Covariate-Adjusted Tensor Classification in High dimensions*

Yuqing Pan , Qing Mai, and Xin Zhang†

Abstract

In contemporary scientific research, it is often of great interest to predict a categorical response based on a high-dimensional tensor (i.e. multi-dimensional array) and additional covariates. Motivated by applications in science and engineering, we propose a comprehensive and interpretable discriminant analysis model, called the CATCH model (in short for Covariate-Adjusted Tensor Classification in High-dimensions). The CATCH model efficiently integrates the covariates and the tensor to predict the categorical outcome. It also jointly explains the complicated relationships among the covariates, the tensor predictor, and the categorical response. The tensor structure is utilized to achieve easy interpretation and accurate prediction. To tackle the new computational and statistical challenges arising from the intimidating tensor dimensions, we propose a penalized approach to select a subset of the tensor predictor entries that affect classification after adjustment for the covariates. An efficient algorithm is developed to take advantage of the tensor structure in the penalized estimation. Theoretical results confirm that the proposed method achieves variable selection and prediction consistency, even when the tensor dimension is much larger than the sample size. The superior performance of our method over existing methods is demonstrated in extensive simulated and real data examples.

Key Words: Discriminant analysis; multicategory classification; multidimensional array; sparsity; tensor classification and regression.

*The authors are grateful to the Editor, Associate Editor and two referees for insightful comments that have led to significant improvements of this paper. The authors would like to thank Dr. Lexin Li for sharing the ADHD and ASD datasets, Dr. Qun Li and Dr. Dan Schonfeld for sharing their code for CMDA and DGTDA, and Dr. Elizabeth Slate for helpful discussion. Research for this paper was supported in part by grants CCF-1617691 and DMS-1613154 from the U.S. National Science Foundation.

†Yuqing Pan (yuqing.pan@stat.fsu.edu) is Ph.D. student, and Qing Mai (mai@stat.fsu.edu) and Xin Zhang (henry@stat.fsu.edu) are assistant professors, at Department of Statistics, Florida State University, Tallahassee, FL, 32306.

1 Introduction

Many contemporary scientific and engineering studies collect data from different categories of subjects in the form of multi-dimensional array, a.k.a. tensor, accompanied by additional covariates. For example, neuroimaging studies often try to identify and understand neurological and neurodevelopmental disorders based on tensor images, such as anatomical magnetic resonance imaging (MRI), positron emission tomography (PET), functional magnetic resonance imaging (fMRI), and electroencephalography (EEG), along with a few additional clinical covariates such as demographic information, psychological evaluations and cognitive scores. Such data also frequently arise in other fields such as computational biology, personalized recommendation, and image recognition analysis.

The increasing popularity of such data brings many new challenges to statistical analysis. First, it is generally unclear how to integrate the information from both the tensor predictor and the vector of covariates to achieve accurate classification. There also lack approaches to model the dependence between the covariates and the tensor, as well as their effects on the response. Second, the tensor predictor is often high-dimensional. For example, in our neuroimaging data applications, we use the structural magnetic resonance imaging (MRI) to study the attention deficit hyperactivity disorder (ADHD) and the autism spectrum disorder (ASD). Each MRI is a three-way tensor with dimension $30\times36\times30$ (ADHD) or $91\times109\times91$ (ASD), which adds up to over $30,000$ or $900,000$ entries for one subject. Moreover, rapid advancements in neuroimaging technology enable researchers to obtain tensor images with increasingly high resolutions and hence increasingly high dimensions. The high tensor dimensions call for new high-dimensional algorithms and methods. Finally, the theoretical studies are much more challenging for high-dimensional tensor methods than those for vector methods.

In this paper, we study the discriminant analysis with a high-dimensional tensor predictor $\mathbf{X} \in \mathbb{R}^{p_1\times\cdots\times p_M}$, $M \geq 2$, a low-dimensional covariate vector $\mathbf{U} \in \mathbb{R}^q$, and a class label $Y \in \{1,\ldots,K\}$ for $K \geq 2$ categories. We propose a unified framework called the CATCH model to jointly utilize information from the tensor and the covariates, while the intrinsic tensor-on-covariate relationship is accounted for through a regression model. We systemati-

cally investigate the effects of $\mathbf{X}$ and $\mathbf{U}$ on Y. First of all, $\mathbf{U}$ has a *direct effect* on separating the classes. Second, $\mathbf{U}$ also has an *indirect effect* on predicting Y through its influence on $\mathbf{X}$. Correctly adjusting for this indirect effect of $\mathbf{U}$ can substantially improve estimation, variable selection and the prediction of Y. Finally, we identify the effect of $\mathbf{X}$ on predicting Y after adjusting for $\mathbf{U}$. To achieve accurate estimation with a limited sample size, we perform variable selection on the covariate-adjusted tensor by adopting a penalized approach. Conceptually, our covariate-adjusted variable selection approach on tensor entries is similar to the partial dimension reduction methods (Chiaromonte et al. 2002, Feng et al. 2013) and net-effect plots (Cook 1998, Zhang 2013) in regression: we reduce the dimension of the tensor predictor without losing any information in classification after conditioning on the covariates.

While numerous high-dimensional classification methods have been developed for vector predictors, they become inefficient and may not scale well with tensor data. As previously mentioned, in many neuroimaging studies, simply vectorizing the tensor predictor results in a vector of length at the order of $10^4 \sim 10^6$. Moreover, many high-dimensional sparse classification methods (Cai & Liu 2011, Fan et al. 2012, Xu et al. 2015, e.g.) require computing the sample covariance, which leads to $10^8 \sim 10^{12}$ parameters. An intuitive remedy for this issue is to perform marginal screening in discriminant analysis (Pan et al. 2016, e.g). Although marginal screening is computationally efficient, it is well-known that marginally important predictors may not be jointly important, and vice versa. More importantly, both the vectorization approach and the marginal screening approach ignore the tensor structure and hence may lose important structural information. As we show in this paper, discarding the tensor structure deprives an opportunity of reducing the number of parameters.

Because of these issues, it is important to develop a method that models the entire tensor without sacrificing its tensor structure to preserve interpretability and improve efficiency. There has been an enormous body of literature on sparse linear discriminant analysis (LDA) for high-dimensional (vector) predictor (Cai & Liu 2011, Shao et al. 2011, Clemmensen et al. 2011, Witten & Tibshirani 2011, Fan et al. 2012, Mai et al. 2012, Xu et al. 2015, Mai et al. 2017, e.g.). However, with the tensor structure and the additional covariates, we face a much more

complicated high-dimensional problem that requires new statistical models, scalable algorithms and more involved theoretical studies.

In the literature, many matrix/tensor discriminant analysis methods have their roots in Fisher's discriminant analysis: they find linear projections (along each mode of the tensor) with the maximum separation between classes. See Zhong & Suslick (2015), Tao et al. (2007), Yan et al. (2005), Lai et al. (2013), Li & Schonfeld (2014), Bao & Chien (2015), Zeng et al. (2015) for example. Our method has significant distinctions from the existing ones. First, existing methods typically do not consider how to incorporate information from these additional covariates. Second, our proposal is based on a probabilistic model instead of the maximization of between-class variability. The probabilistic model enables us to provide strong theoretical guarantee for our method. Our method has variable selection and prediction consistency for ultra-high-dimensional tensors. In addition, our algorithm is shown to converge to the global minimizer with a probability of 1. Third, our numerical studies confirm superb performances of our method in terms of classification accuracy, variable selection, and computational time.

Logistic regression is another popular approach for tensor classification. For example, Zhou et al. (2013) adopted the tensor low-rank structure in a generalized linear model, while Wimalawarne et al. (2016) proposed to add various tensor norms as penalties to the logistic loss. These methods focus on binary classification, but our method is naturally applicable to multiclass problems. Moreover, unlike our theoretical studies, there is no theoretical analysis for classification error or variable selection in Zhou et al. (2013) and Wimalawarne et al. (2016).

It is worth-mentioning that our discriminant analysis framework is related to, but fundamentally different from, recent developments in tensor regression and tensor decomposition. Although discriminant analysis is one of the most common statistical tasks, it has received relatively less attention than regression in the research of tensor data. Many researchers have studied matrix- and tensor-variate regression (Zhou et al. 2013, Zhou & Li 2014, Zhao & Leng 2014, Hoff 2015, Raskutti & Yuan 2015, Sun et al. 2016, Wang & Zhu 2017, Li & Zhang 2017, Zhang & Li 2017, Lock 2017, e.g.). But most of these methods do not directly apply to classification. Moreover, many existing statistical methods on tensor data rely on multi-linear

tensor decomposition (Kolda & Bader 2009, Chi & Kolda 2012, Liu et al. 2017, Zhang & Xia 2017, e.g.). They assume that the tensor has a low-rank structure. Our approach does not require low-rank assumptions on the tensor predictor. Instead, we directly identify the important voxels and thus achieve variable selection and parsimonious modeling. By eliminating the low-rank assumption, we avoid the challenging problem of determining the true rank. On the other hand, the penalization approach proposed in this paper can be easily modified by specifying different penalty terms on different regions of the tensor to incorporate prior information such as smoothness, regions of interests, and regions of gray or white matters in brain images.

The contributions of this article are multi-fold. First, it provides a framework to jointly model the relationships among the categorical response, the multivariate covariates and the high-dimensional tensor predictor. Our CATCH model offers a useful paradigm for systematically and simultaneously studying the tensor-on-covariate regression, and the covariate-on-response, tensor-on-response classification. Second, while existing high-dimensional classification methods concentrate on a vector predictor, we develop new computational and theoretical techniques for classification with tensor. Third, our proposal advances the recent developments of tensor data analysis. While existing approaches largely rely on tensor regression and especially tensor low-rank decomposition, we focus on discriminant analysis and classification. Our method provides an alternative way of tensor dimension reduction by introducing group sparsity directly based on the Bayes' rule and hence achieves optimal classification.

The rest of this paper is organized as follows. In Section 2, we introduce the CATCH model and define the direct and indirect effects in the model. The estimation procedure is introduced in Section 3; and an efficient algorithm for the penalized estimation is developed in Section 4. Section 5 contains theoretical studies of both non-asymptotic and asymptotic properties of the proposed method in ultra-high dimensional settings. Extensive simulations in Section 6 and two real data applications in Section 7 confirm the advantages of our method over existing methods. Finally, Section 8 contains a short discussion and the Supplementary Materials contain additional numerical studies, along with proofs and other technical details. An `R` package `catch` is available at `https://cran.r-project.org/web/packages/catch/`.

2 The CATCH Model

2.1 Review of tensor notation

We first introduce standard tensor notation and operations (Kolda & Bader 2009, for example) that are used frequently in this paper. For positive integers $M \geq 2$, $p_1, \ldots, p_M$, a multidimensional array $\mathbf{A} \in \mathbb{R}^{p_1 \times \cdots \times p_M}$ is referred to as an *M-way* or *M-th order tensor*. The vectorization of a tensor $\mathbf{A}$, $\mathrm{vec}(\mathbf{A})$, is a $(\prod_m p_m \times 1)$ column vector, with $A_{i_1 \cdots i_M}$ being its j-th element, where $j = 1 + \sum_{k=1}^{M}(i_k - 1)\prod_{k'=1}^{k-1} p_{k'}$. The *mode-$k$ matricization*, $\mathbf{A}_{(k)}$, is a matrix of dimension $(p_k \times \prod_{m \neq k} p_m)$, with $A_{i_1 \cdots i_M}$ being its (i_k, j)-th element, where $j = 1 + \sum_{k' = k}(i_{k'} - 1)\prod_{l<k', l \neq k} p_l$. If we fix every index of the tensor but one, then we have a *fiber*. For example, $A_{i_1 \cdots i_{k-1} I_k i_{k+1} \cdots i_M}$ for $I_k = 1, \ldots p_k$, form a $(p_k \times 1)$ vector that is called the mode-k fiber of $\mathbf{A}$.

The *mode-k product* of a tensor $\mathbf{A}$ and a matrix $\boldsymbol{\alpha} \in \mathbb{R}^{d \times p_k}$, denoted by $\mathbf{A} \times_k \boldsymbol{\alpha}$, is an M-way tensor of dimension $p_1 \times \cdots \times p_{k-1} \times d \times p_{k+1} \times \cdots \times p_M$, with each element being the product of a mode-k fiber of $\mathbf{A}$ and a row vector of $\boldsymbol{\alpha}$. The *mode-k vector product* of a tensor $\mathbf{A}$ and a vector $\mathbf{c} \in \mathbb{R}^{p_k}$, denoted by $\mathbf{A} \bar{\times}_k \mathbf{c}$, is a $(M-1)$-way tensor of dimension $p_1 \times \cdots \times p_{k-1} \times p_{k+1} \times \cdots \times p_M$, with each element being the inner product of a mode-k fiber of $\mathbf{A}$ and $\mathbf{c}$. The *Tucker decomposition* of a tensor is defined as $\mathbf{A} = \mathbf{C} \times_1 \mathbf{G}_1 \times_2 \mathbf{G}_2 \cdots \times_M \mathbf{G}_M$, where $\mathbf{C} \in \mathbb{R}^{d_1 \times \cdots \times d_M}$ is the *core tensor*, and $\mathbf{G}_k \in \mathbb{R}^{p_k \times d_k}$, $k = 1, \ldots, M$, are the *factor matrices*. We write the Tucker decomposition as $[\![\mathbf{C}; \mathbf{G}_1, \ldots, \mathbf{G}_m]\!]$ in short. In particular, we frequently use the fact that $\mathrm{vec}([\![\mathbf{C}; \mathbf{G}_1, \ldots, \mathbf{G}_M]\!]) = (\mathbf{G}_M \otimes \cdots \otimes \mathbf{G}_1)\,\mathrm{vec}(\mathbf{C}) = \left(\bigotimes_{m=M}^{1} \mathbf{G}_m\right)\mathrm{vec}(\mathbf{C})$, where $\otimes$ denotes the Kronecker product and $\bigotimes_{m=M}^{1} \mathbf{G}_m$ is short for $\mathbf{G}_M \otimes \cdots \otimes \mathbf{G}_1$.

We further introduce the tensor normal (TN) distribution as a generalization of the matrix normal distribution (Gupta & Nagar 1999). A tensor random variable $\mathbf{Z} \in \mathbb{R}^{p_1 \times \cdots \times p_M}$ is called a standard tensor normal random variable if all elements of $\mathbf{Z}$ independently follow the (univariate) standard normal distribution. If $\mathbf{X} = \boldsymbol{\mu} + [\![\mathbf{Z}; \boldsymbol{\Sigma}_1^{1/2}, \ldots, \boldsymbol{\Sigma}_M^{1/2}]\!]$, we say $\mathbf{X}$ follows a tensor normal distribution $\mathbf{X} \sim TN(\boldsymbol{\mu}, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_M)$, where $\boldsymbol{\Sigma}_j > 0$ imposes the dependence structure on the j-th mode. Hence, $\mathrm{vec}(\mathbf{X}) = \mathrm{vec}(\boldsymbol{\mu}) + \boldsymbol{\Sigma}^{1/2}\mathrm{vec}(\mathbf{Z})$, where $\boldsymbol{\Sigma} = \bigotimes_{m=M}^{1} \boldsymbol{\Sigma}_m$.

2.2 The model assumptions

Consider a random triplet $\{Y, \mathbf{U}, \mathbf{X}\}$, where $Y \in \{1, \ldots, K\}$ is the class label for $K \geq 2$ classes, $\mathbf{U} \in \mathbb{R}^q$ is a vector of covariates, and $\mathbf{X} \in \mathbb{R}^{p_1 \times \cdots \times p_M}$ is an M-th order tensor-variate predictor, $M \geq 2$. Throughout this paper, we assume that $\Pr(Y = k) = \pi_k > 0$ where $\sum_{k=1}^{K} \pi_k = 1$. Our goal is to build a classifier that accurately predicts Y based on integrated information from $\mathbf{U}$ and $\mathbf{X}$. To this end, we propose the CATCH (covariate-adjusted tensor classification in high dimensions) model:

$$\mathbf{U} \mid (Y = k) \sim N(\boldsymbol{\phi}_k, \boldsymbol{\Psi}), \tag{2.1}$$

$$\mathbf{X} \mid (\mathbf{U} = \mathbf{u}, Y = k) \sim TN(\boldsymbol{\mu}_k + \boldsymbol{\alpha} \bar{\times}_{(M+1)} \mathbf{u}, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_M), \tag{2.2}$$

where $\boldsymbol{\phi}_k \in \mathbb{R}^q$, $\boldsymbol{\Psi} \in \mathbb{R}^{q \times q}$, $\boldsymbol{\alpha} \in \mathbb{R}^{p_1 \times \cdots \times p_M \times q}$, $\boldsymbol{\mu}_k \in \mathbb{R}^{p_1 \times \cdots p_M}$, and $\boldsymbol{\Sigma}_m \in \mathbb{R}^{p_m \times p_m}$, $m = 1, \ldots, M$. The symmetric matrices $\boldsymbol{\Psi}$ and $\boldsymbol{\Sigma}_m$ are positive definite. All these parameters have natural interpretations. In (2.1), we assume that $\{Y, \mathbf{U}\}$ follows the classical LDA model, where $\boldsymbol{\phi}_k$ is the mean of $\mathbf{U}$ within class k and $\boldsymbol{\Psi}$ is the common within-class covariance. Similarly, in (2.2), we assume a common within-class covariance structure of $\mathbf{X}$ characterized by $\boldsymbol{\Sigma}_m, m = 1, \ldots, M$, which does not depend on Y after we adjust for the covariates $\mathbf{U}$. The tensor coefficient $\boldsymbol{\alpha}$ characterizes the linear dependence of the tensor predictor $\mathbf{X}$ on the covariates $\mathbf{U}$, and $\boldsymbol{\mu}_k$ is the covariate-adjusted within-class mean of $\mathbf{X}$ in class k.

Our CATCH model is based on the linear discriminant analysis (LDA) model. Although the LDA model may seem stringent, many results in the literature support their applications in practice. For example, Michie et al. (1994), Hand (2006) reported that LDA is competitive on many benchmark datasets, while Cai & Liu (2011), Shao et al. (2011), Clemmensen et al. (2011), Witten & Tibshirani (2011), Fan et al. (2012), Mai et al. (2012), Xu et al. (2015) presented the accurate classification results of sparse LDA methods on high-dimensional datasets. These encouraging results lead us to consider the models in (2.1)–(2.2) for tensor classification. Our real data analysis in Section 7 also confirms that the classifier based on the CATCH model achieves accurate prediction in practice compared to many well-known classifiers. Hence, we expect our classifier to be widely applicable, while model assumptions such as normality are

imposed to provide intuition. Meanwhile, from the statistical perspective, it remains of great interest to develop classifiers under weaker model assumptions. See Section 8 for discussion along this line. We leave this topic for future research.

An important special case of the CATCH model applies to the situation where we only have $\{\mathbf{X}, Y\}$, but not the covariate $\mathbf{U}$. In this case, (2.1)–(2.2) reduce to

$$\mathbf{X} \mid (Y = k) \sim TN(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_m). \tag{2.3}$$

The model in (2.3) implies that the tensor predictor $\mathbf{X}$ follows the tensor normal distribution with different means but a common covariance structure. We refer to the model in (2.3) as the tensor discriminant analysis (TDA) model. It is a natural extension of LDA to incorporate the tensor structure, and is different from modeling $\{Y, \mathrm{vec}(\mathbf{X})\}$ using the classical LDA. By utilizing the tensor structure, we greatly reduce the number of free parameters. In the LDA model on the predictor $\mathrm{vec}(\mathbf{X})$ of dimension $(\prod_{m=1}^{M} p_m) \times 1$, the covariance matrix has $\prod_{m=1}^{M} p_m^2$ elements. In contrast, the covariance structure in (2.3) takes advantage of the tensor structure and only has $\sum_{m=1}^{M} p_m^2$ elements. In Section 4, we show that this structure leads to convenience in computation.

When both covariates and tensor are present, the CATCH model not only simultaneously characterizes the effect of $\mathbf{U}$ and $\mathbf{X}$ on Y, but also models the relationship of $\mathbf{X}$ on $\mathbf{U}$ within each class. To gain more insights, within each class k, we can write (2.2) as

$$\mathbf{X} = \boldsymbol{\mu}_k + \boldsymbol{\alpha} \bar{\times}_{(M+1)} \mathbf{U} + \mathbf{E}, \quad \mathbf{E} \sim TN(0, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_M), \tag{2.4}$$

where $\mathbf{E}$ is the unobservable statistical error independent of $\mathbf{U}$. Equation (2.4) coincides with the tensor response regression (TRR) model proposed by Li & Zhang (2017). The tensor parameter $\boldsymbol{\mu}_k$ is the adjusted mean of $\mathbf{X}$ in class k after removing the effect of covariates $\mathbf{U}$ on $\mathbf{X}$. Estimation and inference of $\boldsymbol{\alpha}$ and $\boldsymbol{\mu}_j - \boldsymbol{\mu}_k$, $j \neq k$, are of great interest in neuroimaging analysis and applications, where $\boldsymbol{\alpha}$ describes the effect of covariates and $\boldsymbol{\mu}_j - \boldsymbol{\mu}_k$ compares tensor images across classes after adjusting for covariates. Although this paper does not focus on studying the interrelationship between $\mathbf{X}$ and $\mathbf{U}$, accounting for this intrinsic regression relation in (2.4) often brings substantial gain in prediction, estimation and even variable selection.

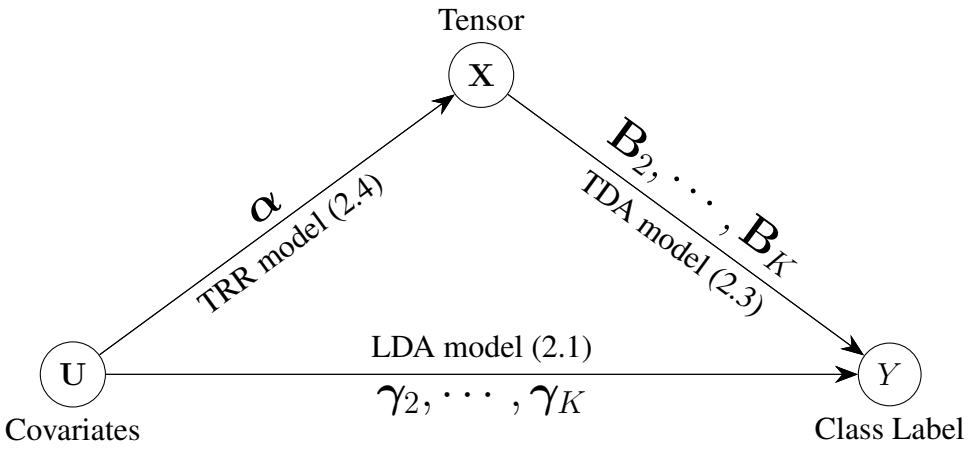

Figure 2.1: Graphical illustration of the direct and indirect effects in the CATCH model. The direct effect of $\mathbf{U}$ on Y, $\{\boldsymbol{\gamma}_2, \ldots, \boldsymbol{\gamma}_K\}$, coincides with the discriminative coefficients in the classical linear discriminant analysis (LDA; 2.1) model. The parameters $\{\mathbf{B}_2, \ldots, \mathbf{B}_k\}$ represent the direct effect of $\mathbf{X}$ on Y in the tensor discriminant analysis (TDA; 2.3) model. The parameter $\boldsymbol{\alpha}$ is the indirect effect of $\mathbf{U}$ on Y through affecting $\mathbf{X}$ by the tensor response regression (TRR; 2.4).

We explain this phenomenon in the following section, where we define the direct and indirect effects.

2.3 The direct and indirect effects

Since our goal is to predict Y based on $\{\mathbf{U}, \mathbf{X}\}$, we derive the optimal classifier – the so called "Bayes' rule" – under the CATCH model. Estimation of this classifier is discussed later in Section 3. Given $\mathbf{X}$ and $\mathbf{U}$, the Bayes' rule that achieves the lowest error rate possible is (e.g. Friedman et al. (2001)),

$$\widehat{Y} = \arg\max_{k=1,\ldots,K} \Pr(Y = k \mid \mathbf{X} = \mathbf{x}, \mathbf{U} = \mathbf{u}) = \arg\max_{k=1,\ldots,K} \pi_k f_k(\mathbf{x}, \mathbf{u}), \tag{2.5}$$

where $\pi_k = \Pr(Y = k)$ and $f_k(\mathbf{x}, \mathbf{u})$ is the joint probability density function of $\mathbf{X}$ and $\mathbf{U}$ conditional on $Y = k$. We have the following results under the CATCH model.

Proposition 1. *The Bayes' rule of CATCH model (2.1, 2.2) is*

$$\widehat{Y} = \arg\max_{k=1,\ldots K} \left\{ a_k + \boldsymbol{\gamma}_k^{\mathrm{T}} \mathbf{U} + \langle \mathbf{B}_k, \mathbf{X} - \boldsymbol{\alpha} \bar{\times}_{(M+1)} \mathbf{U} \rangle \right\}, \tag{2.6}$$

where $\boldsymbol{\gamma}_k = \boldsymbol{\Psi}^{-1}(\boldsymbol{\phi}_k - \boldsymbol{\phi}_1)$, $\mathbf{B}_k = [\![\boldsymbol{\mu}_k - \boldsymbol{\mu}_1; \boldsymbol{\Sigma}_1^{-1}, \ldots, \boldsymbol{\Sigma}_M^{-1}]\!]$, *and* $a_k = \log(\pi_k/\pi_1) - \frac{1}{2}\boldsymbol{\gamma}_k^{\intercal}(\boldsymbol{\phi}_k + \boldsymbol{\phi}_1) - \langle \mathbf{B}_k, \frac{1}{2}(\boldsymbol{\mu}_k + \boldsymbol{\mu}_1)\rangle$ *is a scalar that does not involve* $\mathbf{X}$ *or* $\mathbf{U}$.

The parameters $\{\boldsymbol{\gamma}_k, \boldsymbol{\alpha}, \mathbf{B}_k\}$ are critical in (2.6). We discuss their different interpretations as direct and indirect effects. First, the discriminative coefficient vector $\boldsymbol{\gamma}_k \in \mathbb{R}^q$ is the direct effect of $\mathbf{U}$ on classification. It coincides with the usual LDA discriminative directions in (2.1). Second, the tensor regression coefficient $\boldsymbol{\alpha} \in \mathbb{R}^{p_1 \times \cdots \times p_M \times q}$ is the indirect effect of $\mathbf{U}$ on Y. It characterizes how $\mathbf{U}$ affects Y through its relationship with $\mathbf{X}$. Finally, the discriminative coefficient tensor $\mathbf{B}_k \in \mathbb{R}^{p_1 \times \cdots \times p_M}$ is the direct effect of $\mathbf{X}$ after adjusted for $\mathbf{U}$. In absence of the covariates $\mathbf{U}$, the Bayes' rule of the TDA model (2.3) is $\widehat{Y} = \arg\max_{k=1,\ldots K} \left\{\log(\pi_k/\pi_1) + \langle \mathbf{B}_k, \mathbf{X} - \frac{1}{2}(\boldsymbol{\mu}_k + \boldsymbol{\mu}_1)\rangle\right\}$, where $\mathbf{B}_k$ is defined in Proposition 1. A graphical illustration of the direct and indirect effects is in Figure 2.1.

When the covariates $\mathbf{U}$ have different means $\boldsymbol{\phi}_k$ in each class, they directly contribute to the separation of the classes along the discriminative directions $\boldsymbol{\gamma}_k = \boldsymbol{\Psi}^{-1}(\boldsymbol{\phi}_k - \boldsymbol{\phi}_1)$. Somewhat surprisingly, even when the covariates have no direct effect on separating classes, i.e. $\boldsymbol{\phi}_1 = \cdots = \boldsymbol{\phi}_K$, the inclusion of $\mathbf{U}$ can still bring substantial gain in classification by modeling its indirect effect. This can be seen from comparing the Bayes' errors. Define $R(\mathbf{U}, \mathbf{X})$ as the lowest classification error rate possible if we build the classifier based on $\mathbf{X}$ and $\mathbf{U}$, and similarly, $R(\mathbf{X})$ as that based only on $\mathbf{X}$. The explicit expressions of $R(\mathbf{X})$ and $R(\mathbf{U}, \mathbf{X})$ are given in the Supplementary Materials. The following example demonstrates that ignoring the indirect effect of covariates greatly inflates the classification error and produces misleading variable selection results.

Example 1. *Consider a binary classification example* $Y = 1$ *or* 2 *with equal class probabilities* $\pi_1 = \pi_2 = 0.5$. *The covariates* $U \mid (Y = 1) \sim U \mid (Y = 2) \sim N(0, 1)$ *have no direct effect on classifying* Y. *The tensor* $\mathbf{X}$ *is a* 2×2 *matrix, with* $\mathrm{E}(\mathbf{X} \mid U = u, Y = k) = \boldsymbol{\mu}_k + \boldsymbol{\alpha} \cdot u$, *where* $\boldsymbol{\mu}_1 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$, $\boldsymbol{\mu}_2 = \begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix}$ *and* $\boldsymbol{\alpha} = \begin{pmatrix} \alpha & 0 \\ \alpha & 0 \end{pmatrix}$, *and the covariances* $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \mathbf{I}_2$. *Under this model, the discriminative coefficient matrix* $\mathbf{B}_2 = \boldsymbol{\Sigma}_1^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)\boldsymbol{\Sigma}_2^{-1}$ *is zero everywhere except for its first element, indicating that only* X_{11} *has an effect on classification after we adjust for* U. *If we ignore* U, *then* $\mathbf{X} \mid (Y = k)$ *is no longer a matrix normal random*

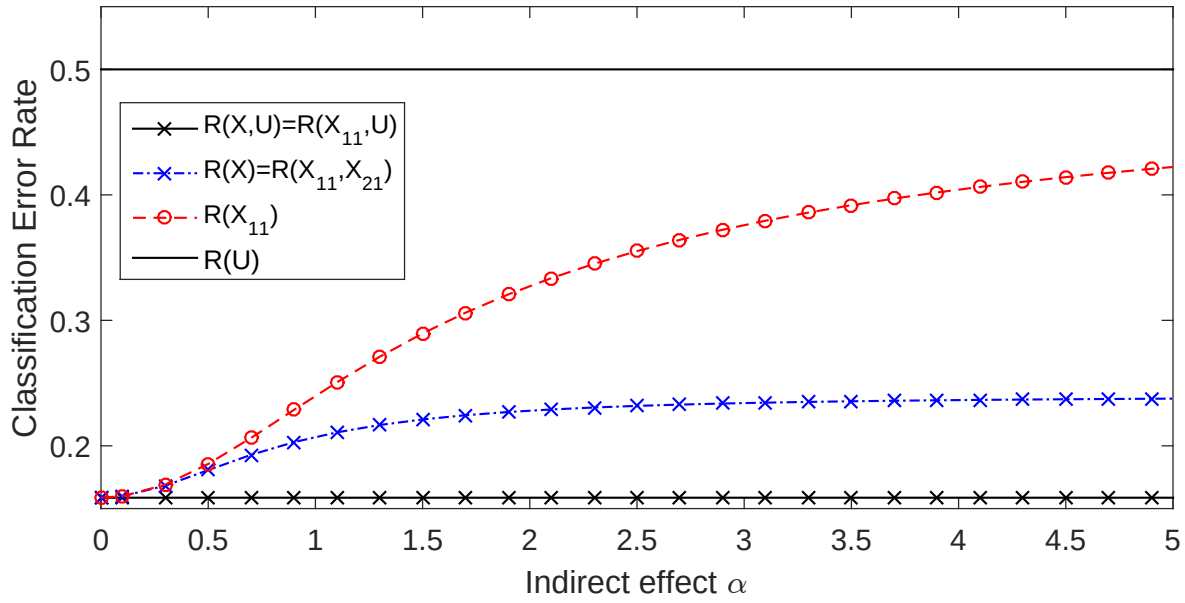

Figure 2.2: Graphical illustration of the best possible classification error rates in Example 1.

variable, but $\mathrm{vec}(\mathbf{X}) \mid (Y = k)$ *is multivariate normal with mean* $\mathrm{vec}(\boldsymbol{\mu}_k)$*, and covariance* $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_2 \otimes \boldsymbol{\Sigma}_1 + \mathrm{vec}(\boldsymbol{\alpha})\mathrm{vec}^{\intercal}(\boldsymbol{\alpha})$*. By straightforward calculation (see Supplementary Materials), we have* $R(U) = 0.5$*,* $R(\mathbf{X}, U) = R(X_{11}, U) = 1 - \Phi(1) = 0.1587$*, where* $\Phi(\cdot)$ *is the cumulative distribution function of* $N(0, 1)$*. If we ignore* $\mathbf{U}$*,* $R(\mathbf{X}) = R(X_{11}, X_{21}) = 1 - \Phi(\sqrt{2 + \alpha^2}/\sqrt{2 + 2\alpha^2}) < R(X_{11}) = 1 - \Phi(1/\sqrt{1 + \alpha^2})$*. As the number* $\alpha \to \infty$*, i.e., as the indirect effect becomes stronger, the error rates* $R(X_{11}) \to 0.5$ *and* $R(\mathbf{X}) \to 1 - \Phi(1/\sqrt{2}) = 0.2398$.

We further plot the error rates using different predictors versus α in Figure 2.2. The Bayes' error is always $R(\mathbf{X}, U) = R(X_{11}, U) = 0.1587$, indicating that the magnitude of the indirect effect $\boldsymbol{\alpha}$ does not affect the classification error if we adjust for U correctly. When we fail to adjust for U, $R(X_{11})$ increases drastically with α and eventually converges to 0.5. Meanwhile, $R(\mathbf{X}) = R(X_{11}, X_{21})$ increases quickly with α and eventually converges to around 0.2398, which is much larger than the Bayes' error of 0.1587 with U. On the other hand, in this example, to achieve the best classification error, we only need one element of the tensor predictor, X_{11},

if we adjust for U, but we need two elements X_{11} and X_{21} if not. Example 1 hence also exhibits the potential impact of the covariates on variable selection of $\mathbf{X}$: the best possible classifier based on $\mathbf{X}$ may have a different sparsity pattern, depending on whether we adjust for covariates correctly.

The above example reveals the important role of covariates even when the covariates are marginally independent of the class label. This observation implies that, even in experiments where the covariates are carefully controlled to be roughly the same within each class in the sampling stage, it is still beneficial to include and adjust for covariates to remove their indirect effect.

2.4 The tensor discriminative set and the multi-class group sparsity

To estimate the Bayes' rule in Proposition 1, we need to estimate $\boldsymbol{\alpha}$ and $\{\pi_k, \boldsymbol{\phi}_k, \boldsymbol{\gamma}_k, \boldsymbol{\mu}_k, \mathbf{B}_k\}_{k=1}^K$. By definition, we have $\boldsymbol{\gamma}_1 = 0, \mathbf{B}_1 = 0$, so we only need to estimate $\boldsymbol{\gamma}_k, \mathbf{B}_k$ for $k = 2, \ldots, K$. In this paper we focus on low-dimensional covariates and high-dimensional tensor predictor, but we briefly discuss a possible extension to high-dimensional covariates in Section 3.1. Henceforth, we assume that (n, p, q) satisfies $q \ll n \ll \prod_{m=1}^M p_m$ unless stated otherwise.

Since $\mathbf{U}$ is low-dimensional, the estimation of parameters $\boldsymbol{\phi}_k$, $\boldsymbol{\gamma}_k$ and $\boldsymbol{\Psi}$ is relatively straightforward. On the other hand, although $\boldsymbol{\alpha}$ is high-dimensional, it is connected to the tensor regression model and can be estimated easily under the $q < n$ scenario. However, the estimation of tensor coefficients $\mathbf{B}_k$, $k = 2, \ldots, K$, is more challenging since they are high-dimensional and depends on the covariance structures $\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_M$. In practice we typically do not have a large enough sample size to accurately estimate all the $(K-1) \cdot \prod_{m=1}^M p_m$ coefficients in $\mathbf{B}_2, \ldots, \mathbf{B}_K$ without additional assumptions. Therefore, we employ the well-received sparsity assumption, which is shown to be crucial in high-dimensional classification (Bickel & Levina 2004, Fan & Fan 2008, e.g.).

From Proposition 1, an entry of the tensor predictor $X_{j_1 \cdots j_M}$ has an effect on the final classification (after adjusted for the covariates) if and only if $b_{k, j_1 \cdots j_M} \neq 0$ for some k, where $b_{k, j_1 \cdots j_M}$ is the $(j_1 \cdots j_M)$-th entry of $\mathbf{B}_k$. Hence, we introduce our notion of tensor discriminative set in

the CATCH model. Define the discriminative set $\mathcal{D}$ and its complement set $\mathcal{D}^c$ as follows:

$$\mathcal{D} = \{(j_1, \ldots, j_M) : b_{k,j_1 \cdots j_M} \neq 0 \text{ for some } k\}, \tag{2.7}$$

$$\mathcal{D}^c = \{(j_1, \ldots, j_M) : b_{1,j_1 \cdots j_M} = \cdots = b_{K,j_1 \cdots j_M} = 0\}. \tag{2.8}$$

Denote $d = |\mathcal{D}|$ as the number of nonzero entries. The sparsity assumption then requires d to be much smaller than the dimension $\prod_{m=1}^{M} p_m$, so that most of the predictors belong to the complement set $\mathcal{D}^c$.

Examination of $\mathcal{D}^c$ reveals that the coefficients in $\mathbf{B}_2, \ldots \mathbf{B}_K$ have a group sparsity structure across classes. For any $(j_1, \ldots, j_M)$, the coefficients $(b_{2,j_1 \cdots J_M}, \ldots, b_{K,j_1 \cdots j_M})$ are all coefficients for the same voxel $X_{j_1 \cdots j_M}$; they are the effects of $X_{j_1 \cdots j_M}$ in separating different pairs of classes. When $X_{j_1 \cdots j_M}$ is not important, i.e., has no effect in separating any pair of classes, all its coefficients have to be 0. Consequently, we have the group sparsity structure across classes (rather than across voxels). For vector data, Hastie et al. (2015) considered a similar group sparsity assumption across classes in multinomial regression, which shares some spirit with our assumption. We remark here, though, that the group structure is present only when $K > 2$. When $K = 2$, we only need one set of coefficients $\mathbf{B}_2$ to separate two classes and d becomes the number of nonzeros in $\mathbf{B}_2$. It follows that (2.7) and (2.8) reduce to $\mathcal{D} = \{(j_1, \ldots, j_M) : b_{2,j_1 \cdots j_M} \neq 0\}$ and $\mathcal{D}^c = \{(j_1, \ldots, j_M) : b_{2,j_1 \cdots j_M} = 0\}$, which resembles the more familiar form of sparsity, such as that in regression problems (Tibshirani 1996).

3 Estimation Procedure

In this section, we assume that we have i.i.d. samples $\{Y^i, \mathbf{U}^i, \mathbf{X}^i\}_{i=1}^n$ and discuss the construction of an accurate classifier based on the data. With a little abuse of notation, we set $\mathbf{Y} \in \mathbb{R}^n$ as a vector that contains all the observed class labels, $\mathbf{U} \in \mathbb{R}^{q \times n}$ as a matrix that contains all the observed covariates and $\mathbf{X} \in \mathbb{R}^{p_1 \times \cdots \times p_M \times n}$ as a $(M+1)$-way tensor that contains all the observed tensor data. As discussed in Section 2.4, the sparsity assumption is only imposed on $\mathbf{B}_k$ and not on other parameters $\boldsymbol{\alpha}, \{\pi_k, \boldsymbol{\phi}_k, \boldsymbol{\gamma}_k, \boldsymbol{\mu}_k\}_{k=1}^K$ or $\{\boldsymbol{\Sigma}_m\}_{m=1}^M$. We hence separately

discuss the unpenalized estimators of $\boldsymbol{\alpha}, \{\pi_k, \boldsymbol{\phi}_k, \boldsymbol{\gamma}_k, \boldsymbol{\mu}_k\}_{k=1}^K$ and $\{\boldsymbol{\Sigma}_m\}_{m=1}^M$ in Sections 3.1 and 3.2, and the penalized estimators of $\mathbf{B}_2, \ldots, \mathbf{B}_K$ in Section 3.3.

3.1 Estimation of $\{\pi_k, \boldsymbol{\phi}_k, \boldsymbol{\gamma}_k, \boldsymbol{\mu}_k\}_{k=1}^K$ and $\boldsymbol{\alpha}$

Let $\overline{\mathbf{U}}_k$ be the sample mean of $\mathbf{U}$ within Class k, and $\overline{\mathbf{X}}_k$ be the sample mean of $\mathbf{X}$ within Class k. We estimate $\{\pi_k, \boldsymbol{\phi}_k, \boldsymbol{\gamma}_k\}$ straightforwardly using the following maximum likelihood estimators (MLE) under the CATCH model (2.1, 2.2),

$$\widehat{\pi}_k = \frac{1}{n}\sum_{i=1}^n 1(Y^i = k), \quad \widehat{\boldsymbol{\phi}}_k = \overline{\mathbf{U}}_k, \quad \widehat{\boldsymbol{\gamma}}_k = \widehat{\boldsymbol{\Psi}}^{-1}(\widehat{\boldsymbol{\phi}}_k - \widehat{\boldsymbol{\phi}}_1), \quad k = 1, \ldots, K, \tag{3.1}$$

where $\widehat{\boldsymbol{\Psi}} = \frac{1}{n}\sum_{k=1}^K \sum_{y^i=k}(\mathbf{U}^i - \widehat{\boldsymbol{\phi}}_k)(\mathbf{U}^i - \widehat{\boldsymbol{\phi}}_k)^\intercal$.

Meanwhile, the MLE for $\boldsymbol{\alpha}$ can be most succinctly expressed using tensor products and the group-wise centered data: for the observations within class k, i.e. $Y^i = k$, let $\widetilde{\mathbf{X}}^i = \mathbf{X}^i - \overline{\mathbf{X}}_k$ and $\widetilde{\mathbf{U}}^i = \mathbf{U}^i - \overline{\mathbf{U}}_k$ and define $\widetilde{\mathbf{X}} \in \mathbb{R}^{p_1 \times \cdots \times p_M \times n}$ and $\widetilde{\mathbf{U}} \in \mathbb{R}^{q \times n}$ to be the tensor and the matrix that consist of $\widetilde{\mathbf{X}}^i$ and $\widetilde{\mathbf{U}}^i$, respectively.

Lemma 1. *Under the CATCH model* (2.1)–(2.2)*, the MLE for* $\boldsymbol{\alpha}$ *is*

$$\widehat{\boldsymbol{\alpha}} = \widetilde{\mathbf{X}} \times_{(M+1)} \{(\widetilde{\mathbf{U}}\widetilde{\mathbf{U}}^\intercal)^{-1}\widetilde{\mathbf{U}}\}. \tag{3.2}$$

Note that we assume $q \ll n$. Hence, $(\widetilde{\mathbf{U}}\widetilde{\mathbf{U}}^\intercal) \in \mathbb{R}^{q \times q}$ is invertible and (3.2) is a feasible estimator for the tensor coefficient $\boldsymbol{\alpha}$. To gain more intuition, note that (2.2) implies that, for each $X_{j_1 \cdots j_M}$, within class k, we have

$$X_{j_1 \cdots j_M} - \mu_{k, j_1 \cdots j_M} = \boldsymbol{\alpha}_{j_1 \cdots j_M}^\intercal \mathbf{U} + \epsilon_{j_1 \cdots j_M} \tag{3.3}$$

where the vector $\boldsymbol{\alpha}_{j_1 \cdots j_M} \in \mathbb{R}^q$ is a mode-$(M+1)$ fiber of $\boldsymbol{\alpha}$, and $\epsilon_{j_1 \cdots j_M}$ is a normal random variable with mean zero and is independent of $\mathbf{U}$. Within each class, each entry in the tensor depends on the covariates through a linear regression model. Meanwhile, (3.2) implies that

$$\widehat{\boldsymbol{\alpha}}_{j_1 \cdots j_M} = \left\{\sum_{k=1}^K \sum_{Y^i=k} (\mathbf{U}^i - \overline{\mathbf{U}}_k)(\mathbf{U}^i - \overline{\mathbf{U}}_k)^\intercal\right\}^{-1} \left\{\sum_{k=1}^K \sum_{Y^i=k} (\mathbf{U}^i - \overline{\mathbf{U}}_k)(X^i_{j_1 \cdots j_M} - \overline{X}_{k, j_1 \cdots j_M})\right\}, \tag{3.4}$$

which closely resembles the ordinary least squares estimator except that both $\mathbf{U}^i$ and $\mathbf{X}^i$ are centered within their individual classes. This distinction comes from the fact that our CATCH model assumes the tensor response regression model (2.4) within each class rather than globally.

The connection of $\widehat{\boldsymbol{\alpha}}_{j_1,\ldots,j_M}$ with the least squares estimator suggests an easy extension for estimating $\boldsymbol{\alpha}_{j_1,\ldots,j_M}$ when the covariates are also high-dimensional. If $q \gg n$, we can replace the least squares estimator with the penalized least squares estimator on $(\widetilde{\mathbf{U}}^i, \widetilde{X}_{j_1\cdots j_M})$. A potential application for this method is disease diagnostics based on both the brain images and genetics data.

To estimate the intercept $\boldsymbol{\mu}_k$ based on the tensor response model (2.4), we have

$$\widehat{\boldsymbol{\mu}}_k = \overline{\mathbf{X}}_k - \widehat{\boldsymbol{\alpha}} \bar{\times}_{(M+1)} \overline{\mathbf{U}}_k, \quad k = 1, \ldots, K, \tag{3.5}$$

where $\widehat{\boldsymbol{\alpha}}$ is obtained in (3.2).

3.2 Estimation of $\{\boldsymbol{\Sigma}_m\}_{m=1}^M$

To estimate $\boldsymbol{\Sigma}_m, m = 1, \ldots, M$, we derive the following Lemma 2. A similar result for matrix normal distribution has been presented in Gupta & Nagar (1999).

Lemma 2. *If* $\mathbf{W} \sim TN(0, \boldsymbol{\Omega}_1, \ldots, \boldsymbol{\Omega}_m)$*, then* $\mathrm{E}\{\mathbf{W}_{(j)}\mathbf{W}_{(j)}^\intercal\} = \boldsymbol{\Omega}_j \cdot \prod_{l \neq j} \mathrm{tr}(\boldsymbol{\Omega}_l)$*, where* $\mathbf{W}_{(j)}$ *is the mode-*j *matricization of* $\mathbf{W}$*.*

A direct implication of Lemma 2 is that we can obtain an unbiased estimator for $\boldsymbol{\Sigma}_m$ up to a scale change, based on the fitted residuals from (2.4). From (2.4) and (3.5), we have the fitted residuals for all i, k, such that for an observation with $Y^i = k$,

$$\widehat{\mathbf{E}}^i = \mathbf{X}^i - \widehat{\boldsymbol{\mu}}_k - \widehat{\boldsymbol{\alpha}} \bar{\times}_{(M+1)} \mathbf{U}^i = (\mathbf{X}^i - \overline{\mathbf{X}}_k) - \widehat{\boldsymbol{\alpha}} \bar{\times}_{(M+1)} (\mathbf{U}^i - \overline{\mathbf{U}}_k).$$

Let $\widetilde{\mathbf{S}}_j = (n \prod_{l \neq j}^M p_l)^{-1} \sum_{i=1}^n \widehat{\mathbf{E}}_{(j)}^i (\widehat{\mathbf{E}}_{(j)}^i)^\intercal$. Then by Lemma 2, $\mathrm{E}(\widetilde{\mathbf{S}}_j) = c\boldsymbol{\Sigma}_j$ for some scalar c. To properly scale $\widetilde{S}_j$, we let

$$\widehat{\boldsymbol{\Sigma}}_j = \widetilde{s}_{j,11}^{-1} \widetilde{\mathbf{S}}_j \text{ for } j = 1, \ldots, M-1; \widehat{\boldsymbol{\Sigma}}_M = \frac{\widehat{\mathrm{var}}(X_{1\cdots 1})}{\prod_{j=1}^M \widetilde{s}_{j,11}} \widetilde{\mathbf{S}}_M. \tag{3.6}$$

It is easy to see that $\widehat{\mathbf{\Sigma}}_j$ is always positive semi-definite. But we have the further result concerning the positivity of $\widehat{\mathbf{\Sigma}}_j$ in the following lemma.

Lemma 3. *If*

$$(n-K)\prod_{m\neq j} p_m > p_j, \tag{3.7}$$

then $\widehat{\mathbf{\Sigma}}_j$ is positive definite with probability 1.

It follows that, if (3.7) holds, our penalized optimization introduced later in (3.11) is strictly convex with a probability of 1, which helps with the convergence analysis for our algorithm. It is also worth noting that, the condition in (3.7) is very mild when the dimensions of each mode $p_m, m = 1, \ldots, M$ are roughly comparable. For example, if $p_1 = \ldots = p_M$, the condition in (3.7) is true as long as $n - K > 1$. Meanwhile, if (3.7) does not hold, we could always perturb $\widehat{\mathbf{\Sigma}}_j$ as follows:

$$\widehat{\mathbf{\Sigma}}_j' = \widehat{\mathbf{\Sigma}}_j + \gamma \mathbf{I}_{p_j} \tag{3.8}$$

where $\gamma > 0$ is a small constant. A similar estimator has been considered in the vector case (Ledoit & Wolf 2004) to guarantee positivity of the covariance estimator. Plugging in the estimator in (3.8) results in a strictly convex optimization problem in (3.11) that we will introduce later.

We would like to remark here that many other proposals exist for estimating $\mathbf{\Sigma}_j$ (Dutilleul 1999, Manceur & Dutilleul 2013, Werner et al. 2008). While other estimators can be directly plugged into our CATCH optimization (3.11), they are generally more computationally demanding than our estimator. Because the parameters $\mathbf{\Sigma}_m, m = 1, \ldots, M$ are nuisance to the Bayes' rule (c.f. Proposition 1), the estimation of them is an intermediate step to constructing estimates of $\mathbf{B}_k$. Therefore, we use the estimator in (3.6) for easy computation. Also, we will show theoretically in Section 5 that they lead to consistent estimate of $\mathbf{B}_k$ in high dimensions.

3.3 Penalized estimation of $\{\mathbf{B}_k\}_{k=2}^K$

By Proposition 1, $\mathbf{B}_k = [\![\boldsymbol{\mu}_k - \boldsymbol{\mu}_1; \mathbf{\Sigma}_1^{-1}, \ldots, \mathbf{\Sigma}_M^{-1}]\!] \in \mathbb{R}^{p_1 \times \cdots \times p_M}, k = 2, \ldots, K$. To facilitate sparse estimation, we rewrite $\{\mathbf{B}_k\}_{k=2}^K$ as the solution to an optimization problem as follows.

Lemma 4. *For* $\mathbf{C}_k \in \mathbb{R}^{p_1 \times \cdots \times p_M}, k = 2, \ldots, K$, *define the objective function*

$$\mathcal{L}(\mathbf{C}_2, \ldots, \mathbf{C}_K) = \sum_{k=2}^{K} \{\langle \mathbf{C}_k, [\![\mathbf{C}_k; \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_M]\!] \rangle - 2\langle \mathbf{C}_k, \boldsymbol{\mu}_k - \boldsymbol{\mu}_1 \rangle\}. \tag{3.9}$$

Then $(\mathbf{B}_2, \cdots, \mathbf{B}_K) = \arg\min_{\mathbf{C}_2, \ldots, \mathbf{C}_K} \mathcal{L}(\mathbf{C}_2, \ldots, \mathbf{C}_K)$.

Lemma 4 implies that the unpenalized estimators $\widetilde{\mathbf{B}}_k \equiv [\![\widehat{\boldsymbol{\mu}}_k - \widehat{\boldsymbol{\mu}}_1; \widehat{\boldsymbol{\Sigma}}_1^{-1}, \ldots, \widehat{\boldsymbol{\Sigma}}_M^{-1}]\!]$, $k = 2, \ldots, K$, must be the solution to the following quadratic optimization problem,

$$(\widetilde{\mathbf{B}}_2, \ldots, \widetilde{\mathbf{B}}_K) = \arg \min_{\mathbf{B}_2, \ldots, \mathbf{B}_K} \sum_{k=2}^{K} \{\langle \mathbf{B}_k, [\![\mathbf{B}_k; \widehat{\boldsymbol{\Sigma}}_1, \ldots, \widehat{\boldsymbol{\Sigma}}_M]\!] \rangle - 2\langle \mathbf{B}_k, \widehat{\boldsymbol{\mu}}_k - \widehat{\boldsymbol{\mu}}_1 \rangle\}, \tag{3.10}$$

where the sample estimators $\widehat{\boldsymbol{\mu}}_k, k = 1, \ldots, K, \widehat{\boldsymbol{\Sigma}}_m, m = 1, \ldots, M$ are obtained in previous sections. Therefore, our CATCH estimator $(\widehat{\mathbf{B}}_2, \ldots, \widehat{\mathbf{B}}_K)$ are defined as the minimizer of the following penalized estimation,

$$\min_{\mathbf{B}_2, \ldots, \mathbf{B}_K} \left[\sum_{k=2}^{K} \left(\langle \mathbf{B}_k, [\![\mathbf{B}_k; \widehat{\boldsymbol{\Sigma}}_1, \ldots, \widehat{\boldsymbol{\Sigma}}_M]\!] \rangle - 2\langle \mathbf{B}_k, \widehat{\boldsymbol{\mu}}_k - \widehat{\boldsymbol{\mu}}_1 \rangle \right) + \lambda \sum_{j_1 \ldots j_M} \sqrt{\sum_{k=2}^{K} b_{k, j_1 \cdots j_M}^2} \right], \tag{3.11}$$

where $\lambda > 0$ is a tuning parameter. In comparison to the original quadratic optimization in the population, (3.10), we have add the group LASSO penalty (Yuan & Lin 2006) to the sample optimization because of the group sparsity structure in $\mathbf{B}_2, \ldots, \mathbf{B}_K$ as discussed in Section 2.4. The penalty reduces to the LASSO penalty (Tibshirani 1996) when $K = 2$. Large values of λ encourage group sparsity among $\widehat{\mathbf{B}}_2, \ldots, \widehat{\mathbf{B}}_K$ at matching coordinates, e.g. $\widehat{b}_{2, j_1 \cdots j_M} = \cdots = \widehat{b}_{K, j_1 \cdots j_M} = 0$. As established later in Theorem 2, with an appropriate λ, we have a consistent estimate of $\mathcal{D}$ with a probability tending to 1.

4 Algorithm and Its Convergence

All the estimates except for $\widehat{\mathbf{B}}_k$ in Section 3 can be implemented straightforwardly. Here, we propose an algorithm for estimating $\mathbf{B}_k$ based on (3.11) that scales well with high dimensions.

For convenience, define $\boldsymbol{\beta}_k = \text{vec}(\mathbf{B}_k)$, $\boldsymbol{\nu}_k = \text{vec}(\boldsymbol{\mu}_k)$ and $\widehat{\boldsymbol{\nu}}_k = \text{vec}(\widehat{\boldsymbol{\mu}}_k)$. Rewrite our problem (3.11) with $\boldsymbol{\beta}_k$ as our parameters as

$$(\widehat{\boldsymbol{\beta}}_2, \ldots, \widehat{\boldsymbol{\beta}}_K) = \arg\min_{\boldsymbol{\beta}_2,\ldots,\boldsymbol{\beta}_K} \left[\sum_{k=2}^{K} \{\boldsymbol{\beta}_k^\intercal (\widehat{\boldsymbol{\Sigma}}_M \otimes \cdots \otimes \widehat{\boldsymbol{\Sigma}}_1)\boldsymbol{\beta}_k - 2(\widehat{\boldsymbol{\nu}}_k - \widehat{\boldsymbol{\nu}}_1)^T \boldsymbol{\beta}_k + \lambda \sum_{j=1}^{p} \|\boldsymbol{\beta}_{\cdot j}\|\} \right], \tag{4.1}$$

where $p = \prod_{m=1}^{M} p_m$, $\boldsymbol{\beta} = (\boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_K)^\intercal \in \mathbb{R}^{(K-1)\times p}$, $\boldsymbol{\beta}_{\cdot j} \in \mathbb{R}^{K-1}$ denotes the j-th column vector of $\boldsymbol{\beta}$, and $\|\boldsymbol{\beta}_{\cdot j}\| = (\sum_{k=2}^{K} \beta_{kj}^2)^{\frac{1}{2}}$. After obtaining $\widehat{\boldsymbol{\beta}}_k$, the CATCH estimator $\widehat{\mathbf{B}}_k$ is obtained by mapping $\widehat{\boldsymbol{\beta}}_k$ back to the original tensor.

At first glance, (4.1) is a penalized quadratic problem. In particular, if we ignore the Kronecker product structure and simply let $\widehat{\boldsymbol{\Sigma}} = \widehat{\boldsymbol{\Sigma}}_M \otimes \cdots \otimes \widehat{\boldsymbol{\Sigma}}_1$, then (4.1) reduces to

$$(\widehat{\boldsymbol{\beta}}_2, \ldots, \widehat{\boldsymbol{\beta}}_K) = \arg\min_{\boldsymbol{\beta}_2,\ldots,\boldsymbol{\beta}_K} \left[\sum_{k=2}^{K} \{\boldsymbol{\beta}_k^\intercal \widehat{\boldsymbol{\Sigma}} \boldsymbol{\beta}_k - 2(\widehat{\boldsymbol{\nu}}_k - \widehat{\boldsymbol{\nu}}_1)^T \boldsymbol{\beta}_k + \lambda \sum_{j=1}^{p} \|\boldsymbol{\beta}_{\cdot j}\|\} \right], \tag{4.2}$$

which resembles the objective function of multiclass sparse discriminant analysis in Mai et al. (2017) and can be solved by the algorithm therein when the dimension for $\widehat{\boldsymbol{\Sigma}}$ is not prohibitive. However, for high-dimensional tensors, the dimension of $\widehat{\boldsymbol{\Sigma}}$ is $(\prod_{m=1}^{M} p_m) \times (\prod_{m=1}^{M} p_m)$. Even the storage of such a high-dimensional matrix can be challenging, let alone further operations on it. Therefore, we propose a new algorithm that takes advantage of the Kronecker product structure of $\widehat{\boldsymbol{\Sigma}} = \bigotimes_{m=M}^{1} \widehat{\boldsymbol{\Sigma}}_m$.

Define the operator $(x)_+ = x$ if $x \geq 0$ and $(x)_+ = 0$ if $x < 0$. The mod operator is defined by modulo operation, however, we let $a \bmod b = b$ if the remainder of a modulo b is 0. We also define two sequences of numbers for each j:

$$s_{M+1} = j, s_m = s_{m+1} \bmod \left(\prod_{i=1}^{m-1} p_i \right), \text{ for } m = M, \ldots, 2 \tag{4.3}$$

$$j_1 = s_2 \bmod p_1, j_m = \left\lceil \frac{s_{m+1}}{\prod_{i=1}^{m-1} p_i} \right\rceil, \text{ for } m = 2, \ldots, M \tag{4.4}$$

We need the two sequences $\{s_m\}, \{j_m\}$ for technical reasons. See Lemma D.1 in the Supplementary Materials for more details. Then our algorithm is based on the following lemma.

Algorithm 1 Algorithm for CATCH

1. Input $\widehat{\mathbf{\Sigma}}_m, m = 1, \ldots, M$ and $\widehat{\boldsymbol{\mu}}_k, k = 1, \ldots, K$. Initialize $\widehat{\mathbf{B}}_k = 0$ for all $k = 2, \ldots, K$.
2. For steps $w = 1, 2, \ldots$, do the following until convergence:
   for each element $j = 1, \ldots, p$,
   (a) Update $\mathbf{B}_k^j$ based on current $\widehat{\boldsymbol{\beta}}_{\cdot j'}^{(w-1)}$ for all $j' \neq j$ and compute

$$\widehat{\beta}_{kj}^{(w)} \longleftarrow \frac{(\widehat{\nu}_{kj} - \widehat{\nu}_{k1}) - [\![\mathbf{B}_k^j; \widehat{\mathbf{\Sigma}}_{1,\cdot j_1}^{\intercal}, \cdots, \widehat{\mathbf{\Sigma}}_{M,\cdot j_M}^{\intercal}]\!]}{\prod_{m=1}^{M} \widehat{\sigma}_{m,j_m j_m}} \tag{4.8}$$

   (b) Compute $\widehat{\beta}_{kj}^{(w)} \longleftarrow \widetilde{\beta}_{kj}^{(w-1)} \left(1 - \frac{\lambda}{\sqrt{\sum_{k=2}^{K} (\widetilde{\beta}_{kj}^{(w-1)})^2}}\right)_+$ for $k = 2, \ldots, K$
3. At convergence, output $\widehat{\mathbf{B}}_k$, where $\mathrm{vec}(\widehat{\mathbf{B}}_k) = \widehat{\boldsymbol{\beta}}_k$.

Lemma 5. *For $m = 1, \ldots, M$, define $\widehat{\mathbf{\Sigma}}_{m,\cdot j}$ as the j'th column of $\widehat{\mathbf{\Sigma}}_m$. For each $j = 1, \ldots, p$, the solution to $\boldsymbol{\beta}_{\cdot j}$ to (4.1) given $\boldsymbol{\beta}_{\cdot j'}, j' \neq j$, is the same as*

$$\arg\min_{\boldsymbol{\beta}_{\cdot j}} \sum_{k=2}^{K} \frac{1}{2} (\beta_{kj} - \widetilde{\beta}_{kj})^2 + \frac{\lambda}{\widehat{\sigma}_{jj}} \|\boldsymbol{\beta}_{\cdot j}\|, \tag{4.5}$$

where

$$\widetilde{\beta}_{kj} = \frac{(\widehat{\nu}_{kj} - \widehat{\nu}_{k1}) - [\![\mathbf{B}_k^j; \widehat{\mathbf{\Sigma}}_{1,\cdot j_1}^{\intercal}, \cdots, \widehat{\mathbf{\Sigma}}_{M,\cdot j_M}^{\intercal}]\!]}{\prod_{m=1}^{M} \widehat{\sigma}_{m,j_m j_m}}, \tag{4.6}$$

with $\widehat{\sigma}_{m,j_m j_m}$ being the (j_m, j_m)-th element of $\widehat{\mathbf{\Sigma}}_m$, and $\mathbf{B}_k^j \in \mathbb{R}^{p_1 \times \cdots \times p_M}$ is a tensor such that the elements in $\mathrm{vec}(\mathbf{B}_k^j)_{j'}$ equals $\beta_{kj'}$ for $j' \neq j$ and 0 otherwise. The indices $j_1, \ldots, j_M$ are defined in (4.4)*. Finally, the solution for* (4.5) *is*

$$\widehat{\boldsymbol{\beta}}_{\cdot j} = \widetilde{\boldsymbol{\beta}}_{\cdot j} \left(1 - \frac{\lambda}{\|\widetilde{\boldsymbol{\beta}}_{\cdot j}\|}\right)_+ . \tag{4.7}$$

By Lemma 5, in order to obtain $\widehat{\mathbf{B}}_k, k = 2, \ldots, K$, we only need to solve for $\widetilde{\boldsymbol{\beta}}_{\cdot j}$ and $\widehat{\boldsymbol{\beta}}_{\cdot j}$ iteratively. Algorithm 1 summarizes our algorithm for obtaining the CATCH estimator $\widehat{\mathbf{B}}_k$. To implement our algorithm, in each iteration we only need certain columns of $\widehat{\mathbf{\Sigma}}_1, \ldots, \widehat{\mathbf{\Sigma}}_M$,

because we take into account the Kronecker structure in (4.6). Hence, the space required to implement our algorithm is of the order $O(\sum_{m=1} p_m^2 + K\prod_{m=1}^M p_m)$, where the storage cost $K\prod_{m=1}^M p_m$ comes from the means $\hat{\boldsymbol{\mu}}_k$. In contrast, if we ignore the Kronecker structure and solve (4.2), we will have to compute $\widehat{\boldsymbol{\Sigma}}$ beforehand, which requires the space at the order of $O(\prod_{m=1}^M p_m^2 + K\prod_{m=1}^M p_m)$. By taking advantage of the Kronecker product structure, we gain considerable saving in storage. Our algorithm scales much better to high-dimensional tensor data.

We also have the following result for per-iteration computational complexity.

Lemma 6. *The computational cost for updating $\widehat{\beta}_{kj}^{(t)}, j = 1, \ldots, p$ is $O(MKd_tp)$, where d_t is the number of important tensor entries in the discriminative set $\widehat{\mathcal{D}}$ at iteration t, $t = 1, 2, \ldots$.*

Finally, we present the convergence result for our blockwise coordinate descent algorithm. For ease of presentation, we assume that (3.7) holds and hence our optimization problem is strictly convex with a probability of 1. If (3.7) does not hold, we can always replace the covariance estimates by those in (3.8) to achieve similar convergence results.

Theorem 1. *If* (3.7) *holds for $m = 1, \ldots, M$, then our blockwise coordinate descent algorithm (Algorithm 1) converges to the global minimizer of* (4.1) *with a probability of 1.*

Theorem 1 shows that there is no gap between the output of our algorithm and the global minimizer of (4.1). This fact facilitates the theoretical studies of CATCH, as will be presented in the next section.

5 Theory

In this section, we study the statistical properties of CATCH. Theorem 2 establishes the estimation and variable selection consistency, while Theorem 3 establishes the prediction consistency.

We first introduce our notation. For a matrix $\mathbf{V} \in \mathbb{R}^{q_1 \times q_2}$, $\|\mathbf{V}\|_\infty = \max_i \sum_{j=1}^{q_2} |v_{ij}|$, $\|\mathbf{V}\|_1 = \max_j \sum_{i=1}^{q_1} |v_{ij}|$. For an m-way tensor $\mathbf{W} \in \mathbb{R}^{q_1 \times \cdots \times q_m}$, $\|\mathbf{W}\|_{\max} = \max_{j_1, \cdots, j_m} |w_{j_1 \cdots j_m}|$. Throughout this section C denotes a generic positive constant that could vary from line to

line. We use $\mathbf{\Sigma} = \bigotimes_{m=M}^{1} \mathbf{\Sigma}_m$ to simplify the presentation of the theoretical results, although in practice we never directly use the estimate of $\mathbf{\Sigma}$ (c.f. Algorithm 1). Recall that $p = \prod_{m=1}^{M} p_m, p_{-m} = \prod_{j \neq m} p_j$, $d = |\mathcal{D}|$; and we also let

$$
\begin{aligned}
b_{\max} &= \max_{k, j_1 \cdots j_M} |b_{k, j_1 \cdots j_M}|, && (5.1)\\
b_{\min} &= \min_{(k, j_1 \cdots j_M): b_{k, j_1 \cdots j_M} \neq 0} |b_{k, j_1 \cdots j_M}|, && (5.2)\\
\varphi &= \max\{\|\mathbf{\Sigma}_{\mathcal{D}^C, \mathcal{D}}\|_\infty, \|\mathbf{\Sigma}^{-1}_{\mathcal{D}, \mathcal{D}}\|_\infty\}, && (5.3)\\
\Delta &= \max\{\| \left(\mathrm{vec}(\boldsymbol{\mu}_1), \cdots, \mathrm{vec}(\boldsymbol{\mu}_K)\right) \|_1, \| \left(\mathrm{vec}(\mathbf{B}_2), \cdots, \mathrm{vec}(\mathbf{B}_K)\right) \|_1\}. && (5.4)
\end{aligned}
$$

For simplicity, we make a few assumptions about the parameters, but all the assumptions in this paragraph can be relaxed, at the price of lengthier proofs. The number of classes, K, the number of covariates, q, and the order of the tensor, M, are all assumed to be fixed. We assume that $\|\mathbf{\Sigma}_j^{1/2}\|_1$, $\|\boldsymbol{\phi}_k\|_{\max}$ and $\|\boldsymbol{\mu}_k\|_{\max}$ are bounded above uniformly with respect to p. We further assume that the diagonal elements of $\mathbf{\Sigma}_j$, $j = 1, \ldots, M$, are all ones.

The following technical conditions will be used in the theorems.

(C1) $\max_{j \in \mathcal{D}^c} \left\{ \sum_{k=2}^{K} (\mathbf{\Sigma}_{j\mathcal{D}} \mathbf{\Sigma}^{-1}_{\mathcal{D}\mathcal{D}} \mathbf{t}_{k\mathcal{D}})^2 \right\}^{1/2} = \kappa < 1$, where $\mathbf{t}_{k\mathcal{D}}$ is defined as the sub-gradient of the group lasso penalty term in the objective function (4.1) with respect to $\boldsymbol{\beta}_{k\mathcal{D}}$.

(C2) There exists $c_1 > 0$ such that $\pi_k \geq \dfrac{c_1}{K}$ for $k = 1, \ldots, K$.

(C3) The largest eigenvalues of $\mathbf{\Sigma}_m, m = 1, \ldots, M$, are uniformly bounded above by a constant $C_2 > 0$.

(C4) $\min_{j,k}\{(\mathrm{vec}(\boldsymbol{\mu}_k - \boldsymbol{\mu}_j))^\intercal \mathbf{\Sigma}^{-1} (\mathrm{vec}(\boldsymbol{\mu}_k - \boldsymbol{\mu}_j))\}$ is bounded away from 0.

(C5) $\left\{ \dfrac{d^2(\log d + \sum_{m=1}^{M} \log p_m)}{n} \right\}^{1/2} = o(b_{\min})$ as $n \to \infty$.

Condition (C1) is a technical assumption similar to the standard condition for group lasso in penalized regression models (Bach 2008). Condition (C2) implies that the classes are reasonably balanced and that as the sample size increases, each class will have a reasonably large sample size. Condition (C3) mimics a popular assumption in high-dimensional data analysis.

For example, in Cai & Liu (2011) where they considered sparse LDA, it was assumed that the largest eigenvalue of the covariance matrix is bounded above. Condition (C4) guarantees that the classes are well separated to allow for accurate prediction and error rate consistency. Condition (C5) imposes a constraint on the dimensions and the signal strength. If $b_{\min} = O(1)$ and $d = O(n^{\xi})$ for $0 < \xi < 1/2$, then we can allow $\log p_m = o(n^{1-2\xi})$. Hence, we can allow the dimension of each mode of the tensor dimension to grow at an exponential rate of the sample size. Meanwhile, we can also allow $b_{\min}$ to decay at a rate determined by the dimensionality.

Theorem 2 (Estimation and Variable Selection Consistency)**.** *Under Conditions (C1)–(C3), there exists a generic constant ψ such that, if $0 < \lambda < \min\{\frac{b_{\min}}{8\varphi}, \psi(1-\kappa), 1\}$, then with a probability greater than*

$$1 - 2K\sum_{m=1}^{M} p_m^2 \exp(-\frac{Cnp_{-m}\lambda^2}{d^2}) - CKq^2 p\exp\{-\frac{Cn\lambda^2}{d^2q^2}\} - 2Kp\exp(-Cn\frac{\lambda^2}{d^2}),$$

we have that $\widehat{\mathcal{D}} = \mathcal{D}$ and $\|\mathrm{vec}(\widehat{\mathbf{B}}_k) - \mathrm{vec}(\mathbf{B}_k)\|_\infty \le 4\varphi\lambda$. If we further assume Condition (C5), and that $\{\frac{d^2(\log d + \sum_{m=1}^{M}\log p_m)}{n}\}^{1/2} \ll \lambda \ll b_{\min}$, $\lambda \to 0$, then we have the following statements with a probability tending to 1,

$$\widehat{\mathcal{D}} = \mathcal{D}, \quad \|\mathrm{vec}(\widehat{\mathbf{B}}_k) - \mathrm{vec}(\mathbf{B}_k)\|_\infty \to 0, \quad k = 1, \ldots, K.$$

Theorem 2 implies that, under Conditions (C1)–(C3), (C5) and (C6), we can correctly identify the important features and accurately estimate the discriminant effects with a probability tending to 1. This supports the application of CATCH in high-dimensional data.

We also remark that the proofs in Theorem 2 are much more involved than those in the sparse LDA literature (Cai & Liu 2011, Fan et al. 2012, Mai et al. 2012, 2017) for two reasons. First, we have tensor normal data and we estimate the covariance by a Kronecker product of marginal sample covariances. Consequently, the existing results for sample covariance do not apply here. A relevant paper is Zhou (2014) where the author presented large deviation results for matrix normal distribution. But in the current paper we show large deviation results for tensor normal distribution, without relying on any of the results in Zhou (2014), which can

be of independent interest. Second, we have two layers of hierarchical structures. When we estimate the parameters for tensor data, we have to resort to the pseudo data $\mathbf{X} - \widehat{\boldsymbol{\alpha}} \bar{\times}_{(M+1)} \mathbf{U}$, where $\widehat{\boldsymbol{\alpha}}$ introduces additional noise. There is no such issue in sparse LDA. Fortunately, in our careful theoretical studies, we observe that $\widehat{\boldsymbol{\alpha}}$ does not inflate the order of the estimation error. In other words, even if we used $\mathbf{X} - \boldsymbol{\alpha} \bar{\times}_{(M+1)} \mathbf{U}$ to construct $\widehat{\mathbf{B}}_k$, we would not have a smaller order of estimation error. This assures that although we have to estimate $\boldsymbol{\alpha}$, it has very little effect on our final estimation. Such results also support our unpenalized estimation procedure for $\boldsymbol{\alpha}$.

In what follows, we further present results concerning the classification error rates. Since we observe that $\widehat{\boldsymbol{\alpha}}$ is generally a good surrogate for $\boldsymbol{\alpha}$, we consider the simplified case where we only have tensor data but not covariates. In the rest of this section, we assume the model in (2.3). For a new observation $\{\mathbf{X}^{\text{new}}, Y^{\text{new}}\}$ not involved in fitting the classifier, define the classification error rate of our CATCH estimator and that of the Bayes rule as follows:

$$R_n = \Pr\left(\widehat{Y}(\mathbf{X}^{\text{new}} \mid \widehat{\mathbf{B}}_k, \widehat{\pi}_k, \widehat{\boldsymbol{\mu}}_k) \neq Y^{\text{new}}\right), \tag{5.5}$$

$$R = \Pr\left(\widehat{Y}(\mathbf{X}^{\text{new}} \mid \mathbf{B}_k, \pi_k, \boldsymbol{\mu}_k) \neq Y^{\text{new}}\right). \tag{5.6}$$

Clearly, we hope R_n to be as close to R as possible. Indeed, in the following theorem we show that R_n converges to R with an overwhelming probability.

Theorem 3 (Optimal Prediction and Bayes' Rule Consistency)**.** *Under Conditions (C1)–(C4), there exists a generic constant $\psi_1 > 0$ such that, if $0 < \lambda < \min\{\frac{b_{\min}}{8\varphi}, \psi_1(1-\kappa), 1\}$, then with a probability greater than*

$$1 - 2K\sum_{m=1} p_m^2 \exp(-C\frac{np_{-m}\lambda^2}{d^2}) - CKq^2p\exp\{-\frac{Cn\lambda^2}{d^2q^2}\} - 2Kp\exp(-Cn\frac{\lambda^2}{d^2}), \tag{5.7}$$

we have

$$|R_n - R| \leq C\lambda^{1/3}.$$

If we further assume Condition (C5), $\frac{d^2(\log d + \sum_{m=1}^M \log p_m)}{n} \ll \lambda \ll b_{\min}$ and $\lambda \to 0$, then with a probability tending to 1,

$$R_n \to R. \tag{5.8}$$

According to Theorem 3, CATCH can asymptotically achieve the best classification accuracy. Therefore, CATCH is a powerful prediction tool as well.

6 Simulations

In Section 6.1, we present binary classification ($K = 2$) simulations without covariates so that we can compare with most existing competitors. In Section 6.2, we present multi-class ($K = 3$ or 4) simulations with both tensor predictor and covariates.

We include various popular and state-of-the-art classification methods as competitors. From the machine learning and high-dimensional statistics literature, we include ℓ_1-penalized Fisher's discriminant analysis (ℓ_1-FDA; Witten & Tibshirani 2011), sparse optimal scoring (SOS; Clemmensen et al. 2011), ℓ_1-penalized generalized linear models (logistic regression for binary, and multinomial logistic regression for multi-class problem) (ℓ_1-GLM; Friedman et al. 2010, Goeman et al. 2012), random forests (RF Breiman 2001, Liaw & Wiener 2014), and ℓ_1-penalized support vector machine (ℓ_1-SVM; Cortes & Vapnik 1995, Dimitriadou et al. 2009, Bradley & Mangasarian 1998, Fung & Mangasarian 2004, Becker et al. 2009). All these methods are designed for vector predictors. From matrix and tensor discriminant analysis literature, we include matrix discriminant analysis (MDA) and penalized MDA (PMDA) (Zhong & Suslick 2015), constrained multilinear discriminant analysis (CMDA) and directly generalized tensor discriminant analysis (DGTDA) (Li & Schonfeld 2014), tensor GLM based on CP decomposition (CP-GLM; Zhou et al. 2013), and sparse tensor discriminant analysis (STDA Lai et al. 2013). CP-GLM has the best performance in our simulations with the rank-3 decomposition. Thus we use the rank of 3 in CP-GLM. Most of the other methods mentioned in Section 1 are not included because they are either inapplicable for multi-class data or too computationally demanding. Furthermore, we also report the error rates of the Bayes' rule as a baseline, as well as the oracle vector classifier and oracle tensor classifier that use the oracle information of important predictors. More details about the implementation can be found in the Supplementary Materials.

In all simulations, we generated the training data such that there are $n_k = 75$ observations

within each class, unless otherwise specified. We further generated an independent validation set of the same size as the training set. The methods were evaluated on a testing set with 10,000 observations. All the simulation results are based on 100 replicates of the above procedure. We also compare the variable selection results of the methods quantified by the true positive rate (TPR) and the false positive rate (FPR), defined as:

$$\text{TPR} = \frac{|\,\widehat{\mathcal{D}} \bigcap \mathcal{D}\,|}{|\,\mathcal{D}\,|},\ \text{FPR} = \frac{|\,\widehat{\mathcal{D}} \bigcap \mathcal{D}^c\,|}{|\,\mathcal{D}^c\,|}. \tag{6.1}$$

When introducing the models, we use the following shorthand notation. For a matrix $\mathbf{\Omega}$, $\mathbf{\Omega} = AR(\rho)$ means $\Omega_{ij} = \rho^{|i-j|}$ for all i, j, while $\mathbf{\Omega} = CS(\rho)$ means $\Omega_{ij} = \rho$ for $i \neq j$ and $\Omega_{ii} = 1$ for all i. For a tensor $\mathbf{C}$ and a number c, $\mathbf{C} = c$ means that all the elements in $\mathbf{C}$ are equal to c.

6.1 Models without covariates

We first compare CATCH with existing methods on six models with only the tensor predictor but no covariates. The first three models (M1–M3) involve matrix predictors of size 64×64 from $K = 4$ classes; the next three models (T1–T3) involve 3-way tensor predictors of size $30 \times 36 \times 30$ from $K = 3$ classes; the last model (T3i) is a special case of T3 with imbalanced (unequal) class sizes. In each model setting, we specify $\mathbf{\Sigma}_m, m = 1, \ldots, M$ and $\mathbf{B}_k, k = 2, \ldots, K$ and set $\boldsymbol{\mu}_1 = 0$ and $\boldsymbol{\mu}_k = [\![\mathbf{B}_k; \mathbf{\Sigma}_1, \ldots, \mathbf{\Sigma}_M]\!]$. Then we generate data from TDA model (2.3).

We let $\mathcal{D}_1$ and $\mathcal{D}_2$ be subsets of $\mathcal{D} = \mathcal{D}_1 \bigcup \mathcal{D}_2$ such that $\mathbf{B}_{k,\mathcal{D}^c} = 0$ for each k. Specifically, $\mathcal{D}_1 = \{(i,j) : i = 1, 2, 11, 12 \text{ and } j = 1, 2\}$, $\mathcal{D}_2 = \{(i,j) : i = 1, 2, 11, 12 \text{ and } j = 11, 12\}$ and $\mathcal{D} = \mathcal{D}_1 \bigcup \mathcal{D}_2$ for models (M1)–(M3); and $\mathcal{D}_1 = \{(i,j,l) : i = 1, 2, 11, 12, j = 1, 11 \text{ and } l = 1\}$ and $\mathcal{D}_2 = \{(i,j) : i = 1, 2, 11, 12, j = 1, 11 \text{ and } l = 11\}$ for models (T1)–(T3i).

Model (M1) (Independent predictors): $\mathbf{\Sigma}_1 = \mathbf{\Sigma}_2 = \mathbf{I}_{64}$, $\mathbf{B}_{2,\mathcal{D}} = 0.6$, $\mathbf{B}_{3,\mathcal{D}_1} = 0.6$, $\mathbf{B}_{3,\mathcal{D}_2} = 1.8$, $\mathbf{B}_{4,\mathcal{D}_1} = -0.6$ and $\mathbf{B}_{4,\mathcal{D}_2} = 0.6$.

Model (M2) (Independent rows): $\mathbf{\Sigma}_1 = \mathbf{I}_{64}, \mathbf{\Sigma}_2 = AR(0.7)$, $\mathbf{B}_{2,\mathcal{D}} = 0.4$, $\mathbf{B}_{3,\mathcal{D}_1} = 0.4$ and $\mathbf{B}_{3,\mathcal{D}_2} = 1.2$, $\mathbf{B}_{4,\mathcal{D}_1} = -0.4$ and $\mathbf{B}_{4,\mathcal{D}_2} = 0.4$.

Model (M3) (Dependent predictors): $\mathbf{\Sigma}_1 = CS(0.3), \mathbf{\Sigma}_2 = AR(0.7)$, $\mathbf{B}_{2,\mathcal{D}} = 0.4$, $\mathbf{B}_{3,\mathcal{D}_1} = 0.4$ and $\mathbf{B}_{3,\mathcal{D}_2} = 1.2$, $\mathbf{B}_{4,\mathcal{D}_1} = -0.4$ and $\mathbf{B}_{4,\mathcal{D}_2} = 0.4$.

Model (T1) (Independent predictors): $\mathbf{\Sigma}_m, m = 1, 2, 3$ are all identity matrices, $\mathbf{B}_{2,\mathcal{D}} = 0.6$, $\mathbf{B}_{3,\mathcal{D}_1} = 0.6$ and $\mathbf{B}_{3,\mathcal{D}_2} = 1.5$.

Model (T2) (Independent mode-2 fibers): $\mathbf{\Sigma}_1 = AR(0.7), \mathbf{\Sigma}_2 = \mathbf{I}_{36}, \mathbf{\Sigma}_3 = CS(0.3)$, $\mathbf{B}_{2,\mathcal{D}} = 0.4$, $\mathbf{B}_{3,\mathcal{D}_1} = 0.4$ and $\mathbf{B}_{3,\mathcal{D}_2} = 1$.

Model (T3) (Dependent predictors): $\mathbf{\Sigma}_1 = AR(0.7), \mathbf{\Sigma}_2 = CS(0.3), \mathbf{\Sigma}_3 = CS(0.3)$, $\mathbf{B}_{2,\mathcal{D}} = 0.4$, $\mathbf{B}_{3,\mathcal{D}_1} = 0.4$ and $\mathbf{B}_{3,\mathcal{D}_2} = 1$.

Model (T3i) (Imbalanced classes): All the parameters are the same as Model (T3), except for $n_1 = n_2 = 40, n_3 = 200$.

Error rate(%)	M1	M2	M3	T1	T2	T3	T3i	S.E.$\leq$
Bayes	14.29	19.24	8.84	14.48	16.17	12.18	8.10	(0.04)
Tensor Oracle	15.71	20.97	9.76	16.28	17.92	13.42	9.21	(0.13)
Vector Oracle	16.20	21.51	10.22	16.56	18.64	14.14	9.48	(0.18)
CATCH	**17.44**	**20.09**	**9.88**	**19.69**	**19.05**	**13.83**	**9.78**	**(0.17)**
STDA	74.22	72.69	47.04	66.39	65.00	57.15	35.10	(0.87)
DGTDA	75.06	74.97	52.04	66.74	66.68	60.84	40.67	(0.20)
CMDA	36.27	40.06	19.81	44.6	37.28	27.57	21.75	(0.26)
MDA	38.34	44.35	29.84	NA	NA	NA	NA	(0.29)
PMDA	27.61	33.55	19.26	NA	NA	NA	NA	(0.46)
ℓ_1-FDA	18.33	23.98	14.64	30.56	25.12	30.82	25.57	(0.33*)
SOS	19.21	24.40	11.82	25.88	26.07	20.33	13.09	(0.22)
ℓ_1-GLM	18.85	22.98	10.85	25.41	22.65	17.22	13.62	(0.16)
RF	53.21	43.75	16.79	NA	NA	NA	NA	(0.17)

Table 1: Prediction comparison. The means and the maximum standard errors (in parentheses) of classification error rates are reported. MDA and PMDA are not applicable for 3-way tensor data; RF can not handle the high-dimensionality in the tensor models; ℓ_1-SVM and CP-GLM are designed for binary (or pairwise) classifications thus are included only in later binary classification problems.
*The maximum S.E. of ℓ_1-FDA is calculated across all the models except for Model (T1). The S.E. of ℓ_1-FDA on Model (T1) is 1.14.

		M1	M2	M3	T1	T2	T3	T3i	S.E.$\leq$
CATCH	TPR	99.06	93.94	92.13	83.13	82.13	86.56	71.38	(1.69)
	FPR	0.16	0.12	0.01	0.05	0.03	0.03	0.01	(0.02)
ℓ_1-FDA	TPR	51.75	60.00	52.38	54.38	91.19	93.63	100	(1.82)
	FPR	0.11	0.29	6.88	6.82	1.29	29.28	19.52	(1.02)
SOS	TPR	99.31	91.19	82.31	60.19	57.69	60.5	55.94	(1.31)
	FPR	0.38	0.35	0.39	0.05	0.05	0.06	0.08	(0.02)
ℓ_1-GLM	TPR	99.44	92.25	85.00	64.69	65.06	65.69	61.81	(1.01)
	FPR	0.35	0.23	0.23	0.06	0.03	0.03	0.17	(0.05)

Table 2: Variable selection comparison. TPR and FPR are defined in (6.1).

The error rates of all methods are reported in Table 1. CATCH significantly outperforms all the other methods, and closely resembles the oracle classifiers across all the models. This supports the application of CATCH. In what follows we discuss the comparison in more details.

First, the comparison among CATCH, MDA and PMDA suggests that it is critical to utilize the sparsity assumption. MDA does not perform variable selection, while PMDA performs variable selection on the rows but not the columns. On the other hand, CATCH can achieve elementwise sparsity. Hence, PMDA significantly improves MDA, while CATCH outperforms PMDA. We can see this point more clearly by noting that the two oracle methods are very close to the Bayes rule, since they have oracle information on the important predictors. By performing variable selection CATCH has error rates close to the oracle methods. Moreover, CATCH significantly outperforms other tensor discriminant analysis methods (STDA, DGTDA and CMDA) that are more or less based on low dimensional (sparse) projections.

Second, although ℓ_1-FDA, ℓ_1-GLM and SOS aggressively take advantage of the sparsity assumption, our CATCH estimator is still more accurate. The margin becomes larger for higher order tensors in Models (T1)–(T3i), and correlated predictors in Models (M3), (T3) and (T3i). This is because CATCH honors the tensor structure and preserves more information. Because ℓ_1-FDA, ℓ_1-GLM and SOS require vectorizing data, they are less efficient. The importance of honoring the tensor structure can be further confirmed by examining the two oracle classifiers. The oracle tensor classifier takes into account the tensor structure, and uniformly outperforms the oracle vector classifier.

We also investigate the variable selection results as summarized in Table 2. We do not include other tensor methods in this comparison, because they do not perform variable selection as aggressively as the reported ones. SOS and ℓ_1-GLM tend to under-select, while ℓ_1-FDA tends to over-select. CATCH usually selects the majority of the important features, with very few false positives. Such results explain why CATCH performs similarly to the oracle methods. It also supports our theoretical results on the variable selection consistency of CATCH.

Finally, for the imbalanced classes Model (T3i), the classification is easier than the balanced model (T3), as shown by the lower Bayes error. Hence, all methods perform better in prediction, but CATCH is still the closest to the Bayes rule. On the other hand, the variable selection becomes more challenging. But CATCH is almost as accurate as the oracle estimator when only 71% of the true variables are selected with nearly zero false positives. Overall, CATCH is not sensitive to imbalanced classes.

6.2 Models with covariates

In this section, we consider the CATCH model (2.1) and (2.2) with tensor predictors of size $30\times36\times30$, covariates $\mathbf{U}\in\mathbb{R}^2$, and the binary classification setting, $K=2$. In each model, we specify the CATCH model parameters $\boldsymbol{\phi}_k, \boldsymbol{\mu}_k, \mathbf{B}_k, \boldsymbol{\Psi}, \boldsymbol{\Sigma}_m$ and $\boldsymbol{\alpha}=[\![\boldsymbol{\alpha}^*;\boldsymbol{\Sigma}_1^{1/2},\ldots,\boldsymbol{\Sigma}_M^{1/2},\mathbf{I}_q]\!]$. We let $\mathcal{D}=\{(i,j,l): i=1,2,11,12, j=1,11 \text{ and } l=1,11\}$, $\boldsymbol{\Psi}=\mathbf{I}_2$ and two possible values of $\boldsymbol{\alpha}^*$ by $\boldsymbol{\alpha}_1^*$ and $\boldsymbol{\alpha}_2^*$, where $\alpha^*_{1,ijst}=0.5$ for $i,j,s=1,\ldots,5$ and $t=1$; $\alpha^*_{2,ijst}=1$ for $i,j,s=1,\ldots,15$ and $t=1$; for all other (k,i,j,s,t), $\alpha^*_{k,ijst}=0$.

Model (C1) (Covariates and tensor with independent predictors): $\boldsymbol{\Sigma}_m, m=1,2,3$ are identity matrices, $\mathbf{B}_{2,\mathcal{D}}=0.8$ and $\mathbf{B}_{2,\mathcal{D}^c}=0$, $\boldsymbol{\phi}_2=0.3$, $\boldsymbol{\alpha}^*=\boldsymbol{\alpha}_2^*$.

Model (C2) (Covariates and tensor with independent mode-2 fibers): $\boldsymbol{\Sigma}_1=AR(0.7), \boldsymbol{\Sigma}_2=\mathbf{I}_{36}, \boldsymbol{\Sigma}_3=CS(0.3)$, $\mathbf{B}_{2,\mathcal{D}}=0.4$ and $\mathbf{B}_{2,\mathcal{D}^c}=0$, $\boldsymbol{\phi}_2=0.3$, $\boldsymbol{\alpha}^*=\boldsymbol{\alpha}_1^*$.

Model (C3) (Covariates and tensor with dependent predictors): $\boldsymbol{\Sigma}_1=AR(0.7), \boldsymbol{\Sigma}_2=CS(0.3), \boldsymbol{\Sigma}_3=CS(0.3)$, $\mathbf{B}_{2,\mathcal{D}}=0.4$ and $\mathbf{B}_{2,\mathcal{D}^c}=0$, $\boldsymbol{\phi}_2=0.3$, $\boldsymbol{\alpha}^*=\boldsymbol{\alpha}_1^*$.

Model (C3a) (Independent covariates and tensor): All the parameters are the same as in (C3) except $\boldsymbol{\phi}_2=1$ and $\boldsymbol{\alpha}^*=\mathbf{0}$.

Model (C3b) (Non-discriminantive covariates): All the parameters are the same as in (C3) except $\boldsymbol{\phi}_2 = \mathbf{0}$.

Model (C3i) (Imbalanced classes): All the parameters are the same as in (C3), but we let $n_1 = 40$ and $n_2 = 200$.

To investigate the influence of the covariates information on each method, we consider two tasks for each method: classification based on $\mathbf{X}$ alone and that based on both $\mathbf{X}$ and $\mathbf{U}$. In the presence of $\mathbf{U}$, only CATCH and CP-GLM (Zhou et al. 2013) can naturally include the covariates in their models. Thus, for other tensor methods (STDA, DGTDA and CMDA), we stack covariates with downsized tensor when these methods apply the nearest neighbor classifier; for vector methods, we stack $\mathbf{U}$ and $\mathrm{vec}(\mathbf{X})$ as a single vector. As suggested by a referee, we also include two other methods, SOS weighted and ℓ_1-GLM weighted that first fit models based on $\mathbf{X}$ and $\mathbf{U}$ separately and then combine the information with a weighted vote. In SOS weighted, we apply LDA on $(Y, \mathbf{U})$ and SOS on $(Y, \mathrm{vec}(\mathbf{X}))$ to obtain $\widehat{\boldsymbol{\alpha}}^{\mathsf{T}}\mathbf{U}$ and $\widehat{\boldsymbol{\beta}}^{\mathsf{T}}\mathrm{vec}(\mathbf{X})$, respectively. Then we find a weighted vote of them by applying LDA to Y with $\{\widehat{\boldsymbol{\alpha}}^{\mathsf{T}}\mathbf{U}, \widehat{\boldsymbol{\beta}}^{\mathsf{T}}\mathrm{vec}(\mathbf{X})\}$ as predictors. Similarly, in ℓ_1-GLM weighted, $\mathbf{U}$ and $\mathrm{vec}(\mathbf{X})$ are first separately modeled by GLM and ℓ_1-GLM, and then combined with GLM.

The results are listed in Table 3. CATCH again uniformly outperforms all the competitors. In the imbalanced case Model (C3i), the Bayes error is lower than that in Model (C3). Consequently, all the methods have improved accuracy, but CATCH remains the best classifier. Moreover, when the covariates are included, the performance of CATCH can be significantly improved. On the other hand, including the covariates in other methods in general does not improve classification performance, except for model (C3a) where the covariate and the tensor are independent within each class. This suggests that adjusting for covariates improves the clasification, which may not be achieved by the cruder approaches that ignore the dependence between the covariates and the tensor. We further study the role of covariates in models (C3a, b) as follows.

Model (C3a) is a special case where $\mathbf{U}$ and $\mathbf{X}$ are independent within each class since $\boldsymbol{\alpha} = 0$. Both the covariates and the tensor predictors contribute to the classification, but LDA

and CATCH without covariates are less accurate than CATCH, because they do not utilize both types of predictors. Naively combining $\mathbf{X}$ (or $\mathrm{vec}(\mathbf{X})$) and $\mathbf{U}$ under this scenario can actually improve many methods (CP-GLM, CMDA, SOS, ℓ_1-GLM and ℓ_1-SVM) significantly, but CATCH is still superior to them.

Model (C3b) is another special case where covariates affect the tensor predictors but do not contribute to the classification themselves since $\boldsymbol{\phi}_1 = \boldsymbol{\phi}_2 = 0$. It can be seen that LDA has an error rate around 50%, as the covariates do not have any power in the classification. Since the tensor predictors are very informative, CATCH without covariates already achieves a very low error rate comparing to other methods. Still, when we adjust for the covariates in CATCH, we have a significant improvement. This reinforces our point that even when covariates are not important themselves, they should still be included for further analysis. Without adjusting for the tensor regression relationship between the tensor and the covariates, all other methods cannot effectively utilize the additional information from $\mathbf{U}$ and thus fail to improve with the inclusion of the covariates.

Finally, the variable selection results also show that CATCH outperforms all the competitors and the inclusion of the covariates leads to better variable selection. These results can be found in the Supplementary Materials.

To demonstrate the applicability of CATCH in high-dimensional data, we further consider variants of models (C1)–(C3) where we increase the tensor dimensions to $80 \times 80 \times 80$ ($512,000$ voxels in total). Many methods become practically inapplicable because of the prohibitive computational costs. Therefore, we only compare CATCH with ℓ_1-GLM, ℓ_1-FDA, CP-GLM and DGTDA. The classification errors and the variable selection results can be found in the Supplementary Materials. CATCH continues to achieve better accuracy and variable selection than the competitors.

We also compare the computational costs for CATCH and the competitors. Because CATCH contains two steps, i.e, adjusting for covariates and penalized estimation of the coefficients $\mathbf{B}_k$, we report the computation time for these two steps along with the total of them. Although CATCH can produce the whole solution path simultaneously, many other methods cannot.

Error rate(%)		C1	C2	C3	C3a	C3b	C3i	S.E. $\leq$
Bayes		5.33	10.97	8.15	6.08	8.39	5.45	(0.03)
LDA	$\mathbf{U}$	42.62	42.21	42.45	24.34	50.02	16.72	(0.18)
CATCH	$\mathbf{X}$	31.03	21.30	16.7	10.78	14.76	10.58	(0.59)
	$\mathbf{X}, \mathbf{U}$	**11.12**	**16.67**	**11.24**	**8.33**	**11.28**	**7.36**	**(0.22)**
CP-GLM	$\mathbf{X}$	40.54	27.45	17.93	16.15	18.02	10.75	(1.40)
	$\mathbf{X}, \mathbf{U}$	39.40	31.16	19.65	13.82	19.12	10.59	(1.30)
STDA	$\mathbf{X}$	48.31	46.17	44.69	41.45	46.42	23.54	(0.38)
	$\mathbf{X}, \mathbf{U}$	48.2	46.08	44.65	40.11	46.54	23.62	(0.34)
DGTDA	$\mathbf{X}$	49.7	49.42	44.99	44.08	46.84	26.01	(0.16)
	$\mathbf{X}, \mathbf{U}$	49.55	49.68	45.88	45.65	47.68	26.73	(0.17)
CMDA	$\mathbf{X}$	39.70	34.60	27.86	25.59	29.16	16.03	(0.32)
	$\mathbf{X}, \mathbf{U}$	39.70	34.45	27.27	22.5	28.73	15.81	(0.32)
ℓ_1-FDA	$\mathrm{vec}(\mathbf{X})$	43.36	30.29	31.72	20.73	30.45	27.32	(0.65)
	$\mathrm{vec}(\mathbf{X}), \mathbf{U}$	43.37	30.31	31.71	20.56	33.36	27.30	(0.65)
SOS	$\mathrm{vec}(\mathbf{X})$	34.30	17.46	14.39	11.87	15.56	9.29	(0.31)
	$\mathrm{vec}(\mathbf{X}), \mathbf{U}$	34.30	17.45	14.39	9.70	15.56	9.29	(0.31)
	Weighted	33.26	16.97	13.78	9.08	14.61	8.91	(0.29)
ℓ_1-GLM	$\mathrm{vec}(\mathbf{X})$	34.01	17.57	14.58	12.31	15.87	10.43	(0.30)
	$\mathrm{vec}(\mathbf{X}, \mathbf{U})$	34.07	17.8	14.95	9.78	15.90	10.43	(0.35)
	Weighted	33.11	17.8	14.55	9.74	15.4	9.51	(0.35)
ℓ_1-SVM	$\mathrm{vec}(\mathbf{X})$	25.53	25.94	19.05	16.85	19.00	10.14	(0.20)
	$\mathrm{vec}(\mathbf{X}, \mathbf{U})$	23.71	22.91	19.03	15.3	18.21	10.95	(0.10)

Table 3: Prediction comparison. The means and the maximum standard errors (in parentheses) of classification error rates based on 100 replicates are reported. Same as models (T1)–(T3i) in Table 1, MDA, PMDA and RF are excluded because MDA and PMDA are not applicable for 3-way tensors and RF can not handle the high-dimensionality.

Therefore, we only compare the methods for pre-chosen tuning parameters that yield the highest accuracy for each method, respectively. For discriminant analysis methods, we need to first find the means and the covariances. This step can be sped up easily by parallel computing and is hence excluded when we calculate the computation time. For the same reason, we did not include the computational time for the standardization step in ℓ_1-FDA that centers and standardizes each variable. The average computation time of 20 replicates for Model (C3) and its higher-dimension variation is listed in Table 4.

The total computation time for CATCH is shorter than most methods, except for ℓ_1-FDA

and ℓ_1-GLM. This shows that CATCH is a computationally efficient method in general. For the comparison of CATCH, ℓ_1-FDA and ℓ_1-GLM, we note that the fast computation of ℓ_1-FDA is somewhat expected, because it assumes that the covariance is diagonal . This simplification greatly improves the computational speed. However, this assumption may lead to lower classification accuracy, as seen in the numerical studies. On the other hand, the penalized estimation step of CATCH has similar computational costs as ℓ_1-GLM because both methods use coordinate descent methods. The major difference between CATCH and ℓ_1-GLM comes from the part where we adjust for the covariates. But we have seen that this step repays us with considerable gain in classification accuracy. Meanwhile, this step can be finished much faster if we implement it in a parallel fashion as the adjustment is element-wise. Hence, the added computational costs of adjusting for the covariates should not be a serious issue.

It is also worth mentioning that CP-GLM (Zhou et al. 2013) has remarkable accuracy (Table 3) and computation speed (Table 4) comparing to other existing methods. Nonetheless, CATCH substantially improves both classification accuracy and computational speed. Moreover, under the higher dimension $80 \times 80 \times 80$, CP-GLM requires a warm-start by first downsizing the tensor to a smaller size and obtaining an initial estimator. Even a rank-1 CP-GLM model has more than 240 model parameters, which is more than the sample size. In contrast, our CATCH model fitting requires no such initialization and is more feasible to high-dimensional sparse situations. In Table 4, we include the warming-up stage of rank-3 CP-GLM. If we use rank-1 CP-GLM instead, the computation time can be reduced to 12.86 seconds, which is however still longer than that of CATCH. In the meantime, rank-1 CP-GLM has a much higher error rate.

7 Real data analysis

In this section, we apply CATCH to a colorimetric sensor array data with matrix predictors, and a neuroimaging application with 3-way tensor predictors and covariates to diagnose the attention deficit hyperactivity disorder (ADHD). The analysis of another dataset on diagnosing autism (the ASD dataset) is presented in the Supplementary Materials. The analysis on the

Dimension		CATCH		ℓ_1-FDA	SOS	CP-GLM
	Adjust	Estimation	Total			
$30 \times 36 \times 30$	0.13	0.17	0.3	0.06	2.79	1.62
$80 \times 80 \times 80$	3.38	4.66	8.04	1.27	70.79	18.34
	STDA	CMDA	DGTDA	ℓ_1-GLM	ℓ_1-SVM	
$30 \times 36 \times 30$	4.63	109.36	1.96	0.19	29.49	
$80 \times 80 \times 80$	22.59	NA	51.16	2.86	NA	

Table 4: Computation time (seconds) averaged from 20 replicates. The parameters were chosen as in Model (C3) (dimension $30 \times 36 \times 30$) and Model (C3H) (in Supplementary Materials, dimension $80 \times 80 \times 80$). The method ℓ_1-SVM and CMDA failed to converge in Model (C3H) in an hour and hence their computation time is reported as "NA".

ADHD and the ASD datasets leads to similar conclusions from the statistical perspective, and hence only one of them is included in the main body of our paper.

7.1 The Colorimetric Sensor Array Data

Colorimetric sensor arrays (CSA) are devices that identify volatile chemical toxicants (VCT). They use chemical dyes to turn the smell of a chemical to optical composite signals. This results in 36×3 matrix predictors, where each row contains the color change of a dye before and after exposure, and the three columns correspond to red, green and blue, respectively.

The CSA data were collected on $n = 147$ chemicals belonging to $K = 21$ classes. One class is non-toxic chemical, while the other 20 classes are high-hazard toxic industrial chemicals. The CSAs were exposed to the chemicals at two conditions: the Immediately Dangerous to Life or Health (IDLH) concentrations for 2 minutes, and the Permissible Exposure Level (PEL) for 5 minutes. The CSA data was used in Zhong & Suslick (2015) to demonstrate MDA and PMDA. Following their approach, we analyze the two conditions separately.

We applied CATCH, ℓ_1-FDA, MDA, PMDA, SOS, ℓ_1-multinomial, Random Forest, SVM and STDA to the IDLH and the PEL levels. In each replicate, we randomly sampled 21 observations as the testing set and used the rest 126 observations as the training set. We used $K - 1 = 20$ discriminant directions for ℓ_1-FDA, MDA and PMDA. Although these methods allow users to choose the number of discriminant directions, we observed that cross validation

Error (%)	CATCH	STDA	DGTDA	CMDA	MDA	PMDA
IDLH	0 (0)	1.7 (0.3)	0.2 (0.1)	0.4 (0.1)	1.6 (0.1)	2.4 (0.1)
PEL	3.2 (0.1)	11.2 (0.6)	7.2 (0.4)	5.1 (0.4)	18.9 (0.2)	19.7 (0.1)
	LDA	ℓ_1-FDA	SOS	ℓ_1-GLM	RF	SVM
IDLH	5.1 (0.3)	0 (0)	0 (0)	0.6 (0.2)	0.2 (0.1)	0.7 (0.2)
PEL	20.1 (0.9)	5.7 (0.1)	10.7 (0.1)	17.5 (0.6)	1.7 (0.3)	4.9 (0.10)

Table 5: Colorimetric sensor array data analysis. Classification error rates of colorimetric sensor array data under IDLH and PEL exposure conditions. Means and standard errors (in parentheses) of error rates of 100 replicates are recorded.

over this parameter led to little improvement of performance on the CSA dataset. The other tuning parameters were chosen by 5-fold cross validation on the training set.

The classification error rates are listed in Table 5. At the IDLH level, all methods have excellent accuracy. In particular, CATCH, ℓ_1-FDA and SOS achieve perfect classification. Meanwhile, at the PEL level, the classification becomes much more difficult, possibly because of the low concentration of chemicals. Random forest is the best classifier, while CATCH is the second best that significantly outperforms all the other methods. However, CATCH also has some noticeable advantages over random forest. First, CATCH performs variable selection to allow easy interpretation. Second, CATCH can handle much higher dimensions, where random forest would not be applicable, such as in the ADHD dataset in Section 7.2. Third, the classifier fitted by random forest is difficult to interpret, while CATCH provides a low-dimensional representation of the data, as we now discuss.

To visualize the classification results of CATCH on IDLH case, we perform principal component analysis on $\langle \widehat{\mathbf{B}}_2, \mathbf{X} \rangle, \ldots, \langle \widehat{\mathbf{B}}_{20}, \mathbf{X} \rangle$, where $\widehat{\mathbf{B}}_k$ are given by the CATCH estimator. Since the first two principal components explain over 90% of the total variability of the 20 discriminative components, we plot these two principal components in Figure 7.1. It can be seen that the different classes fall into different clusters, with very little overlap.

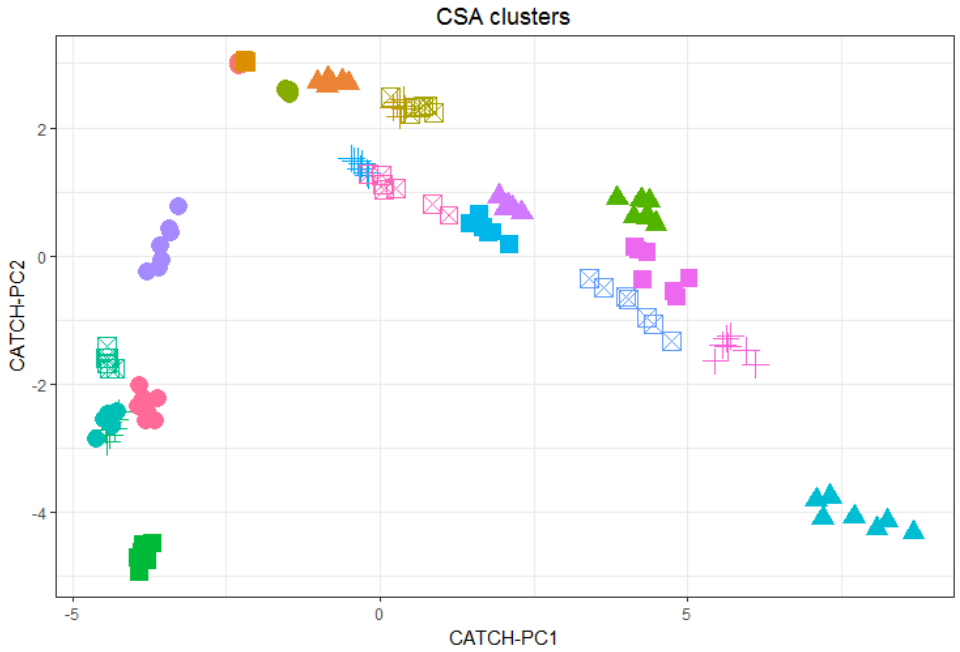

Figure 7.1: Colorimetric sensor array data projected on the first two principal component directions of $\langle \widehat{\mathbf{B}}_k, \mathbf{X} \rangle, k = 2, \ldots, K$ in the IDLH experiment. There are 147 observations from 21 class in total.

Scenarios	$30 \times 36 \times 30$				$24 \times 27 \times 24$			
	Binary		Multiclass		Binary		Multiclass	
	Mean	SE	Mean	SE	Mean	SE	Mean	SE
CATCH	**23.57**	**0.2**	**36.11**	**0.23**	**22.79**	**0.24**	**35.22**	**0.25**
CP-GLM	25.19	0.24	NA	NA	25.05	0.19	NA	NA
STDA	32.44	0.35	49.35	0.33	31.01	0.29	49.35	0.28
DGTDA	29.88	0.30	46.41	0.31	30.51	0.29	47.17	0.30
CMDA	30.3	0.28	46.29	0.32	30.38	0.26	47.94	0.3
SOS	24.11	0.28	37.48	0.26	23.87	0.26	37.07	0.29
Weighted SOS	24.12	0.11	38.89	0.17	24.35	0.10	38.47	0.19
ℓ_1-GLM	23.75	0.17	35.42	0.22	23.99	0.16	35.66	0.21
Weighted ℓ_1-GLM	24.32	0.27	37.81	0.28	23.31	0.22	37.34	0.28
ℓ_1-SVM	26.95	0.3	40.79	0.29	27.54	0.31	41.28	0.32

Table 6: Classification errors on the ADHD datasets of two different tensor sizes. Testing classification error rates, standard errors are based on 100 replicates of training/testing sets. CP-GLM is not applicable for multiclass problem because multinomial logistic is not available.

7.2 The ADHD dataset

We further consider the attention deficit hyperactivity disorder (ADHD) dataset, which contains both tensor predictors and covariates. Neuro Bureau shares the ADHD dataset on NITRC (`http://fcon_1000.projects.nitrc.org/indi/adhd200`) (Bellec et al. 2017). It contains complete rs-fMRI and s-MRI data for 930 individuals, along with their age, gender and handedness. These individuals fall into four categories: Typically Developing Children (TDC), ADHD Combined, ADHD Hyperactive and ADHD Inattentive. The T1-weighted MRI were downsized to $30\times36\times30$ in our analysis. We further downsized the tensors to $24\times27\times24$ and compared the results side-by-side with the $30 \times 36 \times 30$ tensor data. The covariate gender is binary. We stratified the dataset by fitting CATCH on male and female subjects separately. Then we pooled error rates from the two subsets to measure the performance of CATCH. After stratification on gender, we had two continuous covariates, age and handedness. MDA and PMDA could not be applied to this dataset because the images are three-way tensors rather than matrices. The ℓ_1-FDA was not included, because it seemed to be overly sensitive to tuning parameters on this dataset.

We split the data into a training set of 762 subjects and a testing set of 168 subjects. We tested the performance of CATCH in two classification problems. Because only 13 subjects belong to the ADHD Hyperactive category, we combined them with the ADHD Combined class. This gave us a three-class problem. We further considered a binary problem where we grouped ADHD combined and ADHD hyperactive into one class, and the remaining two categories into the other class. Subjects in the former class exhibits hyperactivity, while those do not in the latter class.

Before we fit classifiers on this datasets, we replaced zero tensor element by half of the minimum of the nonzero elements. Then we performed the logarithm transformation $\mathbf{X} \mapsto \log(\mathbf{X})$ such that the variables were on the same scale and more normally distributed. The classification results are listed in Table 6. Overall, CATCH has the best performance among all the methods.

8 Discussion

In this paper, we develop the CATCH model for data with both tensor and covariates. We present a comprehensive study on the integration of the two types of predictors. We also propose the method CATCH to efficiently handle the high dimensions of the tensor predictor. The superior performance of CATCH is demonstrated through both theoretical and numerical studies. Compared with existing approaches for tensor classification, CATCH has several major advantages: (1) it effectively integrates and adjusts for the covariates information; (2) it performs variable selection on the high-dimensional tensor; (3) it is computationally efficient and numerically accurate. Although we only consider low-dimensional continuous covariates, CATCH can be extended to accommodate applications where some of the covariates are discrete, and to applications such as imaging genetics with high-dimensional covariates.

Without the popular low-rank assumption in the high-dimensional tensor literature, CATCH is particularly effective if only a small number of the tensor entries are useful for prediction after adjusted for the covariates. However, if a large number of tensor entries have a collective discriminantive power, the sparsity assumption of CATCH may be vulnerable; and low-rank methods may be more suitable. Some other limitations of the proposed method and potential future research along these lines are discussed in the following.

In the CATCH model, we assume that the covariates and the adjusted tensor predictors are normal with constant covariance matrices across classes. In the future, it will be interesting to study the relaxation of the constant covariance and the normality assumptions. The relaxation of the constant covariance assumption is analogous to the extension from LDA to quadratic discriminant analysis (QDA) in high-dimensional vector data (Fan et al. 2015, Li & Shao 2015, Jiang et al. 2015, Le & Hastie 2014, e.g.). Another important direction for future research is to relax the normality assumption. As pointed out by the associate editor, transformations are often helpful in relaxing normality assumptions. Lin & Jeon (2003), Han et al. (2013), Mai & Zou (2015) discussed methods to transform the data such that they satisfy the discriminant analysis type of assumptions. It will be interesting to investigate the integration of their techniques with CATCH to relax the normality assumption.

References

Bach, F. R. (2008), 'Consistency of the group lasso and multiple kernel learning', *The Journal of Machine Learning Research* **9**(6), 1179–1225.

Bao, Y. T. & Chien, J. T. (2015), 'Tensor classification network', *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)* pp. 1–6.

Becker, N., Werft, W., Toedt, G., Lichter, P. & Benner, A. (2009), 'penalizedsvm: a r-package for feature selection svm classification', *Bioinformatics* **25**(13), 1711–1712.

Bellec, P., Chu, C., Chouinard-Decorte, F., Benhajali, Y., Margulies, D. S. & Craddock, R. C. (2017), 'The neuro bureau adhd-200 preprocessed repository', *NeuroImage* **144**(Pt B), 275 – 286.

Bickel, P. & Levina, E. (2004), 'Some theory for fisher's linear discriminant function, 'naive bayes', and some alternatives when there are many more variables than observations', *Bernoulli* **10**(6), 989–1010.

Bradley, P. S. & Mangasarian, O. L. (1998), 'Feature selection via concave minimization and support vector machines.', *ICML 98* pp. 82–90.

Breiman, L. (2001), 'Random forests', *Machine Learning* **45**(1), 5–32.

Cai, T. & Liu, W. (2011), 'A direct estimation approach to sparse linear discriminant analysis', *Journal of the American Statistical Association* **106**(1), 1566–1577.

Chi, E. C. & Kolda, T. G. (2012), 'On tensors, sparsity, and nonnegative factorizations', *SIAM Journal on Matrix Analysis and Applications* **33**(4), 1272–1299.

Chiaromonte, F., Cook, R. D. & Li, B. (2002), 'Sufficient dimension reduction in regressions with categorical predictors', *The Annals of Statistics* **30**(2), 475–497.

Clemmensen, L., Hastie, T., Witten, D. & Ersbøll, B. (2011), 'Sparse discriminant analysis', *Technometrics* **53**(4), 406–413.

Cook, R. D. (1998), *Regression Graphics: Ideas for Studying Regressions Through Graphics*, Vol. 318, John Wiley & Sons.

Cortes, C. & Vapnik, V. (1995), 'Support-vector networks', *Machine Learning* **20**(3), 273–297.

Dimitriadou, E., Hornik, K., Leisch, F., Meyer, D. & Weingessel, A. (2009), 'E1071: Misc functions of the department of statistics (e1071), tu wien', *CRAN R package* .

Dutilleul, P. (1999), 'The mle algorithm for the matrix normal distribution', *Journal of Statistical Computation and Simulation* **64**(2), 105–123.

Fan, J. & Fan, Y. (2008), 'High dimensional classification using features annealed independence rules', *The Annals of statistics* **36**(6), 2605–2637.

Fan, J., Feng, Y. & Tong, X. (2012), 'A road to classification in high dimensional space: the regularized optimal affine discriminant', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **74**(4), 745–771.

Fan, J., Ke, Z. T., Liu, H. & Xia, L. (2015), 'Quadro: A supervised dimension reduction method via rayleigh quotient optimization', *The Annals of statistics* **43**(4), 1498–1534.

Feng, Z., Wen, X. M., Yu, Z. & Zhu, L. (2013), 'On partial sufficient dimension reduction with applications to partially linear multi-index models', *Journal of the American Statistical Association* **108**(501), 237–246.

Friedman, J., Hastie, T. & Tibshirani, R. (2001), *The elements of statistical learning*, Vol. 1, Springer series in statistics Springer, Berlin.

Friedman, J., Hastie, T. & Tibshirani, R. (2010), 'Regularization paths for generalized linear models via coordinate descent', *Journal of Statistical Software* **33**(1), 1–22.

Fung, G. M. & Mangasarian, O. L. (2004), 'A feature selection newton method for support vector machine classification', *Computational Optimization and Applications* **28**(2), 185–202.

Goeman, J., Meijer, R. & Chaturvedi, N. (2012), 'penalized: L1 (lasso and fused lasso) and l2 (ridge) penalized estimation in glms and in the cox model', *CRAN R package* .

Gupta, A. K. & Nagar, D. K. (1999), *Matrix variate distributions*, Vol. 104, CRC Press.

Han, F., Zhao, T. & Liu, H. (2013), 'Coda: High dimensional copula discriminant analysis', *Journal of Machine Learning Research* **14**(2), 629–671.

Hand, D. J. (2006), 'Classifier technology and the illusion of progress', *Statistical Science* **21**(1), 1–14.

Hastie, T., Tibshirani, R. & Wainwright, M. (2015), *Statistical learning with sparsity: the lasso and generalizations*, CRC press.

Hoff, P. D. (2015), 'Multilinear tensor regression for longitudinal relational data', *The Annals of Applied Statistics* **9**(3), 1169–1193.

Jiang, B., Wang, X. & Leng, C. (2015), 'Quda: A direct approach for sparse quadratic discriminant analysis', *arXiv preprint arXiv:1510.00084* .

Kolda, T. G. & Bader, B. W. (2009), 'Tensor decompositions and applications', *SIAM Review* **51**(3), 455–500.

Lai, Z., Xu, Y., Yang, J., Tang, J. & Zhang, D. (2013), 'Sparse tensor discriminant analysis', *IEEE Transactions on Image Processing* **22**(10), 3904–3915.

Le, Y. & Hastie, T. (2014), 'Sparse quadratic discriminant analysis and community bayes', *arXiv preprint arXiv:1407.4543* .

Ledoit, O. & Wolf, M. (2004), 'A well-conditioned estimator for large-dimensional covariance matrices', *Journal of Multivariate Analysis* **88**(2), 365–411.

Li, L. & Zhang, X. (2017), 'Parsimonious tensor response regression', *Journal of the American Statistical Association* **112**(519), 1131–1146.

Li, Q. & Schonfeld, D. (2014), 'Multilinear discriminant analysis for higher-order tensor data classification', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **36**(12), 2524–2537.

Li, Q. & Shao, J. (2015), 'Sparse quadratic discriminant analysis for high dimensional data', *Statistica Sinica* **25**(2), 457–473.

Liaw, A. & Wiener, M. (2014), 'Package 'randomforest': Breiman and cutler's random forests for classification and regression', *CRAN R package* .

Lin, Y. & Jeon, Y. (2003), 'Discriminant analysis through a semiparametric model', *Biometrika* **90**(2), 379–392.

Liu, T., Yuan, M. & Zhao, H. (2017), 'Characterizing spatiotemporal transcriptome of human brain via low rank tensor decomposition', *arXiv preprint arXiv:1702.07449* .

Lock, E. F. (2017), 'Tensor-on-tensor regression', *Journal of Computational and Graphical Statistics* **In press**.

Mai, Q., Yang, Y. & Zou, H. (2017), 'Multiclass sparse discriminant analysis', *Statistica Sinica* **In press**.

Mai, Q. & Zou, H. (2015), 'Sparse semiparametric discriminant analysis', *Journal of Multivariate Analysis* **135**(1), 175–188.

Mai, Q., Zou, H. & Yuan, M. (2012), 'A direct approach to sparse discriminant analysis in ultra-high dimensions', *Biometrika* **99**(1), 29–42.

Manceur, A. M. & Dutilleul, P. (2013), 'Maximum likelihood estimation for the tensor normal distribution: Algorithm, minimum sample size, and empirical bias and dispersion', *Journal of Computational and Applied Mathematics* **239**(1), 37–49.

Michie, D., Spiegelhalter, D. J. & Taylor, C. C. (1994), *Machine learning, neural and statistical classification*, Ellis Horwood.

Pan, R., Wang, H. & Li, R. (2016), 'Ultrahigh-dimensional multiclass linear discriminant analysis by pairwise sure independence screening', *Journal of the American Statistical Association* **111**(513), 169–179.

Raskutti, G. & Yuan, M. (2015), 'Convex regularization for high-dimensional tensor regression', *arXiv preprint arXiv:1512.01215* .

Shao, J., Wang, Y., Deng, X. & Wang, S. (2011), 'Sparse linear discriminant analysis by thresholding for high dimensional data', *The Annals of Statistics* **39**(2), 1241–1265.

Sun, W. W., Lu, J., Liu, H. & Cheng, G. (2016), 'Provable sparse tensor decomposition', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **79**(3), 899–916.

Tao, D., Li, X., Wu, X. & Maybank, S. J. (2007), 'General tensor discriminant analysis and gabor features for gait recognition', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **29**(10), 1700–1715.

Tibshirani, R. (1996), 'Regression shrinkage and selection via the lasso', *Journal of the Royal Statistical Society, Series B* **58**(1), 267–288.

Wang, X. & Zhu, H. (2017), 'Generalized scalar-on-image regression models via total variation', *Journal of the American Statistical Association* **112**(519), 1156–1168.

Werner, K., Jansson, M. & Stoica, P. (2008), 'On estimation of covariance matrices with kronecker product structure', *Signal Processing, IEEE Transactions on* **56**(2), 478–491.

Wimalawarne, K., Tomioka, R. & Sugiyama, M. (2016), 'Theoretical and experimental analyses of tensor-based regression and classification', *Neural Computation* **28**(4), 686–715.

Witten, D. M. & Tibshirani, R. (2011), 'Penalized classification using fisher's linear discriminant', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **73**(5), 753–772.

Xu, P., Zhu, J., Zhu, L. & Li, Y. (2015), 'Covariance-enhanced discriminant analysis', *Biometrica* **102**(1), 33–45.

Yan, S., Xu, D., Yang, Q., Zhang, L., Tang, X. & Zhang, H.-J. (2005), 'Discriminant analysis with tensor representation', *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* pp. 526–532.

Yuan, M. & Lin, Y. (2006), 'Model selection and estimation in regression with grouped variables', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **68**(1), 49–67.

Zeng, R., Wu, J., Senhadji, L. & Shu, H. (2015), 'Tensor object classification via multilinear discriminant analysis network', *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* pp. 1971–1975.

Zhang, A. & Xia, D. (2017), 'Tensor svd: Statistical and computational limits', *IEEE Transactions on Information Theory* **In press**.

Zhang, X. (2013), 'New developments for net-effect plots', *Wiley Interdisciplinary Reviews: Computational Statistics* **5**(2), 105–113.

Zhang, X. & Li, L. (2017), 'Tensor envelope partial least-squares regression', *Technometrics* **59**(4), 426–436.

Zhao, J. & Leng, C. (2014), 'Structured lasso for regression with matrix covariates', *Statistica Sinica* **24**(2), 799–814.

Zhong, W. & Suslick, K. S. (2015), 'Matrix discriminant analysis with application to colorimetric sensor array data', *Technometrics* **57**(4), 524–534.

Zhou, H. & Li, L. (2014), 'Regularized matrix regression', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **76**(2), 463–483.

Zhou, H., Li, L. & Zhu, H. (2013), 'Tensor regression with applications in neuroimaging data analysis', *Journal of the American Statistical Association* **108**(502), 540–552.

Zhou, S. (2014), 'Gemini: graph estimation with matrix variate normal instances', *The Annals of Statistics* **42**(2), 532–562.