}^{q} \exp(\boldsymbol{c}_l^\top \boldsymbol{x}_i)}, k = 2, \ldots, q. \tag{2}$$

For any given new data $\boldsymbol{x}$, we can calculate the probability of $Pr(y = k|\boldsymbol{x})$ belonging to k-th class via (2) with $\boldsymbol{x}_i$ being replaced by $\boldsymbol{x}$, and then assign it to the class with the maximum probability. That is, we predict that the new data $\boldsymbol{x}$ is in the Kth class if $Pr(y = K|\boldsymbol{x}) = \max_k Pr(y = k|\boldsymbol{x})$.

Table 1. Notation

Notation	Description
n	Number of observations/samples
p	Number of features
q	Number of classes
$\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ip})$	Predictor of the i-th sample
$\boldsymbol{y}_i = (y_{i1}, \ldots, y_{iq})$	Class label of the i-th sample
$\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$	Data with n observations
$\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^\top$	Design matrix
$\boldsymbol{Y} = (y_1, \ldots, y_n)^\top$	Response matrix
$\boldsymbol{C} = (\boldsymbol{c}_2, \ldots, \boldsymbol{c}_q)$	Coefficient matrix of size $p \times (q-1)$
$Pr(\cdot\|\cdot)$	Conditional probability function
$\boldsymbol{P}^{\boldsymbol{C}} = (\boldsymbol{P}_{ik})_{n\times q}$	Conditional probability matrix with $\boldsymbol{P}_{ik}$ defined in (2)
$\tilde{\boldsymbol{Y}}$	Matrix after eliminating the first column of $\boldsymbol{Y}$
$\tilde{\boldsymbol{P}}^{\boldsymbol{C}}$	Matrix after eliminating the first column of $\boldsymbol{P}^{\boldsymbol{C}}$
$\boldsymbol{B}, \boldsymbol{V}$	Decomposition matrices of $\boldsymbol{C} = \boldsymbol{B}\boldsymbol{V}^\top$ such that $\boldsymbol{B} \in \mathbb{R}^{p\times r}$, $\boldsymbol{V} \in \mathbb{R}^{(q-1)\times r}$, $\boldsymbol{V}^\top\boldsymbol{V} = I_r$
$\boldsymbol{\Gamma}$	Dual variable of $\boldsymbol{B}$, defined as $\boldsymbol{\Gamma} = \boldsymbol{X}^\top((2(\tilde{\boldsymbol{Y}} - \tilde{\boldsymbol{P}}^{\boldsymbol{C}^m}) + \boldsymbol{X}\boldsymbol{C}^m)\boldsymbol{V} - \boldsymbol{X}\boldsymbol{B})/n$
Δ_i	Sacrifice for the j-th feature, defined as $\Delta_i = \|\boldsymbol{B}_{i\cdot} + \boldsymbol{\Gamma}_{i\cdot}\|^2/2$
$\boldsymbol{C}^{(k)}$	The iterative at the k-th step of Algorithm 1
$\boldsymbol{C}^m$	The iterative at the m-th step of Algorithm 2
$\boldsymbol{C}^{m,k}$	Estimator in the k-th inner loop of the m-th outer loop in Algorithm 2
$\hat{\boldsymbol{C}}$	Solution of problem (4)
$\boldsymbol{C}^*$	True coefficient matrix
$L(\boldsymbol{C}; \mathcal{D})$	Negative log-likelihood function
$S(\boldsymbol{C}; \mathcal{D}\|\boldsymbol{C}^m)$	Surrogate function at $\boldsymbol{C}^m$

Let $\boldsymbol{C} = (\boldsymbol{c}_2, \ldots, \boldsymbol{c}_q)$ denote the coefficient matrix, then the log-likelihood function of the multinomial logistic model can be written as $l(\boldsymbol{C}; \mathcal{D}) = \sum_{i=1}^n \sum_{k=1}^q y_{ik} \log \boldsymbol{P}_{ik}$. Because of the constraints $\sum_{k=1}^q \boldsymbol{P}_{ik} = 1$ and $\sum_{k=1}^q y_{ik} = 1$ for all $i = 1, \ldots, n$, we can deduce that

$$l(\boldsymbol{C}; \mathcal{D}) = \sum_{i=1}^n \left[\sum_{k=2}^q y_{ik} \boldsymbol{c}_k^\top \boldsymbol{x}_i - \log\left(1 + \sum_{k=2}^q \exp(\boldsymbol{c}_k^\top \boldsymbol{x}_i)\right)\right].$$

The coefficient matrix $\boldsymbol{C}$ can be estimated via the maximum likelihood method, that is, $\max_{\boldsymbol{C}} l(\boldsymbol{C}; \mathcal{D})$. Because a minimization problem is usually formed by convention, we negate the log-likelihood function and average it as follows:

$$L(\boldsymbol{C}; \mathcal{D}) = -\frac{1}{n} \log l(\boldsymbol{C}; \mathcal{D}) = -\frac{1}{n} \sum_{i=1}^n \left[\sum_{k=2}^q y_{ik} \boldsymbol{c}_k^\top \boldsymbol{x}_i - \log\left(1 + \sum_{k=2}^q \exp(\boldsymbol{c}_k^\top \boldsymbol{x}_i)\right)\right]. \tag{3}$$

In the high-dimensional setting where p is larger than n, optimizing over (3) suffers from the "curse of dimensionality," which leads to the problem of heavy computational burden, low interpretability, and overfitting. Motivated by the work of Yee and Hastie (2003), we propose a SRRMLR method to deal with such high-dimensional data. Given two prespecified integers r and s, the SRRMLR problem is defined as

$$\begin{aligned} &\min_{\boldsymbol{C}} L(\boldsymbol{C}; \mathcal{D}) \\ &\text{s.t. } \operatorname{rank}(\boldsymbol{C}) \le r,\ \|\boldsymbol{C}\|_{2,0} \le s, \end{aligned} \tag{4}$$

where $\operatorname{rank}(\boldsymbol{C})$ denotes the rank of $\boldsymbol{C}$, and $\|\boldsymbol{C}\|_{2,0}$ denotes its number of nonzero rows. The first constraint restricts the rank of $\boldsymbol{C}$, which leads to dimensional reduction of the coefficient matrix. The second constraint limits the number of nonzero rows in $\boldsymbol{C}$ to be no more than s, which performs variable selection on the features.

2.2. Majorize Alternating Iterative Algorithm

There are two barriers to solving the optimization problem in (4). First, the optimization in (4) involves expensive computation cost because the minimization of the negative log-likelihood function has no explicit solution. Here, we will introduce an MM (Lange 2016) algorithm to simplify the optimization procedure and reduce the computational cost. Second, the entanglement between the two constraints in (4), as well as the nonconvex and noncontinuous function

(i.e., $\|\cdot\|_{2,0}$), can make the computation infeasible even in data of moderate size. We will address this problem by a matrix decomposition and solve the corresponding subproblems in an alternative iteration way.

Before formulating the algorithm, we introduce the notations and definitions needed in the subsequent section. Let $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^\top$ denote the design matrix. Without loss of generality, we assume each column of $\boldsymbol{X}$ has $\sqrt{n}$ norm. Similarly, let $\boldsymbol{Y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n)^\top$ denote the response matrix, and $\tilde{\boldsymbol{Y}}$ denote the matrix after eliminating the first column of $\boldsymbol{Y}$. Denote the conditional probability matrix by $\boldsymbol{P}^{\boldsymbol{C}} = (\boldsymbol{P}_{ik})_{n\times q}$, where $\boldsymbol{P}_{ik}$ is defined in (2), and $\boldsymbol{C}$ is the coefficient matrix. Similarly, let $\tilde{\boldsymbol{P}}^{\boldsymbol{C}}$ denote the matrix formed by eliminating the first column of $\boldsymbol{P}^{\boldsymbol{C}}$.

Let $[n] = \{1, \ldots, n\}$. For any set $\mathcal{A} \subseteq [n]$, $\mathcal{A}^c = [n] \setminus \mathcal{A}$ denotes the complement of $\mathcal{A}$, and $|\mathcal{A}|$ denotes its cardinality. For matrix $\boldsymbol{M} = (\boldsymbol{M}_{ij}) \in \mathbb{R}^{p\times q}$, we use $\boldsymbol{M}_{i\cdot}$ and $\boldsymbol{M}_{\cdot j}$ to denote the ith row and jth column of $\boldsymbol{M}$, respectively. Then for any set $\mathcal{A} \subseteq [p]$ and $\mathcal{B} \subseteq [q]$, define $\boldsymbol{M}_{\mathcal{A}\cdot} = (\boldsymbol{M}_{i\cdot} : i \in \mathcal{A}) \in \mathbb{R}^{|\mathcal{A}|\times q}$ and $\boldsymbol{M}_{\cdot\mathcal{B}} = (\boldsymbol{M}_{\cdot j} : j \in \mathcal{B}) \in \mathbb{R}^{p\times|\mathcal{B}|}$. We define the Frobenius norm of $\boldsymbol{M}$ by $\|\boldsymbol{M}\|_F = \sqrt{\sum_{i=1}^{p}\sum_{j=1}^{q} M_{ij}^2}$, and the row support set of $\boldsymbol{M}$ by $\mathrm{supp}(\boldsymbol{M}) = \{i : \|\boldsymbol{M}_{i\cdot}\| \neq 0\}$.

2.2.1. MM Algorithm. The main ingredient of the algorithm is the construction of a proper surrogate function. Motivated by the work of Yee and Hastie (2003), we define a surrogate function of $L(\boldsymbol{C}; \mathcal{D})$ via a second-order Taylor expansion. In particular, at the mth step, $m = 0, 1, \ldots$, given the current solution $\boldsymbol{C}^m$, we define the surrogate function $S(\boldsymbol{C}; \mathcal{D}|\boldsymbol{C}^m)$ by

$$S(\boldsymbol{C}; \mathcal{D}|\boldsymbol{C}^m) = L(\boldsymbol{C}^m; \mathcal{D}) + \frac{1}{4n}\|\boldsymbol{X}(\boldsymbol{C} - \boldsymbol{C}^m) - 2(\tilde{\boldsymbol{Y}} - \tilde{\boldsymbol{P}}^{\boldsymbol{C}^m})\|_F^2 - \frac{1}{n}\|\tilde{\boldsymbol{Y}} - \tilde{\boldsymbol{P}}^{\boldsymbol{C}^m}\|_F^2.$$

The next proposition states that $S(\boldsymbol{C}; \mathcal{D}|\boldsymbol{C}^m)$ is indeed a surrogate function of $L(\boldsymbol{C}; \mathcal{D})$.

Proposition 1. *The function $S(\boldsymbol{C}; \boldsymbol{C}^m)$ majorizes the objective function $L(\boldsymbol{C}; \mathcal{D})$ at $\boldsymbol{C}^m$. That is,*

$$S(\boldsymbol{C}; \mathcal{D}|\boldsymbol{C}^m) \geq L(\boldsymbol{C}; \mathcal{D}) \text{ for all } \boldsymbol{C},$$
$$S(\boldsymbol{C}^m; \mathcal{D}|\boldsymbol{C}^m) = L(\boldsymbol{C}^m; \mathcal{D}).$$

The detailed proof is given in the online appendix.

In summary, at the mth step, rather than directly optimizing using the objective function $L(\boldsymbol{C}; \mathcal{D})$, we consider the following simplified problem:

$$\begin{aligned} \min_{\boldsymbol{C}} \quad & S(\boldsymbol{C}; \mathcal{D}|\boldsymbol{C}^m) \\ \text{s.t.} \quad & \mathrm{rank}(\boldsymbol{C}) \leq r,\ \|\boldsymbol{C}\|_{2,0} \leq s. \end{aligned} \tag{5}$$

Denote the solution of (5) by $\boldsymbol{C}^{m+1}$. Then we can iteratively derive a local optimum as m goes to infinity, which is guaranteed by

$$L(\boldsymbol{C}^{m+1}; \mathcal{D}) \leq S(\boldsymbol{C}^{m+1}; \mathcal{D}|\boldsymbol{C}^m) \leq S(\boldsymbol{C}^m; \mathcal{D}|\boldsymbol{C}^m) = L(\boldsymbol{C}^m; \mathcal{D}),$$

and the fact that $L(\boldsymbol{C}; \mathcal{D}) \geq 0$ for all $\boldsymbol{C}$.

2.2.2. Alternating Iterative Procedure. To overcome the computational difficulties caused by entanglement between the two constraints, a rank factorization is used to simplify Problem (5). For any $p \times (q-1)$ matrix $\boldsymbol{C} = (\boldsymbol{C}_{ij})$ satisfying $\mathrm{rank}(\boldsymbol{C}) \leq r$ and $\|\boldsymbol{C}\|_{2,0} \leq s$, we can decompose it via singular value decomposition. Specifically, $\boldsymbol{C}$ can be expressed in the form of $\boldsymbol{C} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$, where $\boldsymbol{U}$ is an $p \times r$ column orthogonal matrix, $\boldsymbol{\Sigma}$ is an $r \times r$ diagonal matrix with singular values on the diagonal, and $\boldsymbol{V}$ is a $(q-1) \times r$ column orthogonal matrix. Define the active set for the rows in $\boldsymbol{C}$ as $\mathcal{A}$, that is, $\mathcal{A} = \{i : \|\boldsymbol{C}_{i\cdot}\| \neq 0\}$. Let $\boldsymbol{P}_{\mathcal{A}}$ be a projection matrix on $\boldsymbol{C}$ that preserves the ith row with its index $i \in \mathcal{A}$ and resets it to zero when $i \notin \mathcal{A}$. Then we can decompose $\boldsymbol{C}$ as

$$\boldsymbol{C} = \boldsymbol{P}_{\mathcal{A}}\boldsymbol{C} = \boldsymbol{P}_{\mathcal{A}}(\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top) = (\boldsymbol{P}_{\mathcal{A}}\boldsymbol{U}\boldsymbol{\Sigma})\boldsymbol{V}^\top \triangleq \boldsymbol{B}\boldsymbol{V}^\top,$$

where $\boldsymbol{B} = \boldsymbol{P}_{\mathcal{A}}\boldsymbol{U}\boldsymbol{\Sigma}$ is a $p \times r$ matrix satisfying $\|\boldsymbol{B}\|_{2,0} \leq s$. We summarize the previous discussion in Proposition 2.

Proposition 2. *For any $p \times (q-1)$ matrix $\boldsymbol{C}$ satisfying the following constraints:*

$$\mathrm{rank}(\boldsymbol{C}) \leq r; \|\boldsymbol{C}\|_{2,0} \leq s, \tag{6}$$

there exist a $p \times r$ matrix $\boldsymbol{B}$ and a $(q-1) \times r$ matrix $\boldsymbol{V}$ satisfy

$$\|\boldsymbol{B}\|_{2,0} \leq s, \quad \boldsymbol{V}^\top\boldsymbol{V} = \boldsymbol{I}_r, \tag{7}$$

such that $\boldsymbol{C} = \boldsymbol{B}\boldsymbol{V}^\top$. Conversely, for any $p \times r$ matrix $\boldsymbol{B}$ and any $(q-1) \times r$ matrix $\boldsymbol{V}$ satisfying (7), $\boldsymbol{C} = \boldsymbol{B}\boldsymbol{V}^\top$ must satisfy (6).

The sufficiency of factorization is already proved by the previous discussion, whereas the necessity is an obvious fact from properties of $\boldsymbol{B}$ and $\boldsymbol{V}$. Proposition 2 states that the interplay between the rank and sparsity constraints can be untangled equivalently. Therefore, Problem (5) can be rewritten as

$$\begin{aligned} \min_{\boldsymbol{B},\boldsymbol{V}} \quad & S(\boldsymbol{B}\boldsymbol{V}^\top; \mathcal{D} | \boldsymbol{C}^m) \\ \text{s.t.} \quad & \boldsymbol{V}^\top \boldsymbol{V} = \boldsymbol{I}_r, \ \|\boldsymbol{B}\|_{2,0} \le s. \end{aligned} \tag{8}$$

The optimization is now with respect to $\boldsymbol{B}$ and $\boldsymbol{V}$, which can be done in a block-wise alternative iteration. Without loss of generality, we consider $\|\boldsymbol{B}\|_{2,0} = s$. To start with, denote the row-wise minimizer of (8) by $\boldsymbol{B}$ and $\boldsymbol{V}$. Given $\boldsymbol{V}$, the constraint $\|\boldsymbol{B}\|_{2,0} = s$ indicates that there are $(p - s)$ rows that would be forced to zero. To determine which rows are nonzero, we define the dual variable of $\boldsymbol{B}$ as

$$\boldsymbol{\Gamma} = \frac{1}{n} \boldsymbol{X}^\top ((2(\tilde{\boldsymbol{Y}} - \tilde{\boldsymbol{P}}^{\boldsymbol{C}^m}) + \boldsymbol{X}\boldsymbol{C}^m)\boldsymbol{V} - \boldsymbol{X}\boldsymbol{B}).$$

The objective function in (8) is equivalent to $\frac{1}{4n}\|2((\tilde{\boldsymbol{Y}} - \tilde{\boldsymbol{P}}^{\boldsymbol{C}^m}) + \boldsymbol{X}\boldsymbol{C}^m) - \boldsymbol{X}\boldsymbol{B}\boldsymbol{V}^\top\|_F^2$ by ignoring some constants. Then by the constraint $\boldsymbol{V}^\top\boldsymbol{V} = \boldsymbol{I}_r$, we know that the objective function in (8) can be transformed into the function $\frac{1}{4n}\|2((\tilde{\boldsymbol{Y}} - \tilde{\boldsymbol{P}}^{\boldsymbol{C}^m}) + \boldsymbol{X}\boldsymbol{C}^m)\boldsymbol{V} - \boldsymbol{X}\boldsymbol{B}\|_F^2$ equivalently. With this transformation, there is a projected response matrix in a low-rank space that is $2((\tilde{\boldsymbol{Y}} - \tilde{\boldsymbol{P}}^{\boldsymbol{C}^m}) + \boldsymbol{X}\boldsymbol{C}^m)\boldsymbol{V}$. Then $\boldsymbol{\Gamma}$ is the correlation between the predictors and the residual with respect to the projected responses. Together with a hard thresholding rule, we classify the predictors into the active set and inactive set, where the rows of primal variable $\boldsymbol{B}$ and dual variable $\boldsymbol{\Gamma}$ are complementary. On the one hand, the rows in primal variable measure the importance of the active predictors. On the other hand, the rows in dual variable measure the importance of the inactive predictor. In each iteration, we re-evaluate these two primal and dual variables to decide which predictors are more important and then update the active and inactive sets. Moreover, with the given $\boldsymbol{V}$, the original problem is reduced into a common multiresponse regression. Therefore, following the argument of Wen et al. (2020), by fixing all rows except the ith row, we know that minimizing the objective function in (8) with the fixed $\boldsymbol{V}$ yields to an optimum point $\boldsymbol{B}_{i\cdot} + \boldsymbol{\Gamma}_{i\cdot}$. Based on this, we can define a sacrifice Δ_i by replacing $\boldsymbol{B}_{i\cdot} + \boldsymbol{\Gamma}_{i\cdot}$ by $\boldsymbol{0}$, which is given by

$$\begin{aligned} \Delta_i &= S((\boldsymbol{B}_{1\cdot}, \dots, \boldsymbol{0}, \dots, \boldsymbol{B}_{p\cdot})^\top \boldsymbol{V}^\top; \mathcal{D} | \boldsymbol{C}^m) - S((\boldsymbol{B}_{1\cdot}, \dots, \boldsymbol{B}_{i\cdot} + \boldsymbol{\Gamma}_{i\cdot}, \dots, \boldsymbol{B}_{p\cdot})^\top \boldsymbol{V}^\top; \mathcal{D} | \boldsymbol{C}^m) \\ &= \frac{1}{2}\|\boldsymbol{B}_{i\cdot} + \boldsymbol{\Gamma}_{i\cdot}\|^2. \end{aligned}$$

Therefore, we may force those rows to zero if they contribute the least total sacrifices to the overall loss. To realize this, let $\Delta_{(1)} \ge \cdots \ge \Delta_{(p)}$ denote the decreasing rearrangement of $\{\Delta_i\}_{i=1}^p$, then truncate the ordered sacrifice vector at position s. Combining the analytical result here with what came before, we have

$$\boldsymbol{B}_{i\cdot} = \begin{cases} \boldsymbol{B}_{i\cdot} + \boldsymbol{\Gamma}_{i\cdot}, & \Delta_i \ge \Delta_{(s)}, \\ \boldsymbol{0} & \text{otherwise.} \end{cases} \tag{9}$$

Define $\mathcal{A} = \text{supp}(\boldsymbol{B})$ and $\mathcal{I} = (\mathcal{A})^c$. From (9), it follows that

$$\mathcal{A} = \{i : \Delta_i \ge \Delta_{(s)}\}, \quad \mathcal{I} = \{i : \Delta_i < \Delta_{(s)}\},$$

and

$$\begin{cases} \boldsymbol{B}_{\mathcal{A}\cdot} = (\boldsymbol{X}_{\cdot\mathcal{A}}^\top \boldsymbol{X}_{\cdot\mathcal{A}})^{-1} \boldsymbol{X}_{\mathcal{A}}^\top (2(\tilde{\boldsymbol{Y}} - \tilde{\boldsymbol{P}}^{\boldsymbol{C}^m}) + \boldsymbol{X}\boldsymbol{C}^m)\boldsymbol{V}, \quad \boldsymbol{B}_{\mathcal{I}\cdot} = \boldsymbol{0}, \\ \boldsymbol{\Gamma}_{\mathcal{I}\cdot} = \boldsymbol{X}_{\mathcal{I}}^\top ((2(\tilde{\boldsymbol{Y}} - \tilde{\boldsymbol{P}}^{\boldsymbol{C}^m}) + \boldsymbol{X}\boldsymbol{C}^m)\boldsymbol{V} - \boldsymbol{X}\boldsymbol{B})/n, \qquad \boldsymbol{\Gamma}_{\mathcal{A}\cdot} = \boldsymbol{0}. \end{cases} \tag{10}$$

By substituting $\boldsymbol{B}$ of (10) into (8), we can derive an explicit expression for $\boldsymbol{V}$; that is, the matrix consists of eigenvectors corresponding to the top r eigenvalues of the following matrix:

$$\left(2(\tilde{\boldsymbol{Y}} - \tilde{\boldsymbol{P}}^{\boldsymbol{C}^m}) + \boldsymbol{X}\boldsymbol{C}^m\right)^\top \boldsymbol{X}_{\cdot\mathcal{A}} (\boldsymbol{X}_{\cdot\mathcal{A}}^\top \boldsymbol{X}_{\cdot\mathcal{A}})^{-1} \boldsymbol{X}_{\cdot\mathcal{A}}^\top \left(2(\tilde{\boldsymbol{Y}} - \tilde{\boldsymbol{P}}^{\boldsymbol{C}^m}) + \boldsymbol{X}\boldsymbol{C}^m\right).$$

We summarize the previous discussion in Algorithm 1.

Algorithm 1 (Alternative Iterative Algorithm for Surrogate Problem (8))

Input: Data matrices $\boldsymbol{X}$ and $\tilde{\boldsymbol{Y}}$, sparsity s, rank r, the initialized value $\boldsymbol{C}^{(0)}$.

1: $\boldsymbol{M} = 2(\tilde{\boldsymbol{Y}} - \tilde{\boldsymbol{P}}^{\boldsymbol{C}^{(0)}}) + \boldsymbol{X}\boldsymbol{C}^{(0)}$.
2: Initialization: $k = 0$, $\mathcal{A}^{(0)} = \{i : \|\boldsymbol{C}_{i\cdot}^{(0)}\| \neq 0\}$, and $\mathcal{I}^{(0)} = (\mathcal{A}^{(0)})^c$.
3: **while** $\boldsymbol{C}^{(k)}$ has not converged **do**
4: Calculate $\boldsymbol{V}^{(k+1)}$ by the eigenvectors corresponding to the top r eigenvalues of matrix $\boldsymbol{M}^\top \boldsymbol{X}_{\cdot\mathcal{A}^{(k)}} (\boldsymbol{X}_{\cdot\mathcal{A}^{(k)}}^\top \boldsymbol{X}_{\cdot\mathcal{A}^{(k)}})^{-1} \boldsymbol{X}_{\cdot\mathcal{A}^{(k)}}^\top \boldsymbol{M}$.
5: Update the primal and dual variables by

$$\begin{cases} \boldsymbol{B}_{\mathcal{A}^{(k)}\cdot}^{(k+1)} = (\boldsymbol{X}_{\cdot\mathcal{A}^{(k)}}^\top \boldsymbol{X}_{\cdot\mathcal{A}^{(k)}})^{-1} \boldsymbol{X}_{\cdot\mathcal{A}^{(k)}}^\top \boldsymbol{M}\boldsymbol{V}^{(k+1)}, & \boldsymbol{B}_{\mathcal{I}^{(k)}\cdot}^{(k+1)} = \boldsymbol{0} \\ \boldsymbol{\Gamma}_{\mathcal{I}^{(k)}\cdot}^{(k+1)} = \boldsymbol{X}_{\cdot\mathcal{I}^{(k)}}^\top (\boldsymbol{M}\boldsymbol{V}^{(k+1)} - \boldsymbol{X}\boldsymbol{B}^{(k+1)})/n, & \boldsymbol{\Gamma}_{\mathcal{A}^{(k)}\cdot}^{(k+1)} = \boldsymbol{0}. \end{cases}$$

6: Compute the sacrifice $\Delta_i^{(k+1)} = \|\boldsymbol{B}_{i\cdot}^{(k+1)} + \boldsymbol{\Gamma}_{i\cdot}^{(k+1)}\|_2$, $i = 1, \ldots, p$.
7: Determine $\mathcal{A}^{(k+1)} = \{i : \Delta_i^{k+1} \geq \Delta_{(s)}^{(k+1)}\}$, $\mathcal{I}^{(k+1)} = (\mathcal{A}^{(k+1)})^c$.
8: Let $\boldsymbol{C}^{(k+1)} = \boldsymbol{B}^{(k+1)}(\boldsymbol{V}^{(k+1)})^\top$.
9: Set $k = k + 1$.
10: **end while**

Output: $\boldsymbol{C}^{(k)}$.

Remark 1. Because our target is to optimize over the coefficient matrix $\boldsymbol{C}$, we stop Algorithm 1 if $\boldsymbol{C}^{(k)}$ has converged, that is, $\|\boldsymbol{C}^{(k+1)} - \boldsymbol{C}^{(k)}\|_F < \tau$ for a prespecified tolerance τ. In practice, we set $\tau = 0.01$ or 0.001.

2.2.3. Majorize Alternating Iterative Algorithm. By combining the ideas in Sections 2.2.1 and 2.2.2, we develop an efficient algorithm we name the majorize alternating iterative (MAI) algorithm for the SRRMLR problem. Specifically, at the mth step of the outer loop, we majorize the objective function in (4) by $S(\boldsymbol{C} : \mathcal{D}|\boldsymbol{C}^m)$, and thus simplify Problem (4) to Problem (5). Then, to solve Problem (5), we use an inner loop to derive an optimum by an alternating iterative procedure. That is, at the kth step of the inner loop, we update $\boldsymbol{B}^{(k)}$ and $\boldsymbol{V}^{(k)}$ iteratively as shown in Algorithm 1.

We summarize the previous steps in Algorithm 2.

Algorithm 2 (MAI Algorithm for the SRRMLR)

Input: Data matrices $\boldsymbol{X}$ and $\tilde{\boldsymbol{Y}}$, sparsity s, rank r, the initialized value $\boldsymbol{C}^0$.
Initialization: $m = 0$.
while $\boldsymbol{C}^m$ has not converged **do**
Run **Algorithm 1 with r, s, and $\boldsymbol{C}^m$. Denote the output by $\boldsymbol{C}^{m+1}$.**
Set $m = m + 1$.
end while
Output: $\boldsymbol{C}^m$

Remark 2. For the initialization of $\boldsymbol{C}$, we simply set $\boldsymbol{C}^0 = \boldsymbol{0}_{p\times(q-1)}$. We stop the iteration according to whether $\|\boldsymbol{C}^{m+1} - \boldsymbol{C}^m\|_F$ is small enough or a maximum allowed iteration number is reached.

Remark 3. A grid search is used for tuning the rank r and sparsity s. We can simply increase r by one each step from 1 to $r_{\max}$ and increase s similarly from 1 to $r_{\max}$. In Section 3, we will derive upper bounds on r and s such that the estimation is consistent with the sample size n. For each pair of r and s, we compute the SRRMLR estimator via Algorithm 2 on the training data, and a criterion (such as the negative log-likelihood value) is calculated on the test data. We then choose the optimal parameters with the smallest criterion value. When the data are not so rich, the cross-validation technique (Hastie et al. 2009) can be applied.

3. Theoretical Properties

3.1. Preliminaries

For the theoretical justification, we need to define some more notations before the main result. The proposed method includes the inner loop, that is, Algorithm 1, and the outer loop, that is, Algorithm 2. Denote the outer iteration index and the inner iteration index by m and k, respectively. Based on this, we further denote $\boldsymbol{C}^{m,k}$ as the estimate at the kth iteration of the inner loop and the mth iteration of the outer loop in Algorithm 2. For clarity, let $\hat{\boldsymbol{C}}$ be the minimizer of (4), and $\boldsymbol{C}^*$ be the true coefficient matrix in model 1.

Next, we introduce some regular conditions required in the theoretical analysis.

Condition 3.1. *With the given sparsity level s, there are two constants $0 < c_-(s) \le c_+(s) < \infty$ such that*

$$c_-(s) \le \inf_{\boldsymbol{u}\neq 0} \frac{\|\boldsymbol{X}_{\cdot\mathcal{A}}\boldsymbol{u}\|_2^2}{n\|\boldsymbol{u}\|_2^2} \le \sup_{\boldsymbol{u}\neq 0} \frac{\|\boldsymbol{X}_{\cdot\mathcal{A}}\boldsymbol{u}\|_2^2}{n\|\boldsymbol{u}\|_2^2} \le c_+(s),$$

where $\mathcal{A}$ is any subset of $[p]$ satisfying $|\mathcal{A}| \le s$.

Condition 3.2. *With the given sparsity s, we assume that there is a constant θ_s such that*

$$\sup_{u\neq 0} \frac{\|\boldsymbol{X}_{\cdot\mathcal{A}}^\top \boldsymbol{X}_{\cdot\mathcal{B}}\boldsymbol{u}\|_2}{n\|\boldsymbol{u}\|_2} \le \theta_s,$$

with $\mathcal{A}, \mathcal{B} \subset [p]$, $|\mathcal{A}| \le s$, $|\mathcal{B}| \le s$, and $\mathcal{A} \cap \mathcal{B} = \emptyset$.

Condition 3.3. *Let $\boldsymbol{p}^C(\boldsymbol{x}) = (\boldsymbol{p}_1^C(\boldsymbol{x}), \ldots, \boldsymbol{p}_q^C(\boldsymbol{x}))^\top$ denote the population version of conditional probability in (2). Suppose that there is some constant $0 < c_C < 1/4$ such that*

$$c_C \le \min_{1\le i\le q} \boldsymbol{p}_i^C(\boldsymbol{x})(1 - \boldsymbol{p}_i^C(\boldsymbol{x})),$$

for all $\boldsymbol{x}$.

We now discuss the relevance of these technical conditions in detail.

Remark 4. From Weyl's theorem and Condition 3.1, we have

$$c_-(s) \le \lambda_s\left(\frac{\boldsymbol{X}_{\cdot\mathcal{A}}^\top \boldsymbol{X}_{\cdot\mathcal{A}}}{n}\right) \le \lambda_1\left(\frac{\boldsymbol{X}_{\cdot\mathcal{A}}^\top \boldsymbol{X}_{\cdot\mathcal{A}}}{n}\right) \le c_+(s).$$

Therefore, Condition 3.1 is the restricted eigenvalue condition bounding the s-sparse eigenvalues of $\boldsymbol{X}^\top\boldsymbol{X}/n$. Condition 3.2 indicates any sparse off-diagonal block of $\boldsymbol{X}^\top\boldsymbol{X}/n$ is upper-bounded so that any two distinct small subsets of variables of $\boldsymbol{X}/\sqrt{n}$ are mutually uncorrelated. Condition 3.2 is a more detailed restriction than Condition 3.1. In fact, it can be shown that $\theta_s \le c_+(s)$ by fundamental algebra.

In the proof of Theorem 1, these two conditions are used to give an upper bound on the terms like $\|\hat{\boldsymbol{C}}_{\hat{A}\cdot} - \boldsymbol{C}^*_{\hat{A}\cdot}\|_F$. The solution $\hat{\boldsymbol{C}}_{\hat{A}\cdot}$ has an explicit form similar to that of the least squares solution. In particular, it involves the term $(\boldsymbol{X}_{\cdot\hat{A}}^\top \boldsymbol{X}_{\cdot\hat{A}})^{-1}$, which is bounded by $1/c_-(s)I_{|\hat{A}|}$ because of Condition 3.1. Considering the difference between $\hat{\boldsymbol{C}}_{\hat{A}\cdot}$ and $\boldsymbol{C}^*_{\hat{A}\cdot}$ yields the terms like $\boldsymbol{X}_{\cdot\hat{A}}^\top \boldsymbol{X}_{\cdot\mathcal{A}^*-\hat{A}}$. Condition 3.2 is applied here to bound it by θ_s from the upper side. These two conditions together guarantee the identification of the solution in (4), and they are commonly used in high-dimensional regression studies (Bickel et al. 2009, Fan and Lv 2014, Huang et al. 2018, Li et al. 2019).

Remark 5. Condition 3.3 imposes a restriction on the true conditional probability, and a similar condition can be found in Meier et al. (2008) in deriving the consistency results in logistic regression. This can be viewed as the balanced outcome assumption, used to show that the surrogate function is a good approximation of the negative log-likelihood function. In particular, we show that the MM iteration converges geometrically to the global optimum of (4) when initialized in its neighborhood. This is derived by showing that each MM iteration is a contraction mapping with constant $1 - 2\eta(1-\eta)$, where η is the element in the conditional probability matrix that is closest to zero or one. Condition 3.3 ensures that this iterative constant is less than one, and thus the MM algorithm must converge.

3.2. Main Results

We focus on the following parameter space indexed by the sparsity level s,

$$\Lambda(s) = \{\boldsymbol{C} : \|\boldsymbol{C}_{\mathcal{A}^c\cdot} - \boldsymbol{C}^*_{\mathcal{A}^c\cdot}\|_F \le \bar{c}\|\boldsymbol{C}_{\mathcal{A}\cdot} - \boldsymbol{C}^*_{\mathcal{A}\cdot}\|_F, \mathcal{A} = \{i : \|\boldsymbol{C}_{i\cdot}\| \neq 0\} \text{ and } |\mathcal{A}| \le s\}$$

for the constant $\bar{c}$. This is a natural extension of the cone set in Wang et al. (2020) to the multivariate response case. Both the sparse solution and true coefficients are in the set $\Lambda(s)$. As a baseline for the comparison with our algorithm, we establish the statistically nonasymptotic property in Theorem 1 for the global optimum of (4).

Theorem 1 (Error Bound of Model Solution). *Suppose that Conditions 3.1 and 3.2 hold for $s \ge s^*$, and Condition 3.3 holds for both $\boldsymbol{C}^*$ and $\hat{\boldsymbol{C}} \in \Lambda(s)$. If $\tilde{c} = \min\{c_{\boldsymbol{C}^*}, c_{\hat{\boldsymbol{C}}}\} \ge \bar{c}\theta_s/(4c_-(s))$, then for $r \ge r^*$ and, $s \ge s^*$, with probability at least $1-\delta$ for any $\delta \in (0,1)$, we have*

$$\|\hat{\boldsymbol{C}} - \boldsymbol{C}^*\|_F \le c\sqrt{\frac{rqs}{n}\log\frac{pq}{\delta}},$$

where the constant c is an absolute positive constant.

Theorem 1 points out that the statistical convergence rate of the global minimizer of (4) can achieve the rate $O(\{n^{-1}rqs\log(pq)\}^{1/2})$, which is as good as the result of the linear reduced-rank regression (Zheng et al. 2019). The result is trivial because the multiresponse logistic regression can be seen as a generalized reduced-rank regression. However, this result is only theoretical guidance because the optimization problem is nonconvex, and the previous algorithms usually achieve a local minimizer and not the global one. The statistical properties of the algorithmic output remain a mystery. Unlike previous literature, we can establish the theoretical guarantee for the algorithmic output, which is presented in the following theorem and corollary.

With the user-specified rank r and sparsity level s, the parameters defined in Conditions 3.1 and 3.2 determine a parameter as follows:

$$\gamma = \frac{\theta_s(1+\sqrt{r})(1+\theta_s)}{c_-^2(s)} + \frac{(1+\sqrt{r})\theta_s}{c_-(s)}.$$

Next, we present the error bound of the estimation at each iteration in Algorithm 2.

Theorem 2 (Error Bound of Estimator at Each Iteration). *Suppose $\boldsymbol{C}^{m,k}$ is the solution of Algorithm 2 for the given $r \geq r^*$, $s \geq s^*$ and the initial estimation $\boldsymbol{C}^0$. Given Conditions 3.1 and 3.2 and $0 < \gamma < 1$, and with probability at least $1-(pq)^{-\alpha}$, we have*

$$\|\boldsymbol{C}^{m,k} - \boldsymbol{C}^*\|_F \leq c_1\sqrt{r}\gamma^k\|\boldsymbol{C}^*\|_{op} + c_2(1-2\zeta(1-\zeta))^m r\|\boldsymbol{C}^0 - \boldsymbol{C}^*\|_F + c_3 r\sqrt{\frac{qs\log(pq)}{n}},$$

where α, c_1, c_2, and c_3 are some positive constants, and the parameter ζ is defined as

$$0 < \zeta \triangleq \inf_{\boldsymbol{C}\in\mathcal{B}(s,r,h_0)} \min_{i,j} \tilde{\boldsymbol{P}}^{\boldsymbol{C}}_{i,j} < 1$$

with $h_0 = 2c_+(s)/c_-(s)(\|\boldsymbol{C}^0 - \boldsymbol{C}^*\|_F + 20\alpha pq)$ and

$$\mathcal{B}(s,r,h_0) = \{\boldsymbol{C} : rank(\boldsymbol{C}) \leq r, \|\boldsymbol{C}\|_{2,0} \leq s, \|\boldsymbol{C} - \boldsymbol{C}^*\|_F \leq h_0\}.$$

The indices m and k are for the outer and inner loops, respectively. Note that $0 < \gamma < 1$ and $1/2 < 1 - 2\zeta(1-\zeta) < 1$, which ensures the convergence of the proposed algorithm. On the one hand, it implies that the algorithm output will geometrically converge. On the other hand, the nonvanishing rate $r\{qs\log(pq)/n\}^{1/2}$ corresponds to the statistical theoretical guarantee. Therefore, with large enough m and k, the output $\boldsymbol{C}^{m,k}$ bridges the gap between the numerical computation and statistical theory.

As for the assumption $\gamma < 1$, γ will be less than one as long as the correlation parameter θ_s is small enough. A special case is that the columns of $\boldsymbol{X}$ are mutually orthogonal, in which case, $\theta_s = 0$. In this case, there actually is a closed formula for the solution in the inner loop (linear approximation), which can be seen in Zheng et al. (2014). A parameter similar to γ can be found in Huang et al. (2018) and Zhu et al. (2020). The difference lies in the factor $\sqrt{r}$ because we consider the multiple responses in this paper. Thus, there is an inflation factor $\sqrt{r}$. Under the univariate response, the parameter γ coincides with Huang et al. (2018), Zhu et al. (2020) because the rank can be seen as one. As for the detailed discussion of the assumption $\gamma < 1$, the reader is referred to Huang et al. (2018); a similar argument can be made on the γ defined in this paper.

Theorem 2 shows that the initial bias ($\|\boldsymbol{C}^0 - \boldsymbol{C}^*\|_F$ and $\|\boldsymbol{C}^*\|_{\text{op}}$) can vanish with increasing iteration numbers k and m. After a large enough number of iterations, we can establish the statistical convergence rate for the numerical output, which we express as Corollary 1.

Corollary 1 (Overall Error Bound and Support Recovery). *Under the same conditions of Theorem 2, if the iteration numbers m and k satisfy $k \geq \log_\gamma c_4\sqrt{r}/(2c_2\|\boldsymbol{C}^*\|_F)\sqrt{qs\log(pq)/n}$, and $m \geq \log_{\zeta'} c_4/(c_3\|\boldsymbol{C}^0 - \boldsymbol{C}^*\|_F)\sqrt{qs\log(pq)/n}$ with $\zeta' = 1-2\zeta(1-\zeta)$, then with the probability at least $1-(pq)^{-\alpha}$, we have*

$$\|\boldsymbol{C}^{m,k} - \boldsymbol{C}^*\|_F \leq 2c_4 r\sqrt{\frac{qs\log(pq)}{n}}.$$

Furthermore, under the assumption $\min_{i\in supp(\boldsymbol{C}^*)}\|\boldsymbol{C}^*_{i\cdot}\|_2 \geq 2c_4 r\sqrt{qs\log(pq)/n}$, the following result holds:

$$Pr(supp(\boldsymbol{C}^{m,k}) = supp(\boldsymbol{C}^*)) \geq 1-(pq)^{-\alpha}.$$

The corollary reveals that the statistical convergence rate of $\boldsymbol{C}^{k,m}$ can achieve $O(r\{qs\log(pq)/n\}^{1/2})$, where it has a $\sqrt{r}$-factor inflation because of the quadratic approximation in the algorithm. This convergence rate is analogous (to a factor of $\sqrt{r}$) to the previous results, such as Zheng et al. (2019) and Uematsu et al. (2019). However, the major difference is that we directly give a statistical guarantee for the numerical solution, which bridges the gap between the computation and the theoretical analysis. The current literature usually uses the coordinate descending algorithm to solve the estimation, leading to their numerical solutions sometimes being only local optimums because of the nonconvexity of the low-rank and sparse constraints. Recently, Chen et al. (2022) established the algorithmic convergence of sparse reduced-rank regression that is based on the stagewise algorithm, but they still do not present a statistical guarantee for their numerical output. Unlike their work, we design a new algorithm and then directly analyze the numerical solution in theory, which enables us to get a statistical theoretical guarantee for the algorithm.

Moreover, under the minimum signal condition, Corollary 1 shows that we can select the true variables with a significant probability. Therefore, the proposed algorithm in this paper can identify the important variables and then enhance the interpretability of the model.

4. Numerical Experiments

In this section, we evaluate the finite-sample performance of the proposed method and compare it with the state-of-art methods. All experiments were conducted on an Ubuntu 18.04.6 LTS Server machine with Intel Xeon Gold 6248 80-core processor @ 2.50 GHz and 125 GB of RAM. All our methods were implemented in R (version 4.1.2). The code used for our experiments are publicly available at https://github.com/C2S2-HF/SRRMLR.

4.1. Experiment Setup

For our method, we terminate the algorithm if the absolute difference between the two estimated coefficients is less than $\tau = 0.001$. We use a grid search for tuning rank and sparsity, that is, range the rank from $\{1,\dots,r_{\max}\}$ and sparsity level from $\{1,\dots,s_{\max}\}$. In addition, in the grid search of optimal values for r and s, we impose the restriction that $r \le s$ and do not run the algorithm when $r > s$. For the upper bounds $r_{\max}$ and $s_{\max}$, we set $s_{\max} = \lceil 10(n/(q\log(pq)))^{1/4}\rceil$ and $r_{\max} = \min(s_{\max}, q-1, \lceil 10(n/(q\log(pq)))^{1/3}\rceil)$ based on the theoretical results in Section 3.

As the baseline method, we include the ordinary multinomial logistic regression model via neural network (NNET), implemented in the *nnet* package (Ripley et al. 2016). We set the maximum number of weights to 6,000 and use the default settings without trace optimization in the *nnet* package. To see the effectiveness of the feature selection, the multinomial lasso (mLasso (Simon et al. 2013), that is, multinomial logistic regression with ℓ_1-norm regularization) is considered for our performance comparison. The mLasso method is one of the most commonly used variable selection methods in multinomial logistic regression models, and it has been implemented in the *glmnet* package. For mLasso, we set all the arguments to the default values; for instance, the number of regularization parameters λ is 100. To evaluate the rank selection performance of the proposal, we compare it with the reduced-rank vector generalized linear model (RRVGLM; Yee and Hastie 2003), implemented in *VGAM* package. To provide a fair comparison, we apply the same maximal rank as our proposal, that is, $r_{\max}$, and search for an optimal rank by ranging from one to $r_{\max}$. The other arguments are set to defaults.

For all the simulation studies, the true coefficient matrix $\boldsymbol{C}^* \in \mathbb{R}^{p\times q}$ is constructed as $\boldsymbol{C}^* = \boldsymbol{B}\boldsymbol{V}^\top$, where $\boldsymbol{B} \in \mathbb{R}^{p\times r}$ and $\boldsymbol{V} \in \mathbb{R}^{q\times r}$ are two matrices defined later. For the matrix $\boldsymbol{B}$, the elements of its top five rows are drawn from the uniform distribution $U(-2/q, -1/q) \cup U(1/q, 2/q)$, and the remaining rows are set to zero. Each element of the matrix $\boldsymbol{V}$ is drawn from the uniform distribution $U(0, q)$ independently. We fix the class size to $q = 8$ and the true rank to $r = 5$ throughout this section.

We consider training data of size $n = 400$, denoted by $\mathcal{D} = \{(\boldsymbol{x}_1, \boldsymbol{y}_1), \dots, (\boldsymbol{x}_n, \boldsymbol{y}_n)\}$. We generate each predictor $\boldsymbol{x}_i \in \mathbb{R}^p$, independently from a multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{0}, \Sigma)$ with covariance matrix $\Sigma = (\Sigma_{ij})_{p\times p}$. Then the class label $\boldsymbol{y}_i \in \mathbb{R}^q$ is generated from Model (2). Two types of covariance matrix are considered:

- Compound symmetries (CS): $\Sigma_{ij} = 0.5$, if $i \ne j$ and $\Sigma_{ii} = 1, i, j = 1, \dots, p$;
- Auto-regressive (AR): $\Sigma_{ij} = 0.5^{|i-j|}$, $i, j = 1, \dots, p$.

To explore the effect of dimension p on the performance, we use values 50, 200, and 500, respectively, representing scenarios of small, modest, and high dimension. Overall, there are six different scenarios of simulation settings, and a total of 100 replications are conducted for each setting.

For all methods, the optimal tuning parameter(s) is determined via a validation data set of size 1,000. Specifically, we determine the optimal parameter(s) as that with the smallest negative log-likelihood value on the validation data. Then, another independent test data set of size 1,000 is used to evaluate the predictive accuracy.

Seven measurements are considered to assess the empirical performance of each method. Denote the nonzero rows and zero rows of the true coefficient matrix $\boldsymbol{C}^*$ as $\mathcal{A}^*$ and $\mathcal{I}^*$, respectively. For given data $\mathcal{D}$, denote an estimator of $\boldsymbol{C}^*$ by $\hat{\boldsymbol{C}}$, and its corresponding nonzero rows as $\hat{\mathcal{A}}$. Let $\hat{\mathcal{I}} = \hat{\mathcal{A}}^c$ denote the zero rows. First, we measure the estimation accuracy (Est) in terms of the mean squared error and predictive accuracy (Pred) with the negative log-likelihood, that is,

$$\text{Est} = \frac{1}{pq}\|\boldsymbol{C} - \hat{\boldsymbol{C}}\|_F^2, \quad \text{Pred} = L(\hat{\boldsymbol{C}}; \mathcal{D}_{test}), \tag{11}$$

where $\mathcal{D}_{test}$ denotes an independent test data set. The next two measurements are to evaluate the performance of variable selection, the sensitivity (Sen) and specificity (Spe), which are defined by

$$\text{Sen} = \frac{|\mathcal{A}^* \cap \hat{\mathcal{A}}|}{|\mathcal{A}^*|}, \quad \text{Spe} = \frac{|\mathcal{I}^* \cap \hat{\mathcal{I}}|}{|\mathcal{I}^*|}.$$

We also report the estimated rank $\hat{r}$ and the number of nonzero rows in $\hat{\boldsymbol{C}}$, that is, $|\hat{\mathcal{A}}|$. Finally, we record the running time (Time) in seconds for each method to evaluate the computational performance.

4.2. Simulation Results

Tables 2 and 3 summarize the average results for all measurements. We can see that our method significantly outperforms the other three methods in terms of variable selection and predictive performance, irrespective of the dimension p and covariance structure in X. The improvement is driven largely by accurate detection of true nonzero rows in the coefficient matrix. The performance of the mLasso method is less impressive than that of our method because the former has comparable sensitivity but much lower specificity. This is consistent with the discussion that lasso-related methods might cause overselection problems (Fan and Li 2001, Fan et al. 2004). In addition, because it does not consider the rank constraint, the mLasso gives larger values in prediction and estimation error. The other two methods cannot select variables and thus have poor performance with regard to variable selection. For the prediction and estimation performance, both NNET and RRVGLM perform very poorly as the dimension p increases. In comparison, our method continues to perform well and is stable as the dimension p increases.

In terms of rank recovery, both mLasso and NNET cannot recover the rank, which indicates they fail to reveal the internal structure of the coefficient matrix. Although RRVGLM gives the estimated rank $\hat{r}$, the gaps between the estimated rank and the true rank are large. Moreover, RRVGLM is sensitive and unstable, as can be proven with simulated data. Our proposed method can realize the rank recovery, and the estimated rank is very close to the true rank.

With respect to the computational time, we can see that the RRVGLM method requires time similar to other methods and cannot deal with the cases when $p > n$. The NNET method does not involve any tuning parameter, and thus it is the fastest in the low-dimensional setting. However, this computational advantage diminishes as p increases. This is largely because of the increasing number of parameters that must be estimated. The SRRMLR and mLasso methods have comparable similar computation time when p is relatively small. However, as p increases, SRRMLR performs slightly worse in terms of running time. This could be expected considering the use of a grid search for tuning parameter determination in SRRMLR; more computations are required for our proposed approach. Nevertheless, SRRMLR

Table 2. Simulation Results with the Compound Symmetry Covariance in X

p	Method	Pred	Estimate	$\hat{r}$	$\lvert\hat{\mathcal{A}}\rvert$	Sen	Spe	Time (s)
50	SRRMLR	1,256.030	0.030	4.240	5.140	1.000	0.997	5.784
	mLasso	1,598.091	0.299	—	33.360	0.996	0.369	10.038
	NNET[a]	6,034.134	4.749	—	50.000	1.000	0.000	0.205
	RRVGLM[a]	1,669.428	0.291	1.677	50.000	1.000	0.000	338.04
200	SRRMLR	1,280.851	0.009	4.270	5.370	0.980	0.998	19.686
	mLasso	1,655.199	0.079	—	60.810	0.990	0.714	7.899
	NNET[a]	69,698.433	72.285	—	200.000	1.000	0.000	3.006
	RRVGLM	—	—	—	—	—	—	—
500	SRRMLR	1,288.622	0.004	4.350	5.300	0.972	0.999	53.667
	mLasso	1,694.433	0.035	—	77.810	0.974	0.853	12.225
	NNET	36,862.754	6.938	—	500.000	1.000	0.000	29.715
	RRVGLM	—	—	—	—	—	—	—

Note. —, these results cannot be obtained.

[a]In the $p = 50$ setting, the number of effective results from NNET is 97 in 100 repetitions, and the number of effective results from RRVGLM is 95 in 100 repetitions. In the $p = 200$ setting, the number of effective results from NNET is 98 in 100 repetitions.

Table 3. Simulation Results with the Autoregression Covariance in X

p	Method	Pred	Estimate	$\hat{r}$	$\lvert\hat{\mathcal{A}}\rvert$	Sen	Spe	Time (s)
50	SRRMLR	1,219.186	0.038	4.200	5.020	0.998	0.999	6.364
	mLasso	1,625.903	0.335	—	33.190	1.000	0.374	9.808
	NNET[a]	7,355.270	8.893	—	50.000	1.000	0.000	0.199
	RRVGLM[a]	1,666.573	0.240	1.667	50.000	1.000	0.000	237.509
200	SRRMLR	1,240.753	0.010	4.250	5.100	0.996	0.999	23.368
	mLasso	1,681.168	0.087	—	64.720	0.992	0.694	7.214
	NNET[a]	61,368.110	50.830	—	200.000	1.000	0.000	2.858
	RRVGLM	—	—	—	—	—	—	—
500	SRRMLR	1,222.785	0.004	4.170	5.020	0.996	0.999	59.998
	mLasso	1,693.959	0.036	—	88.840	0.996	0.831	11.523
	NNET	40,812.972	5.979	—	500.000	1.000	0.000	27.325
	RRVGLM	—	—	—	—	—	—	—

Note. —, these results cannot be obtained.
[a]In the $p = 50$ setting, the number of effective results from NNET is 98 in 100 repetitions, and the number of effective results from RRVGLM is 97 in 100 repetitions. In the $p = 200$ setting, the number of effective results from NNET is 99 in 100 repetitions.

terminates in less than one minute even when $p = 500$, which suggests the scalability of our method and its applicability to high-dimensional data.

4.3. Algorithmic Analysis

In this section, we investigate the convergence properties of our algorithm. The SRRMLR algorithm is tested on the simulated example from Section 4.1 by setting $n = 400$, $q = 8$, $s = 10$, and $r = 5$. We run Algorithm 1 with input $s = 10$ and $r = 5$.

Figure 1 plots the mean squared error (MSE) of the estimator and the true value versus the iterations in the outer loop during 100 replications. Because of space limits, we only present the results of the CS covariance matrix; those of AR are omitted because they are similar. As shown in Figure 1, the estimation errors converge to zero within a hundred iterations for all tested numbers of dimensions. This implies that Algorithm 2 converges at a fast rate and can converge to the true value of the estimated parameters, which is consistent with Theorem 2.

Next, we evaluate how many iterations are needed in the inner loop. Figure 2 shows bar plots of inner iterations for each outer loop iteration, that is, the kth step in the inner loop of the m-th step in the outer loop. At the beginning of the outer loop, the average number of iterations fluctuates around 3 for CS and 2 for AR. These values reflect the impact of inaccurate estimation when the algorithm starts. As the number of outer iterations increases, the number of the iterations in the inner loop decreases, which is expected because the algorithm is approaching the true solution. When the iterations in the outer loop reach around 150, the average iterations in the inner loop converge to zero. This means that when the estimate becomes accurate, the active set tends to remain the same, removing the need to update the active set.

Finally, we investigate how the algorithm converges in terms of the active set as n increases. In particular, let the sample size n increase from 200 to 800, and other settings are as in the paragraph above. To visualize the convergence of selected rows, we plot the average number of intersections of the current active set $\hat{\mathcal{A}}^m$ and the true active set $\mathcal{A}^*$,

Figure 1. (Color online) Plots of MSE vs. Number of Iterations in the Outer Loop, Where Different Colors Represent Various Randomized Experiments Under Fixed, Well-Selected s and r

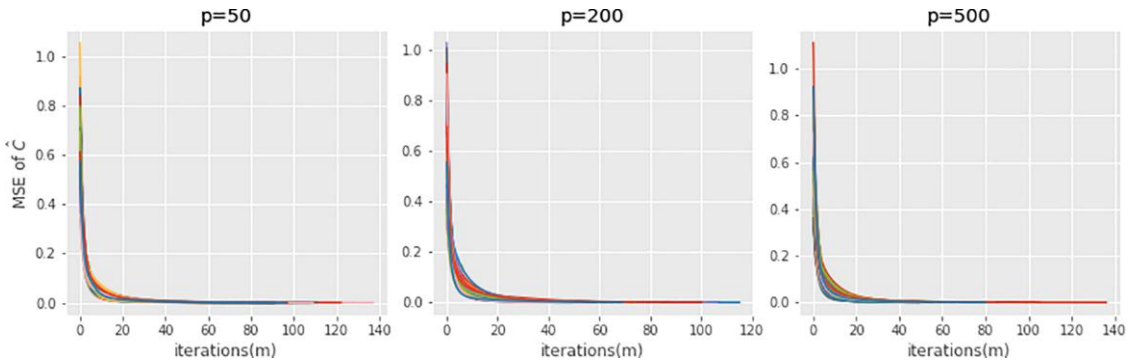
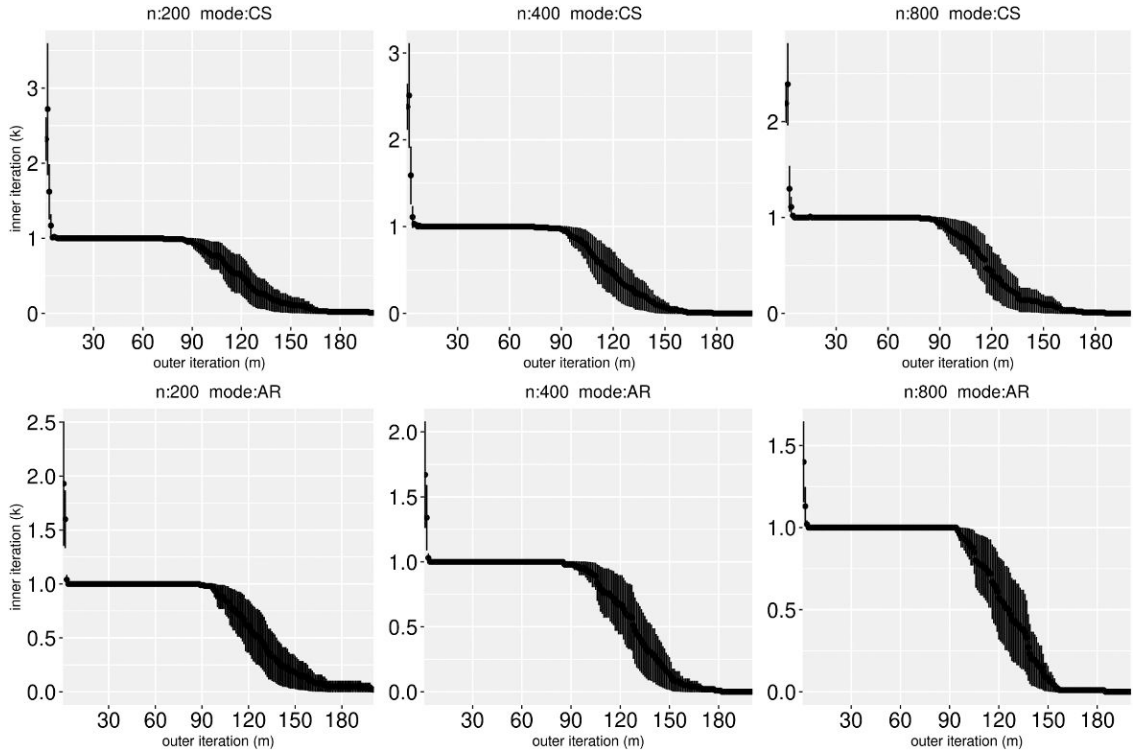


Figure 2. Bar Plots of Inner Iterations for Each Outer Iteration



Notes. The left, center, and right panels correspond to $n = 200$, 400, and 800, respectively. The top panel corresponds to the results of CS structure in X, and the bottom panel corresponds to the results of AR structure in X.

that is, $|\mathcal{A}^* \cap \hat{\mathcal{A}}^m|$, versus the iterations m during 100 replications in Figure 3. It can be seen that after approximately 50 iterations, the algorithm keeps selecting the same order of active set. Furthermore, as the sample size n increases, the estimated active set converges to the true active set.

Figure 4 plots the average number of intersections of the estimated active set $\hat{\mathcal{A}}$ and the true active set $\mathcal{A}^*$, that is, $|\mathcal{A}^* \cap \hat{\mathcal{A}}|$, versus the sample size n during 100 replications. As for the case with $p = 50$ and AR structure, the true active set of rows is correctly identified most times, even when the sample size is small, for example, $n = 200$. When p increases, it becomes difficult to correctly determine the true relevant rows with limited sample size, and the number of intersections fluctuates. Nevertheless, as the sample size n increases, all true relevant rows are selected by our new method, which is consistent with the theoretical results expressed in Corollary 1.

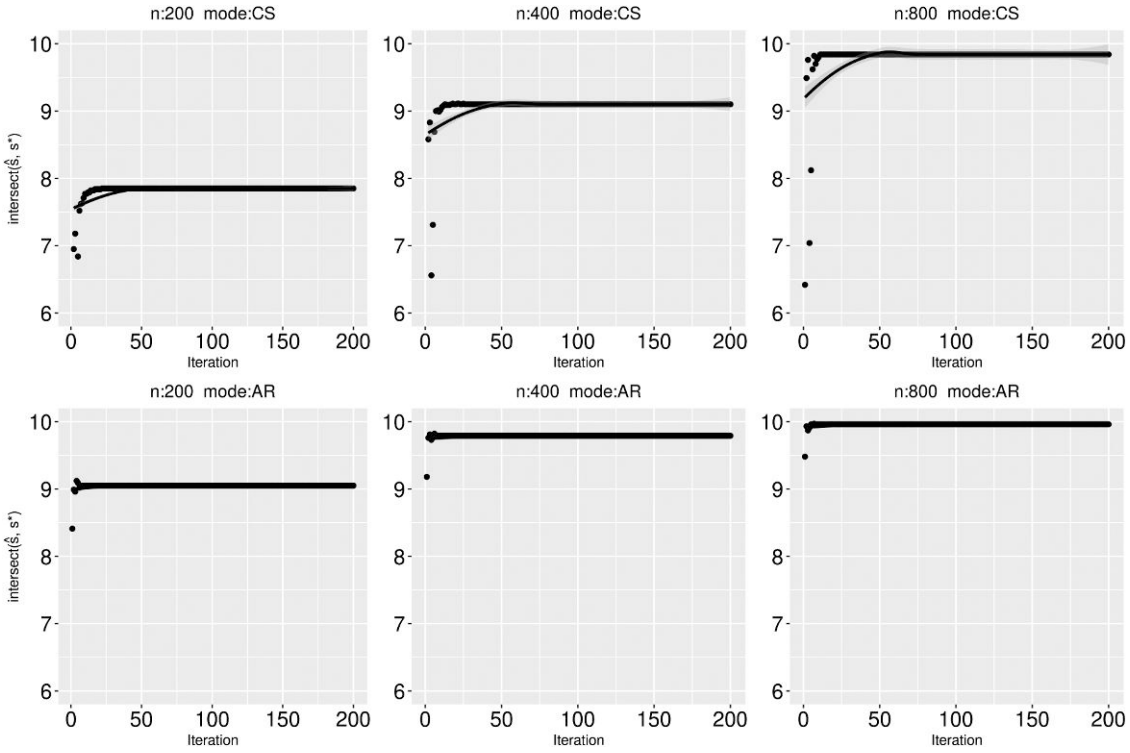
5. Real Data Analysis

In this section, we illustrate the practical application of our proposal by analyzing two real data sets. The MNIST data set (LeCun et al. 1998) is from the Modified National Institute of Standards and Technology database, a data set of handwritten Arabic numerals. It is popularly considered a standard data set for handwritten digit classification in optical character recognition and machine learning research. In addition, another analysis of a more complicated data set of the pigmented skin lesion data, HAM10000 (Połap et al. 2021), is given in the online appendix.

We applied all four methods in Section 4 to these two data sets and evaluated their performance via three measurements. To measure the estimation accuracy, we consider the negative log-likelihood value defined by Pred $= L(\hat{C}; \mathcal{D}_{test})$, where $\mathcal{D}$ is a test data set. We also record the estimated rank $\hat{r}$ and the estimated sparsity $|\hat{\mathcal{A}}|$ for each method.

All the parametric settings are similar to those in Section 4 except for the tuning parameter selection strategy. To determine an optimal pair of rank and sparsity, we consider the five-fold cross-validation technique. To be specific, we divide the whole data set into five almost equal-size groups. For each group, we take the group as a validation data set and the remaining groups as a training data set. Then a model is fitted on the training set, and the negative log-likelihood value is computed on the test set. We choose (an) optimum tuning parameter(s) with the smallest negative log-likelihood value. We use the previous strategy to tune the rank and sparsity in SRRMLR, λ in mLasso, and rank in RRVGLM.

Figure 3. Plots of the Average Number of Intersections of the Current Active Set $\hat{\mathcal{A}}^m$ and the True Active Set $\mathcal{A}^*$ vs. Iterations m over 100 Replications



Notes. The left, center, and right panels correspond to $n = 200$, 400, and 800, respectively. The top panel corresponds to the results of CS structure in $\boldsymbol{X}$, and the bottom panel corresponds to the results of AR structure in $\boldsymbol{X}$.

The MNIST data set is widely used in the field of machine learning and can be thought of as a "Hello, World" data set. It contains 60,000 training images and 10,000 test images. All these grayscale images are size-normalized to 28×28 pixels, and the center of gravity of the intensity lies at the center of the image. Because the image is 28×28 in dimension and RRVGLM does not work when the dimension is too high, we reduce the dimension to 100 via the principal component analysis technique, which can explain approximately 91.5% of the total variance. The results are presented in Table 4.

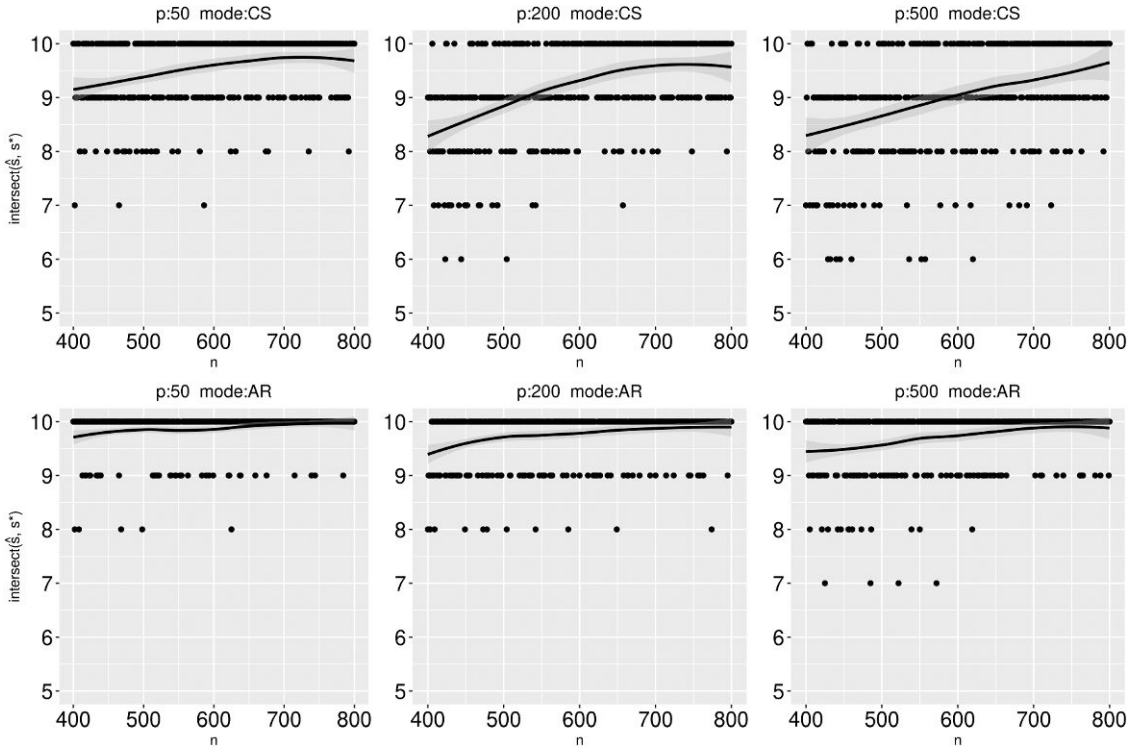
We can see from Table 4 that our proposal yields the sparsest model in terms of the number of nonzero rows, whereas the other three methods cannot produce a sparse model. Furthermore, the Pred value of SRRMLR is much lower than other methods, which indicates its superior power in distinguishing different categories. NNET performs poorly with the highest Pred value, which is expected because it results in a full model with too many negligible elements in its coefficient matrix. However, it is known that the images of handwritten Arabic numerals have structured patterns, and thus the dimension can be reduced further. With respect to the rank, only SRRMLR can yield a coefficient matrix with low rank and the estimated ranks using SRRMLR is nine. Considering that the MNIST data contains 10 categories and one category is used for baseline, the perfect rank of coefficient matrix estimation should be nine.

To provide further insight into the stability of the previous methods, we randomly select 100 samples from each category, and a total of 1,000 samples are used. We then treat 500 of them as training data and the remainder as test data. All of the aforementioned methods are performed using the training data, and the predictive performance is assessed on the test data. This random sampling process is replicated 100 times. The results are summarized in Table 5. Overall, the results are consistent with those in Table 4. Again, the proposed SRRMLR method has the best performance in terms of all measurements. In particular, the average prediction error of SRRMLR is the smallest, and the corresponding standard deviation is the lowest, indicating the superior and robust behavior of our proposed technique.

6. Discussion

In this paper, we propose a novel group best subset selection procedure in a reduced-rank multinomial logistic regression model to realize the goal of joint dimension reduction and variable selection. We develop an iterative algorithm

Figure 4. Scatterplots of the Number of Intersections of the Estimated Active Set $\hat{\mathcal{A}}$ and the True Active Set $\mathcal{A}^*$ During 100 Replications



Notes. In each subfigure, a smooth local polynomial regression fitting curve, as well as its confidence band, is added to show the trend over the sample size. The left, center, and right panels correspond to $p = 50$, 200, and 500, respectively. The top panel corresponds to the results of CS structure in $\boldsymbol{X}$, and the bottom panel corresponds to the results of AR structure in $\boldsymbol{X}$.

based on a group version of the primal-dual active set algorithm and the MM algorithm to efficiently solve the sparse reduced-rank multinomial logistic regression problem. We show that our proposed estimator enjoys important theoretical properties, including estimation ability and variable selection consistency. The simulation studies also demonstrate that our method performs better in identifying the correct model than well-known previous methods. Finally, we apply the proposed method to two real datasets from imaging classification and obtain meaningful results. Therefore, our new approach is a valuable toolbox for the high-dimensional multiclass classification problem.

Although SRRMLR performs satisfactorily in both simulated and real data, it still has a few aspects that can inspire further studies. For the tuning of r and s, we use a grid search here and set the upper bounds $r_{\max}$ and $s_{\max}$ according to the theoretical results in Section 3. A smarter strategy could be used to speed up the tuning procedure. For instance, we could set r to be its maximum possible value, $r = \min(p, q-1)$, and determine an appropriate value for s by searching all possible values along $\{1, \ldots, s_{\max}\}$. Then based on this s value, we could determine the optimal r value by solving the optimization problem in a setting with a much lower dimension. This one-dimensional searching strategy could lower the computational time substantially. It will be interesting to investigate how well this strategy guarantees estimation accuracy and row support recovery.

Table 4. Results for the Whole MNIST Data Set

Method	Pred	$\hat{r}$	$\lvert\hat{\mathcal{A}}\rvert$
SRRMLR	3,429.53	9	99
mLasso	11,906.89	—	100
NNET	8,337.42	—	100
RRVGLM	—	—	—

Note. —, these results cannot be obtained.

Table 5. Average Results on the Selected MNIST Data Set, with the Corresponding Standard Deviation Indicated in Parentheses

Method	Pred	$\hat{r}$	$\lvert\hat{\mathcal{A}}\rvert$
SRRMLR	341.12 (30.99)	8.61 (0.72)	17.98 (2.60)
mLasso	491.54 (35.08)	—	94.82 (2.53)
NNET	2,897.09 (629.21)	—	100(0)
RRVGLM	2,347.93 (913.31)	3.00 (3.45)	100(0)

Note. —, these results cannot be obtained.

References

Bayaga A (2010) Multinomial logistic regression: Usage and application in risk analysis. *J. Appl. Quant. Methods* 5(2):288–297.
Bertsimas D, Mundru N (2020) Sparse convex regression. *INFORMS J. Comput.* 33(1):262–279.
Bertsimas D, King A, Mazumder R (2016) Best subset selection via a modern optimization lens. *Annals Statist.* 44(2):813–852.
Bickel PJ, Ritov Y, Tsybakov AB (2009) Simultaneous analysis of lasso and dantzig selector. *Ann. Statist.* 37(4):1705–1732.
Bunea F, She Y, Wegkamp MH (2012) Joint variable and rank selection for parsimonious estimation of high-dimensional matrices. *Ann. Statist.* 40(5):2359–2388.
Cawley GC, Talbot NL, Girolami M (2006) Sparse multinomial logistic regression via Bayesian l1 regularisation. *Proc. 19th Internat. Conf. Neural Inform. Processing Systems,* 209–216.
Chen K (2016) Model diagnostics in reduced-rank estimation. *Statist. Interface* 9(4):469.
Chen L, Huang JZ (2012) Sparse reduced-rank regression for simultaneous dimension reduction and variable selection. *J. Amer. Statist. Assoc.* 107(500):1533–1545.
Chen K, Dong R, Xu W, Zheng Z (2022) Fast stagewise sparse factor regression. *J. Machine Learn. Res.* 23(271):1–45.
Chen K, Hoffman EA, Seetharaman I, Jiao F, Lin CL, Chan KS (2016) Linking lung airway structure to pulmonary function via composite bridge regression. *Ann. Appl. Statist.* 10(4):1880.
Cheng Y, Wang X, Xia Y (2020) Supervised t-distributed stochastic neighbor embedding for data visualization and classification. *INFORMS J. Comput.* 33(2):566–585.
Choi S, Hoffman EA, Wenzel SE, Castro M, Fain SB, Jarjour NN, Schiebler ML, et al. (2015) Quantitative assessment of multiscale structural and functional alterations in asthmatic populations. *J. Appl. Physiol.* 118(10):1286–1298.
Fan J, Li R (2001) Variable selection via nonconcave penalized likelihood and its oracle properties. *J. Amer. Statist. Assoc.* 96(456):1348–1360.
Fan Y, Lv J (2014) Asymptotic properties for combined l 1 and concave regularization. *Biometrika* 101(1):57–70.
Fan J, Peng H (2004) Nonconcave penalized likelihood with a diverging number of parameters. *Ann. Statist.* 32(3):928–961.
Hastie T, Tibshirani R, Friedman J (2009) *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (Springer Science & Business Media, Boston).
Huang J, Jiao Y, Liu Y, Lu X (2018) A constructive approach to l0 penalized regression. *J. Machine Learn. Res.* 19(1):403–439.
Izenman AJ (1975) Reduced-rank regression for the multivariate linear model. *J. Multivariate Anal.* 5(2):248–264.
Jiang S, Fang SC, Jin Q (2020) Sparse solutions by a quadratically constrained $\ell_q(0 < q < 1)$ minimization model. *INFORMS J. Comput.* 33(2):511–530.
Krishnapuram B, Carin L, Figueiredo MA, Hartemink AJ (2005) Sparse multinomial logistic regression: Fast algorithms and generalization bounds. *IEEE Trans. Pattern Anal. Machine Intelligence* 27(6):957–968.
Lange K (2016) *MM Optimization Algorithms* (Society for Industrial and Applied Mathematics, Philadelphia).
LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. *Proc. IEEE* 86(11):2278–2324.
Lee G, O'Leary JT, Lee SH, Morrison A (2002) Comparison and contrast of push and pull motivational effects on trip behavior: An application of a multinomial logistic regression model. *Tourist Anal.* 7(2):89–104.
Li J, Bioucas-Dias JM, Plaza A (2010) Semisupervised hyperspectral image segmentation using multinomial logistic regression with active learning. *IEEE Trans. Geosci. Remote Sensing* 48(11):4085–4098.
Li Y, Yang Q, Yang L, Lei N, Zheng K (2019) A scalable electrochemical dehydrogenative cross-coupling of p(o)h compounds with rsh/roh. *Chemical Comm.* 55:4981–4984.
Liu N, Ma Y, Topaloglu H (2020) Assortment optimization under the multinomial logit model with sequential offerings. *INFORMS J. Comput.* 32(3):835–853.
Maxwell MS, Restrepo M, Henderson SG, Topaloglu H (2010) Approximate dynamic programming for ambulance redeployment. *INFORMS J. Comput.* 22(2):266–281.
Meier L, Van De Geer S, Bühlmann P (2008) The group lasso for logistic regression. *J. Royal Statist. Soc. Ser. B Statist. Methodology* 70(1):53–71.
Mistry M, Letsios D, Krennrich G, Lee RM, Misener R (2020) Mixed-integer convex nonlinear optimization with gradient-boosted trees embedded. *INFORMS J. Comput.* 33(3):1103–1119.
Pandya D, Upadhyay S, Harsha SP (2014) Fault diagnosis of rolling element bearing by using multinomial logistic regression and wavelet packet transform. *Soft Comput.* 18(2):255–266.
Połap D, Srivastava G, Yu K (2021) Agent architecture of an intelligent medical system based on federated learning and blockchain technology. *J. Inform. Security Appl.* 58:102748.
Ripley B, Venables W, Ripley MB (2016) *Package 'nnet'. R package version 7(3-12):700* (R, Vienna).
She Y (2017) Selective factor extraction in high dimensions. *Biometrika* 104(1):97–110.
Simon N, Friedman J, Hastie T (2013) A blockwise descent algorithm for group-penalized multiresponse and multinomial regression. Preprint, submitted November 26, https://arxiv.org/abs/1311.6529.
Tibshirani R (1996) Regression shrinkage and selection via the lasso. *J. Royal Statist. Soc. B* 58:267–288.
Tutz G, Pössnecker W, Uhlmann L (2015) Variable selection in general multinomial logit models. *Comput. Statist. Data Anal.* 82:207–222.

Uematsu Y, Fan Y, Chen K, Lv J, Lin W (2019) Sofar: Large-scale association network learning. *IEEE Trans. Inform. Theory* 65(8):4924–4939.
Van der Maaten L, Hinton G (2008) Visualizing data using t-SNE. *J. Machine Learn. Res.* 9(November):2579–2605.
Vincent M, Hansen NR (2014) Sparse group lasso and high dimensional multinomial classification. *Comput. Statist. Data Anal.* 71:771–786.
Wang X (2009) Dimension reduction techniques in quasi-Monte Carlo methods for option pricing. *INFORMS J. Comput.* 21(3):488–504.
Wang Y (2005) A multinomial logistic regression modeling approach for anomaly intrusion detection. *Comput. Security* 24(8):662–674.
Wang L, Peng B, Bradic J, Li R, Wu Y (2020) A tuning-free robust and efficient approach to high-dimensional regression. *J. Amer. Statist. Assoc.* 115(532):1700–1714.
Wen C, Zhang A, Quan S, Wang X (2020) Bess: An r package for best subset selection in linear, logistic and cox proportional hazards models. *J. Statist. Software* 94(4):1–24.
Won D, Manzour H, Chaovalitwongse W (2020) Convex optimization for group feature selection in networked data. *INFORMS J. Comput.* 32(1):182–198.
Yee TW, Hastie TJ (2003) Reduced-rank vector generalized linear models. *Statist. Modeling* 3(1):15–41.
Yuan M, Lin Y (2006) Model selection and estimation in regression with grouped variables. *J. Royal Statist. Soc. Ser. B Statist. Methodology* 68(1):49–67.
Zheng Z, Fan Y, Lv J (2014) High dimensional thresholded regression and shrinkage effect. *J. Royal Statist. Soc. Ser. B Statist. Methodology* 76(3):627–649.
Zheng Z, Bahadori MT, Liu Y, Lv J (2019) Scalable interpretable multi-response regression via seed. *J. Machine Learn. Res.* 20(107):1–34.
Zhu J, Wen C, Zhu J, Zhang H, Wang X (2020) A polynomial algorithm for best-subset selection problem. *Proc. National Acad. Sci. USA* 117(52):33117–33123.