Published in final edited form as:
J Comput Graph Stat. 2023 ; 32(2): 730–743. doi:10.1080/10618600.2022.2099404.

Multiway sparse distance weighted discrimination

Bin Guo[1], **Lynn E. Eberly**[1,2], **Pierre-Gilles Henry**[2], **Christophe Lenglet**[2], **Eric F. Lock**[1]

[1]Division of Biostatistics, School of Public Health

[2]Center for Magnetic Resonance Research, University of Minnesota

Abstract

Modern data often take the form of a multiway array. However, most classification methods are designed for vectors, i.e., 1-way arrays. Distance weighted discrimination (DWD) is a popular high-dimensional classification method that has been extended to the multiway context, with dramatic improvements in performance when data have multiway structure. However, the previous implementation of multiway DWD was restricted to classification of matrices, and did not account for sparsity. In this paper, we develop a general framework for multiway classification which is applicable to any number of dimensions and any degree of sparsity. We conducted extensive simulation studies, showing that our model is robust to the degree of sparsity and improves classification accuracy when the data have multiway structure. For our motivating application, magnetic resonance spectroscopy (MRS) was used to measure the abundance of several metabolites across multiple neurological regions and across multiple time points in a mouse model of Friedreich's ataxia, yielding a four-way data array. Our method reveals a robust and interpretable multi-region metabolomic signal that discriminates the groups of interest. We also successfully apply our method to gene expression time course data for multiple sclerosis treatment. An R implementation is available in the package MultiwayClassification at http://github.com/lockEF/MultiwayClassification.

Keywords

Distance weighted discrimination; Multiway Classification; Sparsity; Tensors

1 Introduction

Development and wide deployment of advanced technologies have produced tools that generate massive amounts of data with complex structure. These data are often represented as a multiway array, which extends the two-way data matrix to higher dimensions. This paper concerns the task of classification from multiway data. As our motivating data application we consider magnetic resonance spectroscopy (MRS) data for a study

Eric.F.Lock@gmail.com .

Competing Interests
The authors report there are no competing interests to declare.

Supplementary Materials
A supplementary document describes the results of several additional simulation studies to assess and compare the multiway sparse DWD method, and additional details of the application in Section 6.1.

of Friedreich's Ataxia in mice. The MRS data measures the concentration of several metabolites across multiple regions of the brain, and across multiple time points after a treatment, yielding a four-way data array: mice × metabolites × regions × time. We are interested in identifying signal that distinguishes the treatment groups using the totality of the MRS data.

A naive approach is to transform the multiway array to a vector, and then apply high-dimensional classifiers designed for vector-valued data to the transformed vector. However, the effects of the same metabolite in different brain regions and at different time points on the classification are very likely correlated. Ignoring the dependence across different dimensions may result in inaccurate classifications and complicate interpretation. Thus, the multiway structure should be considered in the model. In addition, only some metabolites may be useful for distinguishing the classes, and these distinctions may only be present for some time points. Exclusion of uninformative features can improve classification performance and interpretation, and this motivates an approach that also accommodates a sparse structure. In what follows we briefly review existing high-dimensional classifiers (Section 1.1), sparse high-dimensional classifiers (Section 1.2), and classifiers for multiway data (Section 1.3); our methodological contributions are summarized in Section 1.4.

1.1 High-dimensional classification of vectors

Traditional approaches to classification like logistic regression and Fisher's linear discriminant analysis (LDA) are prone to overfitting with a large number of features, and this has motivated several classification methods for high-dimensional vector-valued data. We can roughly divide these methods into two categories: non-linear classifiers such as k-nearest neighbor classification (Cover and Hart 1967) and random forests (Breiman 2001), and linear classifiers such as penalized LDA (Witten and Tibshirani 2011), support vector machines (SVM) (Cortes and Vapnik 1995), and distance weighted discrimination (DWD) (Marron et al. 2007). Non-linear classifiers are flexible to use as they require minimal assumptions, but may not have straightforward interpretations. In contrast, linear classifiers use a weighted sum of the measured features. SVM and DWD are both commonly used in biomedical research. However, DWD tends to outperform SVM in scenarios with high-dimensional data and a lower sample size; here SVM suffers from the data piling problem, which means many cases will pile up at the discriminating margins as a symptom of overfitting (Marron et al. 2007).

1.2 High-dimensional classification of vectors with sparsity

In their original formulations, SVM, DWD, and other linear classification approaches use all available variables. However, in practice there may be only a few important variables affecting the outcome, especially in biomedical applications to disease classification with imaging or genomics data (Zou 2019). Thus, methods that use all variables for classification may include too many noise features and deteriorate the classification performance due to error accumulation (Fan and Fan 2008). Sparse methods, in which only a subset of variables are used for classification, can improve performance in this respect and also improve interpretation by identifying a small number of informative features. Zou (2019) gave a detailed review of high-dimensional classification methods that can account for sparsity. Many sparse classifiers were derived based on high dimensional extensions of the classical discriminant analysis, such as linear programming discriminant (LPD) (Cai and Liu 2011), penalized LDA (Witten and Tibshirani 2011), regularized optimal affine discriminant (ROAD) (Fan et al. 2012) and direct sparse discriminant analysis (DSDA) (Mai et al. 2012). Other methods have extended margin-based approaches like SVM and DWD. Hastie et al. (2009) gives a nice complete introduction of SVM from its geometric view to its statistical loss + penalty formulation. A class of methods extended SVM to enforce sparsity by using different loss functions or penalties, such as lasso penalized SVM (Bradley and Mangasarian 1998), elastic-net penalized SVM (Wang et al. 2008) and SCAD penalized SVM (Zhang et al. 2006). For DWD, the objective function can also be represented as an analogous loss + penalty formulation (Liu et al. 2011). Wang and Zou (2016) proposed sparse DWD (SDWD) by adding lasso and elastic-net penalties on the model coefficients, which can improve performance and efficiency of DWD in high-dimensional classification.

1.3 High-dimensional classification of multiway arrays

There is a growing literature on multiway classification by extending classifiers of vectors to multiway arrays using factorization and dimension reduction techniques. Ye et al. (2004) and Bauckhage (2007) extended LDA and related approaches to multiway data. Tao et al. (2005) proposed a supervised tensor learning framework by performing a rank-1 decomposition on the coefficients to reduce dimension, in which the coefficients are factorized into a single set of weights for each dimension. Wimalawarne et al. (2016) investigated tensor-based classification where a logistic loss function and a penalty term with different continuous tensor norms for the coefficients are considered. Pan et al. (2018) developed a classification approach that allows for multiway data and covariate adjustment; their proposal, termed Covariate-Adjusted Tensor Classification in High Dimensions (CATCH), assumes a multiway structure in the residual covariance but not in the signals discriminating the classes. Lyu et al. (2017) proposed multiway versions of DWD and SVM under the assumption that the coefficient array is low-rank. Their implementation of multiway DWD was shown to dramatically improve performance over two-way classifiers when the data have multiway structure, and also tended to outperform analogous extensions of SVM across all applications and simulation scenarios. However, their method is restricted to use for three-way data and does not account for sparsity.

1.4 High-dimensional classification of multiway arrays with sparsity: present contributions

In this paper, we developed a general framework for multiway classification that is applicable to any number of dimensions and any degree of sparsity. Because of its strong performance relative to SVM in prior work, we focus on extensions of DWD. Our proposed approach builds a connection between the two approaches of multiway DWD and sparse DWD. The central assumption is that the signal discriminating the groups can be efficiently represented by meaningful patterns in each dimension, which we identify by imposing a low-rank structure on the coefficient array. Adding penalty terms to multiway DWD can enforce sparsity and give better performance even when the data are not sparse. The rest of the paper is organized as follows. In Section 2 we introduce our formal mathematical framework and notation. In Section 3 we briefly review the standard DWD, sparse DWD and multiway DWD methods. In Section 4 we describe our proposed multiway sparse DWD method and algorithms for estimation. We show the multiway approach has improved performance and interpretation compared with existing methods through simulation studies (Section 5). We considered two data applications in Section 6, and these results showed that the multiway sparsity model is robust to data with any degree of sparsity and has competitive classification performance over other classification techniques. The article concludes with discussion and future directions in Section 7.

2 Notation and framework

Throughout this article bold lowercase characters (**a**) denote vectors, bold uppercase characters (**A**) denote matrices, and blackboard bold uppercase characters ($\mathbb{A}$) denote multiway arrays of the specified dimension (e.g., $\mathbb{A}: P_1 \times P_2 \times \cdots \times P_K$). Square brackets index entries within an array, e.g., $\mathbb{A}[p_1, p_2, \cdots, p_K]$. The operators $\|\cdot\|_1$ and $\|\cdot\|_2$ define the generalization of the L_1 and L_2 norms, respectively:

$$\|\mathbb{A}\|_1 = \sum_{p_1=1}^{P_1} \cdots \sum_{p_K=1}^{P_K} |\mathbb{A}[p_1, p_2, \cdots, p_K]|, \|\mathbb{A}\|_2^2 = \sum_{p_1=1}^{P_1} \cdots \sum_{p_K=1}^{P_K} \mathbb{A}[p_1, p_2, \cdots, p_K]^2 .$$

The generalized inner product for two arrays $\mathbb{A}$ and $\mathbb{B}$ of the same dimension is

$$\mathbb{A} \cdot \mathbb{B} = \sum_{p_1=1}^{P_1} \cdots \sum_{p_K=1}^{P_K} \mathbb{A}[p_1, \ldots, p_K]\mathbb{B}[p_1, \ldots, p_K],$$

the generalized outer product for $\mathbb{A}: P_1 \times \cdots \times P_K$ and $\mathbb{B}: Q_1 \times \cdots \times Q_L$ is $\mathbb{A} \circ \mathbb{B}: P_1 \times \cdots \times P_K \times Q_1 \ldots Q_L$ where

$$\mathbb{A} \circ \mathbb{B}[p_1, \ldots, p_K, q_1, \ldots, q_L] = \mathbb{A}[p_1, \ldots, p_K]\mathbb{B}[q_1, \ldots, q_L] .$$

For our context, $\mathbb{X}: N \times P_1 \times \cdots \times P_K$ gives data in the form of a K-way array for $\boldsymbol{N}$ subjects, where P_k is the size of the k^{th} dimension for $k = 1, \ldots, K$. Each subject belongs to one of two classes denoted by −1 and + 1; let $y_i \in \{-1, 1\}$ give the class label for each subject and $\mathbf{y} = [\mathrm{y}_1, \ldots, y_N]$. Our goal is to predict the class labels **y** based on the multiway covariates $\mathbb{X}$.

3 DWD and its extensions

3.1 Distance Weighted Discrimination (DWD)

Here we briefly describe the standard DWD for high-dimensional vector-valued $\mathbf{x}_i$ for each subject, given by the rows of $\mathbf{X}$: $N \times P$. The goal of DWD and related methods is to find the hyperplane $\mathbf{b} = [b_1, \ldots, b_p]$ which best separates the two classes via the subject *scores* **Xb**. To solve this standard binary classification problem, SVM (Cortes and Vapnik 1995) identifies the hyperplane that maximizes the margin separating the two classes. However, SVM can suffer from the data piling problem as shown in Marron et al. (2007), which means many data points may pile up on the margin when the SVM is applied to high-dimensional data. To tackle this issue, they proposed distance weighted discrimination (DWD) which finds the separating hyperplane that minimizes the sum of the inverse distance from the data points to the hyperplane. The standard DWD is formulated as the following optimization problem:

$$\underset{\mathbf{r}, \mathbf{b}, b_0, \xi}{\operatorname{argmin}} \sum_{i=1}^{N} \frac{1}{r_i} + C\,\xi_i$$

subject to $r_i = y_i(\mathbf{x}_i^T\mathbf{b} + b_0) + \xi_i \geq 0$ and $\xi_i \geq 0$ for $i = 1, \ldots, N$, and $\| \mathbf{b} \| \leq 1$. Here, C is the penalty parameter and b_0 is an intercept term. The classification rule is given by the sign of $\mathbf{x}_i\mathbf{b} + b_0$, and thus ξ_i can be considered a penalty for misclassification.

3.2 Sparse DWD

The standard DWD may not be suitable for high dimensional classification when the underlying signal is sparse, as it does not conduct variable selection. To further improve performance and efficiency of DWD in high-dimensional classification, Wang and Zou (2016) proposed sparse DWD by adding penalties on the model coefficients to enforce sparsity. The generalization of standard DWD to sparse DWD is based on the fact that the objective function for standard DWD can be decomposed into two components: loss function and penalty (Liu et al. 2011):

$$\underset{b_0, \mathbf{b}}{\arg\min} \frac{1}{N} \sum_{i=1}^{N} V\left(y_i\left(b_0 + \mathbf{x}_i^T\mathbf{b}\right)\right) + \frac{\lambda_2}{2}\left\|\mathbf{b}\right\|_2^2 \quad (1)$$

where the loss function is given by

$$V(u) = \begin{cases} 1 - u, & \text{if } u \leq 1/2 \\ 1/(4u), & \text{if } u > 1/2. \end{cases} \quad (2)$$

To account for sparsity, many variations of penalties can be added in the model, such as lasso and elastic-net (Wang and Zou 2016). The elastic-net penalty often outperforms the lasso in prediction, and thus the elastic-net penalized DWD is attractive, with the objective function

$$\underset{b_0, \mathbf{b}}{\arg\min} \frac{1}{N} \sum_{i=1}^{N} V\left(y_i\left(b_0 + \mathbf{x}_i^T\mathbf{b}\right)\right) + P_{\lambda_1, \lambda_2}(\mathbf{b}) \quad (3)$$

with $V(\cdot)$ defined as in (2) and

$$P_{\lambda_1, \lambda_2}(\mathbf{b}) = \lambda_1 \|\mathbf{b}\|_1 + \frac{\lambda_2}{2}\|\mathbf{b}\|_2^2$$

in which λ_1 and λ_2 are tuning parameters for regularization. Both parameters control the shrinkage of the coefficients toward 0. However, the L_1 penalty controlled by λ_1 may result in some model coefficients being shrunk exactly to 0, and in this way, sparsity (retention of relatively few of the covariates for the classification) is imposed. In practice, λ_1 and λ_2 can be determined by cross-validation. Comparing the objectives (1) and (3), note that the penalized elastic net DWD is equivalent to standard DWD when $\lambda_1 = 0$.

3.3 Multiway DWD

The standard DWD (Section 3.1) and sparse DWD (Section 3.2) are designed for high-dimensional classification on vector-valued data. Lyu et al. (2017) proposed a multiway DWD model which extends the standard DWD from single vector to multiway features. The assumption of this model is that the multiway coefficient matrix can be decomposed into patterns that are particular to each dimension, giving a low-rank-representation.

Consider classification of subjects with matrix valued covariates $\mathbf{X}_i$: $P_1 \times P_2$, concatenated to form the three-way array $\mathbb{X}: N \times P_1 \times P_2$ (for example, mice by metabolites by brain regions). In this multiway context, the coefficients take the form of a matrix $\mathbf{B}$: $P_1 \times P_2$, and the separating hyperplane is given by $f(\mathbf{X}_i) = \mathbf{X}_i\mathbf{B}$ where $\mathbf{B}$ is the coefficient matrix. The rank-1 multiway DWD model assumes that the coefficient matrix $\mathbf{B}$ has the rank-1 decomposition: $\mathbf{B} = \mathbf{u}_1\mathbf{u}_2^\top$, where $\mathbf{u}_1$: $P_1 \times 1$ and $\mathbf{u}_2$: $P_2 \times 1$ denote vectors of weights for each dimension. Under this assumption, the hyperplane to separate the two classes is:

$$f(\mathbf{X}_i) = \mathbf{X}_i \cdot \mathbf{B} = \sum_{i_1 = 1}^{P_1} \sum_{i_2 = 1}^{P_2} \mathbf{X}_i[i_1, i_2]\mathbf{u}_1[i_1]\mathbf{u}_2[i_2] .$$

For example, for MRS data of the form $\mathbf{X}_i$: Metabolites × Regions, $\mathbf{u}_1$ gives a discriminating profile across the metabolites and $\mathbf{u}_2$ weights that profile across the different regions. The rank R model allows for additional patterns in each dimension via $\mathbf{B} = \mathbf{U}_1\mathbf{U}_2^\top$ where $\mathbf{U}_1$: $P_1 \times R$ and $\mathbf{U}_2$: $P_2 \times R$. The coefficients are estimated by iteratively updating the weights in each dimension to optimize an objective function. It has been shown that this multiway DWD model can improve classification accuracy when the underlying true model has a multiway structure and can provide a simple and straightforward interpretation (Lyu et al. 2017).

4 Proposed methods

We propose a general framework for classifying high-dimensional multiway data that combines aspects of sparse DWD (Section 3.2) and multiway DWD (M-DWD) (Section 3.3). The proposed method can be considered as a multiway version of sparse DWD that allows for any number of dimensions $\mathbb{X}_i: P_1 \times \cdots \times P_K$, combined across subjects to form $\mathbb{X}: N \times P_1 \times \cdots \times P_K$. In the following we first describe our generalization of multiway DWD to an arbitrary number of dimensions (Section 4.1), then we describe our sparsity inducing objective function classification method that assumes the coefficient array has a low-rank decomposition (Section 4.2).

4.1 Generalized multiway DWD

The generalized rank-1 multiway model assumes that the coefficient array $\mathbb{B}: P_1 \times \cdots \times P_K$ has a rank-1 decomposition $\mathbb{B} = \mathbf{u}_1 \circ \mathbf{u}_2 \circ \cdots \circ \mathbf{u}_K$, where $\{\mathbf{u}_k\}_{k=1}^K$ denote the vector of weights for each dimension. The hyperplane for separating the two classes is $f(\mathbb{X}_i) = \mathbb{X}_i \cdot \mathbb{B}$, and thus for the rank-1 model

$$f(\mathbb{X}_i) = \sum_{i_1 = 1}^{P_1} \cdots \sum_{i_K = 1}^{P_K} \mathbb{X}[i, i_1, \cdots, i_K]\mathbf{u}_1[i_1]\mathbf{u}_2[i_2]\cdots\mathbf{u}_K[i_K] .$$

For our motivating application, $\mathbf{u}_1$ identifies a metabolite profile that is weighted ($\mathbf{u}_2$) and time points ($\mathbf{u}_3$) (or equivalently, a time profile that is over the regions (weighted across the metabolites and regions) to discriminate the classes.

The rank-1 model assumes the classification is given by combining a single pattern in each dimension. However, in practice multiple patterns may contribute, e.g., a different metabolite profile may affect the classification for different regions of the brain. Thus, we propose a rank-R model for the coefficient array $\mathbb{B}$ that is assumed to have a rank-R Candecomp/Parafac (CP) factorization (Harshman 1970):

$$\mathbb{B} = [[\mathbf{U}_1, \ldots, \mathbf{U}_K]] = \sum_{r = 1}^{R} \mathbf{u}_{1r} \circ \cdots \circ \mathbf{u}_{Kr}$$

where $\mathbf{U}_k$: $P_k \times R$ for $k = 1, \ldots, K$ with columns $\mathbf{u}_{kr}$ as the weight for the k^{th} dimension and r^{th} rank component. The coefficient array $\mathbb{B}$ in the rank-1 multiway model is a special case of the rank-R multiway model when $R = 1$. Moreover, as the CP factorization extends the matrix rank, this model is equivalent to that for the multiway DWD approach in Section 3.3 when $K = 2$. Note that with no constraints on the coefficient array $\mathbb{B}$ the number of free parameters is $\prod_{k=1}^{K} P_k$, and with the rank-R constraint $\mathbb{B}$ has $R(P_1 + \cdots + P_K)$ free parameters. Thus, in addition to facilitating interpretation of relevant patterns in each dimension, a low-rank approach can reduce over-fitting and improve performance when there is multiway structure. While further restrictions (e.g., orthogonality) are needed for the identifiable of the components when $K = 2$, the components are identifiable under a much broader set of conditions when $K > 2$, including linear independence of the columns $\mathbf{U}_k$ in at least two dimensions (Bro 1997). Thus, while we rotate the components to satisfy orthogonality for the two-way model in Section 3.3, no such step is needed for the higher-order context.

4.2 Objective functions

To allow for sparsity in the generalized multiway DWD model, we consider an extension of the sparse DWD objective (3),

$$h(\mathbf{y}, \mathbb{X}; \mathbb{B}, b_0) = \frac{1}{N}\sum_{i = 1}^{N} V\left(\mathbf{y}_i\left(b_0 + \mathbb{X}_i \cdot \mathbb{B}\right)\right) + P_{\lambda_1, \lambda_2}(\mathbb{B}), \tag{4}$$

which we minimize under the restriction that rank $(\mathbb{B}) = \mathrm{R}$. For the penalty $P_{\lambda_1, \lambda_2}(\mathbb{B})$ we consider different extensions of the elastic net to low-rank multiway coefficients, which do or do not distribute the L_1 and L_2 penalties across the factorization components:

$$P^{\mathbb{B}}_{\lambda_1,\lambda_2}(\mathbb{B}) = \lambda_1\|\mathbb{B}\|_1 + \frac{\lambda_2}{2}\|\mathbb{B}\|_2^2$$

$$P^{\mathrm{U}}_{\lambda_1,\lambda_2}(\mathbb{B}) = \lambda_1\sum_{r=1}^{R}\prod_{k=1}^{K}\|\mathbf{u}_{kr}\|_1 + \frac{\lambda_2}{2}\sum_{r=1}^{R}\prod_{k=1}^{K}\|\mathbf{u}_{kr}\|_2^2, \text{ or } P_{\lambda_1,\lambda_2}(\mathbb{B}) = \lambda_1\sum_{r=1}^{R}\prod_{k=1}^{K}\|\mathbf{u}_{kr}\|_1 + \frac{\lambda_2}{2}\|\mathbb{B}\|_2^2.$$

For $R = 1$, the three penalties are equivalent. For $R > 1$, we find that the solution of the objective for the penalty $P^{\mathbf{U}}_{\lambda_1,\lambda_2}(\mathbb{B})$ is often shrunk to a lower rank than the specified rank R (see Section S2 of the supplementary material). This is due to the distributed L_2 penalty term, and similar behavior has been observed in other contexts; for example, when $K = 2$ the distributed L_2 penalty is equivalent to the nuclear norm penalty for matrices (Lock 2018), which favors a smaller rank.

However, the distributed version of the L_1 penalty in $P^{\mathbf{U}}_{\lambda_1,\lambda_2}(\cdot)$ and $P_{\lambda_1,\lambda_2}(\cdot)$ intuitive because it enforces sparsity in the weights for each dimension (e.g., for the metabolites, regions, and time points) separately for each rank-1 component. For example, if different subsets of metabolites are discriminative in different brain regions, that is efficiently captured by multiple rank-1 components with distinct sparsity. Moreover, $P_{\lambda_1,\lambda_2}(\cdot)$ is amenable to coordinate-wise approaches to optimization (see Section 4.3), which are not straightforward for $P^{\mathbb{B}}_{\lambda_1,\lambda_2}(\cdot)$ Thus, in what follows, we use $P_{\lambda_1,\lambda_2}(\cdot)$ as our penalty.

This objective function for generalized sparse multiway DWD subsumes standard DWD, multiway DWD, and sparse DWD. The objective function is equivalent to that for standard DWD when $K = 1$ and $\lambda_1 = 0$ (Marron et al. 2007; Liu et al. 2011), it is equivalent to sparse DWD (Wang and Zou 2016) when $K = 1$ and $\lambda_1 > 0$, and it is equivalent to multiway DWD (Lyu et al. 2017) if $\lambda_1 = 0$ and $K = 2$.

4.3 Optimization

Here we describe the estimation algorithm for fixed penalty parameters λ_1 and λ_2; selection of these parameters is discussed in Section 4.4. To obtain estimates of the coefficient array $\mathbb{B}$ and intercept $\mathbf{b}_0$, we iteratively optimize the objective function $h(\mathbf{y}, \mathbb{X}, \mathbb{B}, b_0)$ (4) for each dimension to obtain the estimated weights for that dimension with other dimension's weights fixed. This general iterative estimation approach is described in Algorithm 1.

1. *Initialization*. Generate K random matrices $\mathbf{U}_k$: $P_k \times R$ for $k = 1, \ldots, K$. In our implementation, entries are generated independently from a Uniform[0,1] distribution. Compute the coefficient array $\mathbb{B} = [[\mathbf{U}_1, \ldots, \mathbf{U}_K]]$, and initialize $\tilde{b}_0 = 0$.

2. *Iteration*. Update $\mathbf{U}_1$ and $\tilde{b}_0$ by optimizing the conditional objective,

$$\left\{\tilde{b}_0, \mathbf{U}_1\right\} = \underset{\{b_0, \mathbf{U}_1\}}{\arg\min}\, h(\mathbf{y}, \mathbb{X}; [[\mathbf{U}_1, \mathbf{U}_2, \ldots, \mathbf{U}_K]], b_0)\,. \tag{5}$$

Similarly update $\mathbf{U}_2, \ldots, \mathbf{U}_K$, with the value for b_0 updated at each step. The details of the optimization sub-step (5) are described below.

3. *Convergence*. Set $\mathbb{B}^{\text{New}} = [[\mathbf{U}_1, \ldots, \mathbf{U}_K]]$ If $\|\mathbb{B}^{\text{New}} - \mathbb{B}\|_2^2 > \epsilon$ for some prespecified threshold $\boldsymbol{\epsilon}$, update $\mathbb{B} = \mathbb{B}^{\text{New}}$ and repeat Step 2. Otherwise, our final estimates are $\mathbb{B} = \mathbb{B}$ and $\hat{b}_0 = \tilde{b}_0$.

Algorithm 1: General estimation steps.

The procedure for solving the optimization problem in equation (5) depends on whether the model is rank-1 ($R = 1$) or higher rank ($R \geq 2$). If $R = 1$, then $\mathbb{B} = \mathbf{u}_1 \circ \cdots \circ \mathbf{u}_2$ and the objective in (5) can be expressed as

$$\frac{1}{N}\sum_{i=1}^{N} V\left(y_i\left(b_0 + \mathbf{x}_i^{(1)T}\mathbf{u}_1\right)\right) + \left(\lambda_1 \prod_{k=2}^{K} \|\mathbf{u}_k\|_1\right)\|\mathbf{u}_1\|_1 + \left(\frac{\lambda_2}{2}\prod_{k=2}^{K}\|\mathbf{u}_k\|_2^2\right)\|\mathbf{u}_1\|_2^2,$$

where

$$\mathbf{x}_i^{(1)}[p_1] = \mathbb{X}_i[p_1, \ldots] \cdot (\mathbf{u}_2 \circ \cdots \circ \mathbf{u}_K) \text{ for } p_1 = 1, \ldots, P_1 . \tag{6}$$

This is equivalent to the vector sparse DWD objective (3) as a function of b_0 and $\mathbf{u}_1$, and thus can be solved using existing software such as the sdwd function in the R package SDWD (Wang and Zou 2015). For rank $R \geq 2$, we solve (5) using a coordinate descent algorithm based on the majorization-minimization (MM) principle (Hunter and Lange 2004), thus extending the algorithm described in Wang and Zou (2016). The steps for this procedure are given in Algorithm 2, and analogous updates are used for $\mathbf{U}_2, \ldots, \mathbf{U}_K$.

To derive Step b of Algorithm 2, consider replacing $\mathbf{U}_1[j, r]$ with $\mathbf{U}_1[j, r]$ in $\mathbb{B}$ to obtain $\mathbb{B}*$. Then,

$$h(\mathbf{y}, \mathbb{X}_{;}, \mathbb{B}*, b_0) = \frac{1}{N}\sum_{i=1}^{N} V\left(\tilde{\mu}_i + y_i\mathbf{X}_i^{(1)}[j, r](\mathbf{U}_1[j, r] - \mathbf{U}_1[j, r])\right) + P_{\lambda_1, \lambda_2}(\mathbb{B}*),$$

and we approximate $h(\mathbf{y}, \mathbb{X}_{;}\mathbb{B}*, b_0)$ with a quadratic form analogous to that in Wang and Zou (2016),

$$\frac{1}{N}\sum_{i=1}^{N} V(\mu_i) + \frac{1}{N}\sum_{i=1}^{N} V'(\tilde{\mu}_i)(\mathbf{U}_1[j, r] - \mathbf{U}_1[j, r]) + 2(\mathbf{U}_1[j, r] - \mathrm{U}_1[j, r])^2 \\ + P_{\lambda_1, \lambda_2}(\mathbb{B}*) . \tag{7}$$

Note that

$$P_{\lambda_1, \lambda_2}(\mathbb{B}*) = \lambda_1 q_r|\mathbf{U}_1[j, r]| + \frac{\lambda_2}{2}\left(\mathbf{W}[r, r]\mathbf{U}_1[j, r]^2 + \sum_{r' \neq r}^{R} \mathbf{W}[r, r']\mathbf{U}_1[j, r']\right) + C,$$

for a value C that is constant with respect to $\mathbf{U}_1[j, r]$. Thus, the minimizer of (7) over $\mathbf{U}_1[j, r]$ is given by $\mathbf{U}_1[j, r]^{\text{new}}$ in Step b. A similar justification is used for the update of the intercept in Step c.

Step a: Compute $\mathbf{X}_i^{(1)}: P_1 \times R$ with columns defined as in (6) for each rank component $r = 1, \ldots, R$:

$$\mathbf{X}_i^{(1)}[p_1, r] = \mathbb{X}_i[p_1, \ldots] \cdot (\mathbf{u}_{2r} \circ \cdots \circ \mathbf{u}_{Kr})$$

and let $\tilde{\mu}_i = y_i\left(\tilde{b}_0 + \mathbf{U}_1 \cdot \mathbf{X}_i^{(1)}\right)$ for $i = 1, \ldots, N$ (note: $\mathbf{U}_1 \cdot \mathbf{X}_i^{(1)} = \mathbb{X}_i \cdot \mathbb{B}$). Define $\mathbf{W}$: $R \times R$ by $\mathbf{W} = \left(\mathbf{U}_2^T\mathbf{U}_2\right) \cdots \left(\mathbf{U}_K^T\mathbf{U}_K\right)$.

Step b: Update $\mathbf{U}_1$ via the MM principle and cyclic coordinate decent over $j = 1, \ldots, P_1$ and $r = 1, \ldots, R$.

- Compute $z = 4\mathbf{U}_1[j, r] - \frac{1}{N}\sum_{i=1}^N V'(\tilde{\mu}_i)\mathbf{X}_i^{(1)}[j, r]y_i - \lambda_2\sum_{r' \neq r}^R \mathbf{W}[r, r']\mathbf{U}_1[j, r']$, where V' is the derivative of V in (2).
- Compute $\mathbf{U}_1[j, r]^{\text{new}} = \frac{S(z, \lambda_1 q_r)}{4 + \lambda_2\mathbf{W}[r, r]}$, where $q_r = \prod_{k=2}^K \|\widetilde{U}_{kr}\|_1$ is the weight for the L_1 penalty, and $S(z, g) = sign(z)(|z| - g)_+$ is the soft-thresholding operator in which $\omega_+ = max(\omega, 0)$.
- Update $\tilde{\mu}_i = \mu_i + y_i\left(\mathbf{U}_1[j . r]^{\text{new}} - \mathbf{U}_1[j, r]\right)\mathbf{X}_i[j, r]$ for $i = 1, \ldots, N$.
- Set $\mathbf{U}_1[j, r] = \mathbf{U}_1[j, r]^{\text{new}}$.

Step c: Update the intercept.

- Compute $\tilde{b}_0^{\text{new}} = \tilde{b}_0 - \sum_{i=1}^N V'(\tilde{\mu}_i)y_i/(4N)$
- Update $\tilde{\mu}_i = \tilde{\mu}_i + y_i\left(\tilde{b}_0^{\text{new}} - \tilde{b}_0\right)$
- Set $\tilde{b}_0 = \tilde{b}_0^{\text{new}}$.

Step d: Repeat steps b-c until the difference between new estimates and previous estimates is smaller than a prespecified threshold, for example, $\sum_{j=1}^{P_1}\sum_{r=1}^R \left(\mathbf{U}_1[j, r]^{\text{new}} - \mathbf{U}_1[j, r]\right)^2 + \left(\tilde{b}_0^{\text{new}} - \tilde{b}_0\right)^2 < \epsilon$. At the end of iterations, we have the updated weight matrix $\mathbf{U}_1$.

Algorithm 2: Update for $\mathbf{U}_1$ when $R \geq 2$.

In general, Algorithm 1 does not guarantee convergence to a global optimum; it converges to a coordinate-wise optimum, meaning the objective cannot be improved by changing any of the components $\mathbf{U}_1, \mathbf{U}_2, \ldots, \mathbf{U}_K$ or b_0 while keeping the others fixed. We conducted simulation studies to assess the convergence of the proposed method and other alternative methods in Section 5.2, and in Section S4 of the supplementary material. In practice, we find that starting with multiple initial values and pruning the paths with inferior objective values can improve the algorithm and obtain more robust results.

4.4 Selection of tuning parameters

To determine the best pair of λ_1 and λ_2, K-fold cross-validation can be used across a grid of λ_1 and λ_2 values. For example, by default we set the following candidates for the tuning parameters: $\lambda_1 = (10^{-4}, 0.001, 0.005, 0.01, 0.025, 0.05, 0.1, 0.25, 0.5, 0.75, 1)$ and $\lambda_2 = (0.25, 0.50, 0.75, 1.00, 3, 5)$. The misclassification rate under cross-validation is one potential criterion; however, different parameter values may yield the same misclassification rate, especially if the classes are perfectly separated or nearly perfectly separated (see Section S7). We thus recommend using t-test statistics for the DWD scores between the two classes on the validation sets as a more general measure of separation for selecting tuning parameters. For each pair of λ_1 and λ_2 we compute the predicted DWD scores for all subjects via K-fold cross-validation, and then report t-test statistics for the difference of the predicted DWD scores between the two classes using the test-data. We selected as optimal the pair of λ_1 and λ_2 with the maximum t-test statistic. For a fixed λ_2, the warm-start trick is used in the algorithm to select the optimal λ_1. That is, we use the solution at a smaller λ_1 as the initial value (the warm-start) to compute the solution at the next λ_1, improving computational efficiency. The performance of cross-validation for selecting λ_1 an λ_2 is evaluated through simulation studies presented in the supplementary material (Section S1). If the rank R is unknown it can also be estimated via cross-validation, e.g., by selecting the triplet $(\lambda_1, \lambda_2, R)$ with the largest t-statistic on the validation sets; this is evaluated in Section S6 of the supplementary material.

5 Simulations

In our simulation studies, we first compare the proposed rank-1 multiway sparse model with the existing methods in Section 5.1. In Section 5.2 we explore convergence and its impact on performance. In Section 5.3, we show the rank-R sparse model performs well when the true data generating model has a higher rank. Additional simulation studies are presented in the supplementary material, including simulations to assess cross-validation for parameter selection, rank misspecification, correlated predictors, and different penalization approaches.

5.1 Rank-1 model simulation design

To evaluate the performance of the rank-1 multiway sparse DWD model (M-SDWD), we compared it with a rank-1 multiway sparse DWD model with $\lambda_1 = 0$ (M-SDWD $\lambda_1 = 0$), non-sparse multiway DWD (M-DWD), and the full model sparse DWD (Full SDWD). For M-DWD we use an extended version of multiway DWD (Lyu et al. 2017) that allows for data of any dimension $K > 2$. The full model sparse DWD is a naive way to analyze multiway data by applying the standard sparse DWD (Wang and Zou 2016) to the vectorized multiway data, without considering multiway structure (i.e., no rank constraint). In this simulation study, data were generated under several conditions, including different multiway array dimensions, sample sizes, and sparsity levels. For all scenarios, training datasets with two classes of equal size ($N_0 = N_1 = N / 2$, $N = 40$ or 100) were generated. The predictors have the form of a three-way array of dimensions $\mathbb{X}_i: P_1 \times P_2 \times P_3$. We consider two settings of different dimensionality: higher dimensional ($30 \times 15 \times 15$) and lower dimensional ($15 \times 4 \times 5$). In each training dataset, for the N_0 samples corresponding to class -1, the entries of $\mathbb{X}_i$ were generated independently from a $N(0, 1)$ distribution. For the other N_1 samples corresponding to class 1, the entries of $\mathbb{X}_i$ were generated independently from a normal distribution with variance 1 and the mean for each entry given by the array $\sqrt{\alpha}\mu_1$ where $\mu_1 = \mathbf{u}_1 \circ \mathbf{u}_2 \circ \mathbf{u}_3$; the non-zero values of $\mathbf{u}_k$, $k = 1, 2, 3$, were generated independently from a $N(0, 1)$ distribution. Here, α gives the signal to noise ratio, i.e., the variance of the mean difference between the two classes over the residual variance. We set $\alpha = 0.2$ for this simulation study. Section S3 of the supplementary material presents results with different signal to noise ratios. Section S7 presents results under a multiway residual covariance structure assumed by the CATCH method (Pan et al. 2018), with comparisons to CATCH.

We consider three scenarios with varying degrees of sparsity: more sparsity, less sparsity and no sparsity. Under the more sparsity scenario, about 1/3 of the mean weights in each dimension ($\{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\}$) are set to zero. Specifically, under the high dimensional case (30 × 15 × 15), 25 entries of $\mathbf{u}_1$, and 10 entries each of $\mathbf{u}_2$ and $\mathbf{u}_3$ are set to zero, that is, 5 ×5 × 5 of the variables have signal discriminating the classes; under the low dimensional case (15 × 4 × 5), only 5 entries of $\mathbf{u}_1$, and 2 entries each of $\mathbf{u}_2$ and $\mathbf{u}_3$ are non-zero. Under the less sparsity scenario, sparsity is considered for only one dimension. In particular, 10 entries of $\mathbf{u}_1$ under both the high and low dimensional cases are set to zero, and the rest of the $\mathbf{u}_k$ are non-zero. Under the no sparsity case, all variables for each $\mathbf{u}_k$ are nonzero. For the high dimensional case we also considered an additional case with even more sparsity in the model: only 3 variables of each dimension have signals and the rest of variables for each dimension are zero. Each scenario was replicated 200 times.

Under all these scenarios, as a scale-invariant measure of similarity with the true discriminating signal we computed the sample Pearson correlation between the vectorized estimated coefficients $\mathbb{B}$ and the vectorized “true” coefficients μ_1. Here, μ_1 is the mean difference between the classes and corresponds to the Bayes linear classifier. We assess predictive performance by considering misclassification rates for test data that were generated from the same distributions as the training data with the same sample sizes ($N = 40$ or 100). To assess recovery of the sparsity structure, we computed the true positive rates and true negative rates for the proportions of non-zero/zero weights that were correctly estimated. All these statistics were computed based on the vectorized weights for multi-way based methods. The margin of error across the replicates for each statistic was also computed.

Table 1 and Table 2 summarize the simulation results for lower and higher dimensional data, respectively. Both tables show the proposed rank-1 multiway sparse DWD model has the best performance, with higher correlation and lower misclassification rates than other methods, when the true model is more sparse. When the true model is less sparse or not sparse, the proposed method has comparable performances to multiway DWD (M-DWD). All multiway based methods have higher correlations and lower misclassification rates than the full sparse DWD model under different scenarios. Overall the proposed rank-1 multiway sparse DWD model performs well, but some of the advantages are not obvious. In particular, the mean correlations are relatively low for the very sparse models, since the correlations with the truth are very small for some simulation replications. This is because the algorithm may converge to local minima, which is explored further in Section 5.2.

5.2 Assessment of convergence

To assess the convergence of the four methods, we show the distributions of correlations between the true and estimated hyperplane for the “more sparsity” scenario for high dimensional data in Figure 1. The distribution is bimodal for the three multiway methods, where for some replications the signal is estimated very well (correlation ≈ 1) and others do not recover the true signal at all (correlation ≈ 0). We find that the poor performing replications are due to the algorithm converging to a local optimum, which does not occur for the full (vectoriozed) SDWD method. The proposed multiway sparse DWD (M-SDWD) performs better than other methods with more reasonable correlations. In addition, the bimodality becomes less severe when the signal to noise ratio increases (see Section S4 of the Supplementary Material). Considering prediction performance without separating out the replications that do not converge appropriately may complicate interpretation of the results, hense we present another two tables that show those simulations with correlation greater than 0.5 for each method, and the statistics among those simulations with correlation greater than 0.5 for that method. From Table 3 and Table 4 we can see the proposed method performs the best under the “even more” and “more” sparsity cases. To reduce convergence to non-optimal solutions, we recommend running the algorithm with multiple initial values for the first several iterations of the algorithm and then selecting the optimal path to proceed; by default our implementation randomly generates 5 sets of initial values, runs Algorithm 1 for 10 updating cycles over all parameters for each set of initial values, and selects the set that results in the lowest value of the objective function after 10 cycles to proceed with estimation until convergence.

5.3 Rank-R model simulation and results

In this simulation study, we aim to validate the higher rank multiway sparse model with a scenario in which $R = 2$. The procedure for data generation is similar to that of the rank-1 model. Instead of generating vectors $\mathbf{u}_k$ for each dimension, we generated three matrices $\mathbf{U}_k$, $k = 1, 2, 3$ with dimensions $P_k \times 2$. The entries of $\mathbf{U}_k$ are generated independently from a normal distribution $N(0, 1)$, then we compute $\mu_1 = [[\mathbf{U}_1, \ldots, \mathbf{U}_K]]$. The N_1 samples corresponding to class 1 were generated from a multivariate normal distribution $N(\mu_1, \mathbf{I}_{P_1P_2P_3 \times P_1P_2P_3})$. The N_0 samples in class −1 were generated independently from a $N(0, 1)$ distribution. Different scenarios are considered in terms of dimensions, sample size and sparsity levels. The results in Table 5 show that the rank-2 model has the best performance when the data were generated from a true rank-2 model under different scenarios. The performances of these methods are similar for a higher rank (R=5) and low dimensional case (see Section S5 of the Supplementary Material).

6 Applications

6.1 MRS data

We apply the proposed method to our motivating application with MRS data to illustrate its utility. Friedreich’s ataxia (FRDA) is an early-onset neurodegenerative disease caused by abnormalities in the frataxin gene (Pandolfo 2008) resulting in knockdown of frataxin protein. The motivation of this study is to assess brain metabolic changes in a transgenic mouse model in which frataxin knockdown can be turned on by giving the antibiotic doxycycline (Chandran et al. 2017). Wild type (WT) mice and transgenic (TG) mice were randomly assigned to two different treatment groups: doxycycline treated group (N_1 = 11) and controls (N_0 = 10). Thus, the study included 4 groups of mice: WT treated with dox (WT-dox), WT without dox (WT-nodox), TG treated with dox (TG-dox), TG without dox (TG-nodox). Treatment was applied during weeks 1–12, then stopped during weeks 12–24 (recovery). Frataxin knockdown during weeks 1–12 was expected only in the TG-dox group, but not in WT animals nor in animals not receiving dox. The concentrations of 13 metabolites were measured in three different regions (cerebellum, cortex, and cervical spine). These animals were scanned at three time points: 0 weeks, 12 weeks and 24 weeks. The data have a multiway structure with four dimensions: mice × metabolites × regions × time, $\mathbb{X}: 21 \times 13 \times 3 \times 3$. Our goal is to summarize the signal distinguishing the two treatment groups (dox vs. no dox) across the different dimensions, and assess how the effect of treatment depends on WT/TG status.

The penalty parameters for multiway sparse DWD were selected through 10-fold cross validation (see Section S9 of the supplementary material), and the rank-1 model was selected. Using the selected tuning parameters, Figure 2 shows the DWD scores computed under leave-one-out cross-validation based on the rank-1 multiway sparse DWD model. These scores show robust separation of the no-dox and dox groups (which correspond to the two classes $y_i = -1$ and $y_i = 1$) but only for the transgenic (TG) mice. The three scores for the TG-nodox, WT-nodox, and WT-dox groups are very similar. Because we train the classifier on both genetic cohorts but only observe a difference in the transgenic mice, this confirms the hypothesis that wild type mice are not affected by treatment; further, the results illustrate the robustness of the approach to within-class heterogeneity.

In order to capture the uncertainty of the estimated weights and construct 95% confidence intervals, 500 bootstrap samples were generated. For each bootstrap sample, 21 mice were resampled with replacement, and then the model was fit to the bootstrap sample to estimate weights for each dimension. The 2.5% and 97.5% quantiles for each estimated weight across the 500 bootstrap samples were computed to construct the 95% confidence interval. Figure 3 shows the estimated weights and their 95% bootstrap confidence intervals. The metabolites with large absolute weights, such as PCho+GPC, Cr+PCr, and Ins, are considered important for Friedreich's ataxia research (França et al. 2009; Iltis et al. 2010; Gramegna et al. 2017), so were expected to have high weights for distinguishing the two groups. Two metabolites (Lac and Glu) and one time point (0 weeks) did not inform the classification as their estimated weights are exactly 0 in the full data fit. Having no effect at baseline (0 weeks) makes sense as this is prior to the dox treatment. The observed changes at the later time points are similar to those observed in the R6/2 mouse, a severe mouse model of Huntington's disease (Zacharoff et al. 2012). This is consistent with the fact that both Friedreich's ataxia and Huntington's disease are characterized by impairment in energy metabolism.

We compared misclassification rates for the M-SDWD, M-SDWD with $\lambda_1 = 0$, M-DWD, CATCH, Full SDWD and Random Forests by 10-fold cross validation. The 21 mice were randomly partitioned into 10 test subgroups of approximately equal size. Tuning parameters were estimated within each training set by an inner cross-validation procedure. This procedure was repeated 100 times with different random partitions and the average test misclassification rates for dox vs. no dox were as follows: 22.1% for M-SDWD, 20.9% for M-SDWD with $\lambda_1 = 0$, 23.6% for M-DWD, 22.3% for CATCH, 42.9% for Full SDWD, and 22.3% for Random Forest. To better reflect the reality of the treatment effect, we further considered the misclassification rate for TG-dox vs. the other three groups under this approach: 3.1% for M-SDWD, 1.9% for M-SDWD with $\lambda_1 = 0$, 4.6% for M-DWD, 4.0% for CATCH, 24.6% for Full SDWD, and 10.5% for Random Forest. Overall, the multiway methods perform better than the non-multiway methods (Full SDWD and Random Forest), illustrating the advantages of accounting for multiway structure. Moreover, the factorization of weights across dimensions in Figure 3 is useful for interpretation, and is not provided by the CATCH model. We conclude that the performance of M-SDWD model is competitive with alternatives. The rank-2 multiway sparse model was also applied, resulting in a higher misclassification rate (29.9% for dox vs. no dox and 10.8% for TG-dox vs. the other three groups), indicating that the rank-1 model is sufficient. Results for the rank-1 model were the same over 10 runs of the algorithm with random initializations.

6.2 Gene time course data

We further applied multiway sparse DWD to the gene expression time course data described in Baranzini et al. (2004). The purpose of this study is to classify clinical response to treatment for multiple sclerosis (MS) patients. Recombinant human interferon beta (rIFNβ) was given to 53 patients for controlling the symptoms of MS. For each patient, gene expression was measured for 76 genes at 7 time points: baseline (i.e. before treatment) and 6 follow-up time points (3 months, 6 months, 9 months, 12 months, 18 months, and 24 months). The data are a multi-way array with 3 dimensions: patients × genes × time points, $\mathbb{X}: 53 \times 76 \times 7$. Based on clinical characteristics, each patient was designated as a good responder or poor responder to rIFNβ. The proposed multiway sparse DWD model was used to differentiate good responders from poor responders, and a rank-1 model was selected by 10-fold cross-validation. Figure 4 shows the DWD scores under leave-one-out cross validation for the rank-1 multiway model. The groups of these good and poor responders are nearly perfectly separated. The coefficient estimates and 95% bootstrap confidence intervals for each gene and each time point are shown in Figure 5. The coefficients across time had little variability, which suggests that the distinction between good and poor responders is not driven by effects that vary over the time course. This agrees with the results reported in Baranzini et al. (2004) where they found there was no group*time interaction effects. Note that the data are not very sparse as almost all estimated coefficients for gene and time are not zero. We also compared the performance of the rank-1 model with other competitors. The mean misclassification rates were 13.0% for M-SDWD, 12.6% for M-SDWD with $\lambda_1 = 0$, 17.4% for M-DWD, 20.8% for CATCH, 20.5% for Full SDWD and 31.5% for Random Forest. The rank-1 multiway sparse models have the best classification over other methods. It is notable that the multiway sparse DWD model outperforms multiway DWD even when the degree of sparsity in the data is low. We also considered the rank-2 multiway sparse model to classify good and poor responders. A rank-2 model performs worse than the rank-1 model with a higher misclassification rate (18.7 %), which agrees with the results in Lyu et al. (2017). Results for the rank-1 model were the same over 10 runs of the algorithm with random initializations.

7 Discussion

We have proposed a general framework for high dimensional classification on multiway data with any number of dimensions, which can account for sparsity in the model. Both the simulation and data analysis results have shown that the proposed multiway sparse DWD model can improve classification accuracy when the underlying signal is sparse or has multiway structure. The proposed method is robust to any degree of sparsity in the model, which has been demonstrated in both simulation and applications. Moreover, the method is robust to the complexity of the shared signal across the different dimensions, by allowing for higher rank models. Lastly, the use of multiway structure and sparsity can facilitate and simplify interpretation. For our motivating application in Section 6.1, a single application of multiway sparse DWD provided the following insights: (1) TG mice show a distinct brain metabolomic profile after receiving doxycycline but WT mice do not, (2) these changes to the metabolite profile are similar across three neurological regions but most pronounced in the cerebellum, and (3) these changes are most prominent at the end of 12 weeks of treatment, but subside post-treatment.

Despite the flexibility of multiway sparse DWD, the results are sensitive to the choice of rank and tuning parameters. The simulation in Section 5.1 shows that when the model is not sparse, the rank-1 model with $\lambda_1 = 0$ performs better than the rank-1 model with λ_1 selected by cross validation; thus, the sparsity penalty only improves results when the signal is truly sparse. Also, the rank-1 model performs poorly when the true model has higher rank, as expected. For certain applications these choices may depend on domain-specific knowledge and goals, e.g., whether a sparse ($\lambda_1 > 0$) or higher rank ($R > 1$) solution makes sense in context; otherwise, we suggest selecting these parameters by cross-validation. Also, as discussed in Sections 4.3, 5.2, and the Supplement S4, algorithms using cyclic coordinate descent, including ours, may not converge to a global optimum. In practice, we suggest running the algorithm from multiple starting values to assess the stability of the results and (if results are sensitive to starting values) select the solution that yields the smallest value for the objective function. We have prioritized computational efficiency in our implementation. For the applications in Section 6 computing time ranged from less than a second for a single run of the algorithm, to 10 minutes using multiple random starts and cross-validation to select tuning parameters (see Supplement S10). There are many approaches to penalization in the sparse multiway context, and while our chosen penalty has advantages with respect to its flexibility, interpretation and feasibility, other possibilities discussed in Section 4.2 are worth pursuing in more depth. Moreover, in this article we only consider binary classification, and extensions to multi-category classification (e.g., as in Huang et al. (2013)) is another worthwhile future direction.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgements

This work was supported in part by the National Institutes of Health (NIH) grant R01GM130622. CMRR (P.G.H. and C.L.) is partly supported by NIH grants P41 EB027061 and P30 NS076408. The study on Friedreich's Ataxia mice was supported by the Friedreich's Ataxia Research Alliance (FARA) and the CureFA Foundation. We acknowledge and thank UCLA and Drs. Vijayendran Chandran and Daniel Geschwind for providing the inducible frataxin knock down mouse model, with which the MRS data was obtained and used in this manuscript to illustrate the performance of the proposed algorithm (Section 6.1).

References

Baranzini SE, Mousavi P, Rio J, Caillier SJ, Stillman A, Villoslada P, Wyatt MM, Comabella M, Greller LD, Somogyi R et al. (2004), 'Transcription-based prediction of response to ifnβ using supervised computational methods', PLoS Biol 3(1), e2. [PubMed: 15630474]

Bauckhage C (2007), Robust tensor classifiers for color object recognition, *in* 'International Conference Image Analysis and Recognition', Springer, pp. 352–363.

Bradley PS and Mangasarian OL (1998), Feature selection via concave minimization and support vector machines., in 'ICML', Vol. 98, pp. 82–90.

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

Bro R (1997), 'Parafac. tutorial and applications', Chemometrics and intelligent laboratory systems 38(2), 149–171.

Cai T and Liu W (2011), 'A direct estimation approach to sparse linear discriminant analysis', Journal of the American statistical association 106(496), 1566–1577.

Chandran V, Gao K, Swarup V, Versano R, Dong H, Jordan MC and Geschwind DH (2017), 'Inducible and reversible phenotypes in a novel mouse model of friedreich's ataxia', Elife 6, e30054. [PubMed: 29257745]

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

Cover T and Hart P (1967), 'Nearest neighbor pattern classification', IEEE transactions on information theory 13(1), 21–27.

Fan J and Fan Y (2008), 'High dimensional classification using features annealed independence rules', Annals of statistics 36(6), 2605. [PubMed: 19169416]

Fan J, Feng Y and 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. [PubMed: 23074363]

França MC, D'Abreu A, Yasuda CL, Bonadia LC, Da Silva MS, Nucci A, Lopes-Cendes I and Cendes F (2009), 'A combined voxel-based morphometry and 1 h-mrs study in patients with friedreich's ataxia', Journal of neurology 256(7), 1114–1120. [PubMed: 19280106]

Gramegna LL, Tonon C, Manners DN, Pini A, Rinaldi R, Zanigni S, Bianchini C, Evangelisti S, Fortuna F, Carelli V et al. (2017), 'Combined cerebellar proton mr spectroscopy and dwi study of patients with friedreich's ataxia', The Cerebellum 16(1), 82–88. [PubMed: 26897753]

Harshman RA (1970), 'Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis', UCLA Working Papers in Phonetics 16, 1–84.

Hastie T, Tibshirani R and Friedman J (2009), The elements of statistical learning: data mining, inference, and prediction, Springer Science & Business Media.

Huang H, Liu Y, Du Y, Perou CM, Hayes DN, Todd MJ and Marron JS (2013), 'Multiclass distance-weighted discrimination', Journal of Computational and Graphical Statistics 22(4), 953–969.

Hunter DR and Lange K (2004), 'A tutorial on mm algorithms', The American Statistician 58(1), 30–37.

Iltis I, Hutter D, Bushara KO, Clark HB, Gross M, Eberly LE, Gomez CM and Öz G (2010), '1h mr spectroscopy in friedreich's ataxia and ataxia with oculomotor apraxia type 2', Brain research 1358, 200–210. [PubMed: 20713024]

Liu Y, Zhang HH and Wu Y (2011), 'Hard or soft classification? Large-margin unified machines', Journal of the American Statistical Association 106(493), 166–177. [PubMed: 22162896]

Lock EF (2018), ‘Tensor-on-tensor regression’, Journal of Computational and Graphical Statistics 27(3), 638–647. [PubMed: 30337798]

Lyu T, Lock EF and Eberly LE (2017), ‘Discriminating sample groups with multi-way data’, Biostatistics 18(3), 434–450. [PubMed: 28115314]

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

Marron JS, Todd MJ and Ahn J (2007), ‘Distance-weighted discrimination’, Journal of the American Statistical Association 102(480), 1267–1271.

Pan Y, Mai Q and Zhang X (2018), ‘Covariate-adjusted tensor classification in high dimensions’, Journal of the American statistical association.

Pandolfo M (2008), ‘Friedreich Ataxia’, Archives of Neurology 65(10), 1296–1303. [PubMed: 18852343]

Tao D, Li X, Hu W, Maybank S and Wu X (2005), Supervised tensor learning, in ‘Fifth IEEE International Conference on Data Mining (ICDM’05)’, IEEE, pp. 8–pp.

Wang B and Zou H (2015), sdwd: Sparse Distance Weighted Discrimination. R package version 1.0.2.

Wang B and Zou H (2016), ‘Sparse distance weighted discrimination’, Journal of Computational and Graphical Statistics 25(3), 826–838.

Wang L, Zhu J and Zou H (2008), ‘Hybrid huberized support vector machines for microarray classification and gene selection’, Bioinformatics 24(3), 412–419. [PubMed: 18175770]

Wimalawarne K, Tomioka R and Sugiyama M (2016), ‘Theoretical and experimental analyses of tensor-based regression and classification’, Neural computation 28(4), 686–715. [PubMed: 26890354]

Witten DM and Tibshirani R (2011), ‘Penalized classification using Fisher’s linear discriminant’, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 73(5), 753–772. [PubMed: 22323898]

Ye J, Janardan R and Li Q (2004), ‘Two-dimensional linear discriminant analysis’, Advances in neural information processing systems 17, 1569–1576.

Zacharoff L, Tkac I, Song Q, Tang C, Bolan PJ, Mangia S, Henry P-G, Li T and Dubinsky JM (2012), ‘Cortical metabolites as biomarkers in the r6/2 model of huntington’s disease’, Journal of Cerebral Blood Flow & Metabolism 32(3), 502–514. [PubMed: 22044866]

Zhang HH, Ahn J, Lin X and Park C (2006), ‘Gene selection using support vector machines with non-convex penalty’, Bioinformatics 22(1), 88–95. [PubMed: 16249260]

Zou H (2019), ‘Classification with high dimensional features’, Wiley Interdisciplinary Reviews: Computational Statistics 11(1), e1453.

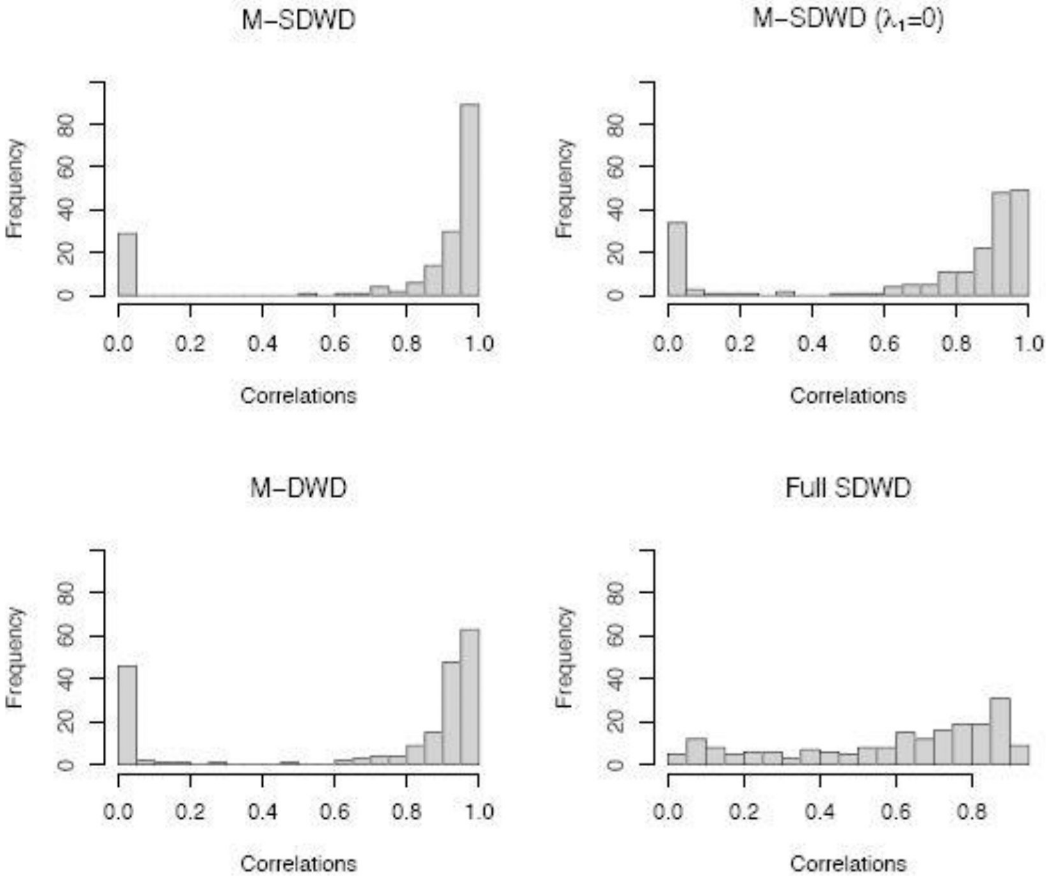

Fig. 1.
Histogram of correlations between true hyperplane and estimates with four classification methods under the more sparsity case based on high dimensional data (30 × 15 × 15) and sample size $N = 100$.

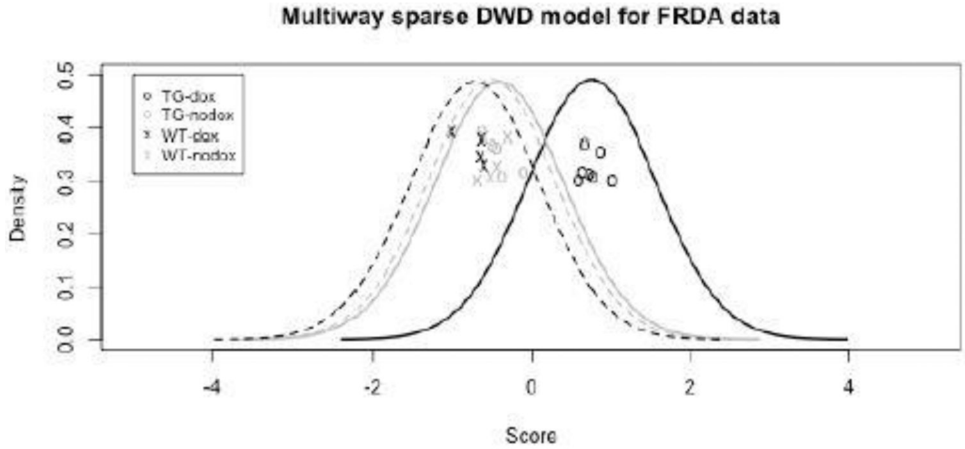

Fig. 2.

FRDA mouse study: Rank-1 multiway sparse DWD scores under leave-one-out cross-validation to classify mice that did or did not receive dox treatment. The transgenic (TG) mice that received dox are clearly distinguished from the TG mice that did not and from the wild-type (WT) mice. A kernel density estimate is shown for the scores in each subgroup.

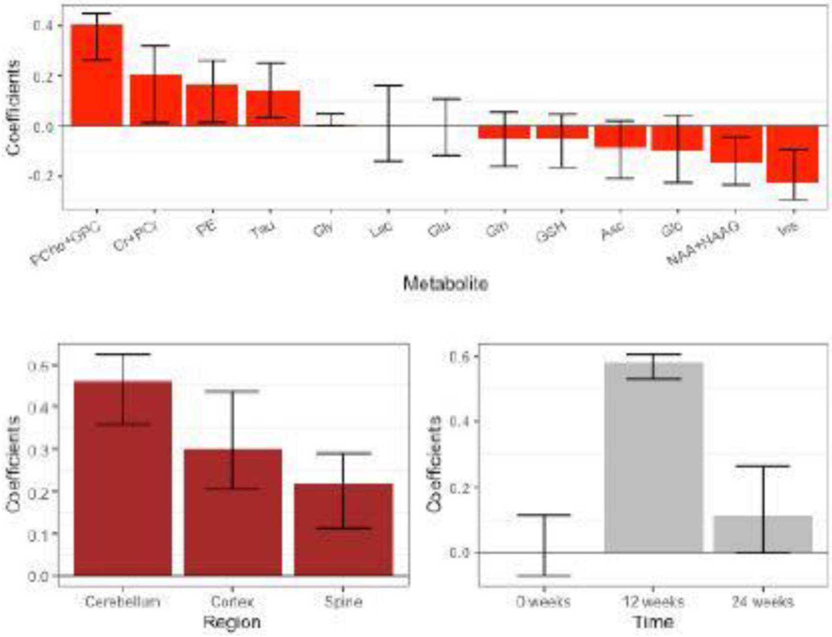

Fig. 3.

FRDA mouse study: Rank-1 multiway sparse DWD weights for metabolites, regions and time with 95% bootstrap confidence intervals.

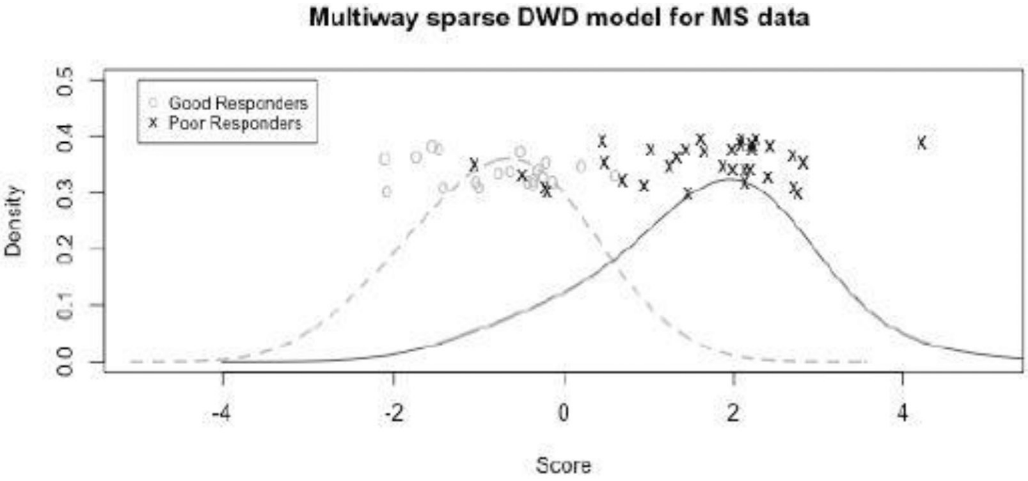

Fig. 4.

MS treatment study: Rank-1 multiway sparse DWD scores under leave-one-out cross-validation for good and poor treatment responders, with a kernel density estimate for each group.

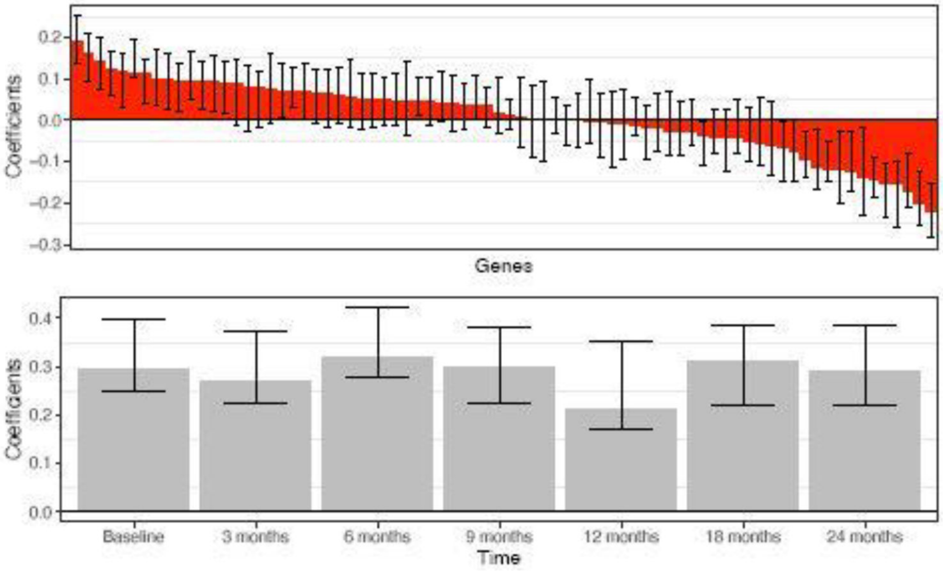

Fig. 5.

MS treatment study: Rank-1 multiway sparse DWD weights for genes and time with 95% bootstrap confidence intervals.

Table 1

Simulation results under the low dimensional scenario (15 × 4 × 5). In the Sparsity column, the numbers in parentheses indicate the number of non-zero variables in each dimension. “Cor” is the correlation between the estimated linear hyperplane and the true hyperplane. “Mis” is the average misclassification rate. “TP” is the true positive rate, i.e., the proportion of non-zero coefficients that are correctly estimated to be non-zero.“TN” is the true negative rate, i.e., the proportion of zero coefficients that are correctly estimated to be zero. The margins of error (2* standard errors across 200 replicates) for each statistic are also listed following the ± symbol.

N	Sparsity	Methods	Cor	Mis	TP	TN
40	More (5 × 2 × 2)	M-SDWD	**0.359±0.054**	0.390±0.025	0.492±0.057	0.644±0.056
		M-SDWD ($\lambda_1 = 0$)	0.318±0.049	**0.386±0.025**	1.000±0.000	0.000±0.000
		M-DWD	0.313±0.049	0.388±0.026	1.000±0.000	0.000±0.000
		Full SDWD	0.284±0.039	0.402±0.023	0.337±0.041	0.748±0.043
	Less (5 × 4 × 5)	M-SDWD	**0.773±0.034**	0.156±0.024	0.646±0.045	0.527±0.053
		M-SDWD ($\lambda_1 = 0$)	0.768±0.035	**0.146±0.022**	1.000±0.000	0.000±0.000
		M-DWD	0.757±0.042	0.154±0.024	1.000±0.000	0.000±0.000
		Full SDWD	0.532±0.031	0.214±0.024	0.320±0.038	0.776±0.039
	No (15 × 4 × 5)	M-SDWD	0.894±0.025	0.052±0.015	0.792±0.038	-
		M-SDWD ($\lambda_1 = 0$)	**0.901±0.026**	**0.048±0.014**	1.000±0.000	-
		M-DWD	0.896±0.030	0.054±0.017	1.000±0.000	-
		Full SDWD	0.639±0.026	0.096±0.020	0.444±0.045	-
100	More (5 × 2 × 2)	M-SDWD	**0.574±0.056**	0.332±0.025	0.536±0.055	0.713±0.051
		M-SDWD ($\lambda_1 = 0$)	0.489±0.055	**0.329±0.025**	1.000±0.000	0.000±0.000
		M-DWD	0.489±0.056	0.331±0.025	1.000±0.000	0.000±0.000
		Full SDWD	0.448±0.049	0.337±0.024	0.318±0.037	0.841±0.034
	Less (5 × 4 × 5)	M-SDWD	**0.902±0.022**	0.112±0.018	0.720±0.038	0.551±0.050
		M-SDWD ($\lambda_1 = 0$)	0.890±0.025	0.113±0.018	1.000±0.000	0.000±0.000
		M-DWD	0.898±0.027	**0.111±0.018**	1.000±0.000	0.000±0.000
		Full SDWD	0.695±0.028	0.156±0.021	0.357±0.034	0.794±0.033
	No (15 × 4 × 5)	M-SDWD	0.954±0.014	**0.035±0.012**	0.870±0.031	-
		M-SDWD ($\lambda_1 = 0$)	0.963±0.012	**0.035±0.011**	1.000±0.000	-
		M-DWD	**0.965±0.014**	**0.035±0.012**	1.000±0.000	-
		Full SDWD	0.782±0.021	0.061±0.015	0.433±0.036	-

Table 2

Simulation results under the high dimensional scenario (30 × 15 × 15). In the Sparsity column, the numbers in parentheses indicate the number of non-zero variables in each dimension. “Cor” is the correlation between the estimated linear hyperplane and the true hyperplane. “Mis” is the average misclassification rate. “TP” is the true positive rate, i.e., the proportion of non-zero coefficients that are correctly estimated to be non-zero.“TN” is the true negative rate, i.e., the proportion of zero coefficients that are correctly estimated to be zero. The margins of error (2* standard errors across 200 replicates) for each statistic are also listed following the ± symbol.

N	Sparsity	Methods	Cor	Mis	TP	TN
40	Even more (3 × 3 × 3)	M-SDWD	0.214±0.053	0.426±0.025	0.077±0.012	0.738±0.054
		M-SDWD ($\lambda_1 = 0$)	0.154±0.042	0.415±0.025	0.216±0.000	0.000±0.000
		M-DWD	0.171±0.043	0.404±0.025	0.216±0.000	0.000±0.000
		Full SDWD	**0.233±0.041**	**0.388±0.024**	0.039±0.008	0.900±0.035
	More (5 × 5 × 5)	M-SDWD	**0.644±0.056**	0.218±0.033	0.489±0.052	0.778±0.047
		M-SDWD ($\lambda_1 = 0$)	0.516±0.056	**0.216±0.031**	1.000±0.000	0.000±0.000
		M-DWD	0.505±0.058	0.236±0.033	1.000±0.000	0.000±0.000
		Full SDWD	0.389±0.039	0.238±0.027	0.129±0.026	0.932±0.026
	Less (10 × 15 × 15)	M-SDWD	0.983±0.002	0.000±0.000	0.872±0.026	0.394±0.060
		M-SDWD ($\lambda_1 = 0$)	0.988±0.001	0.000±0.000	1.000±0.000	0.000±0.000
		M-DWD	**0.990±0.001**	0.000±0.000	1.000±0.000	0.000±0.000
		Full SDWD	0.654±0.016	0.002±0.002	0.143±0.019	0.943±0.017
	No (30 × 15 × 15)	M-SDWD	0.987±0.003	0.000±0.000	0.869±0.028	-
		M-SDWD ($\lambda_1 = 0$)	0.994±0.001	0.000±0.000	1.000±0.000	-
		M-DWD	**0.997±0.000**	0.000±0.000	1.000±0.000	-
		Full SDWD	0.724±0.010	0.000±0.000	0.262±0.032	-
100	Even more (3 × 3 × 3)	M-SDWD	0.140±0.045	0.467±0.016	0.018±0.004	0.784±0.049
		M-SDWD ($\lambda_1 = 0$)	0.084±0.032	0.464±0.016	0.064±0.000	0.000±0.000
		M-DWD	0.101±0.035	0.455±0.017	0.064±0.000	0.000±0.000
		Full SDWD	**0.218±0.047**	**0.431±0.020**	0.018±0.003	0.868±0.038
	More (5 × 5 × 5)	M-SDWD	**0.849±0.038**	**0.089±0.020**	0.668±0.040	0.806±0.041
		M-SDWD ($\lambda_1 = 0$)	0.796±0.040	0.101±0.021	1.000±0.000	0.000±0.000
		M-DWD	0.766±0.049	0.121±0.025	1.000±0.000	0.000±0.000
		Full SDWD	0.636±0.035	0.136±0.022	0.172±0.023	0.957±0.022
	Less (10 × 15 × 15)	M-SDWD	0.992±0.001	0.000±0.000	0.890±0.021	0.461±0.060
		M-SDWD ($\lambda_1 = 0$)	0.995±0.000	0.000±0.000	1.000±0.000	0.000±0.000
		M-DWD	**0.996±0.000**	0.000±0.000	1.000±0.000	0.000±0.000
		Full SDWD	0.816±0.009	0.000±0.001	0.199±0.014	0.954±0.008
	No (30 × 15 × 15)	M-SDWD	0.995±0.001	0.000±0.000	0.934±0.016	-
		M-SDWD ($\lambda_1 = 0$)	0.997±0.001	0.000±0.000	1.000±0.000	-

N	Sparsity	Methods	Cor	Mis	TP	TN
		M-DWD	**0.999±0.000**	0.000±0.000	1.000±0.000	-
		Full SDWD	0.848±0.005	0.000±0.000	0.342±0.027	-

Table 3

Simulation results among simulations with correlation greater than 0.5 under the high dimensional scenario (30 × 15 × 15). In the Sparsity column, the numbers in parentheses indicate the number of non-zero variables in each dimension. "Cor" is the correlation between the estimated linear hyperplane and the true hyperplane. "Mis" is the average misclassification rate. "TP" is the true positive rate, i.e., the proportion of non-zero coefficients that are correctly estimated to be non-zero."TN" is the true negative rate, i.e., the proportion of zero variables that are correctly estimated to be zero. The margins of error (2* standard errors across 200 replicates) for each statistic are also listed following the ± symbol.

N	Sparsity	Methods	Cor	Mis	TP	TN	Prop. cor >0.5
40	Even more (3 × 3 × 3)	M-SDWD	**0.905±0.031**	**0.045±0.018**	0.145±0.020	0.853±0.109	0.16
		M-SDWD ($\lambda_1 = 0$)	0.783±0.039	0.074±0.029	0.216±0.000	0.000±0.000	0.175
		M-DWD	0.816±0.035	0.068±0.023	0.216±0.000	0.000±0.000	0.175
		Full SDWD	0.739±0.031	0.112±0.031	0.033±0.006	1.014±0.001	0.22
	More (5 × 5 × 5)	M-SDWD	**0.888±0.017**	**0.034±0.011**	0.638±0.052	0.802±0.055	0.61
		M-SDWD ($\lambda_1 = 0$)	0.840±0.022	0.039±0.012	1.000±0.000	0.000±0.000	0.585
		M-DWD	0.863±0.021	0.039±0.013	1.000±0.000	0.000±0.000	0.56
		Full SDWD	0.682±0.021	0.044±0.014	0.091±0.015	0.997±0.001	0.41
	Less (10 × 15 × 15)	M-SDWD	0.983±0.002	0.000±0.000	0.872±0.026	0.394±0.060	1
		M-SDWD ($\lambda_1 = 0$)	0.988±0.001	0.000±0.000	1.000±0.000	0.000±0.000	1
		M-DWD	**0.990±0.001**	0.000±0.000	1.000±0.000	0.000±0.000	1
		Full SDWD	0.682±0.013	0.000±0.000	0.135±0.015	0.956±0.011	0.885
	No (30 × 15 × 15)	M-SDWD	0.987±0.003	0.000±0.000	0.869±0.028	-	1
		M-SDWD ($\lambda_1 = 0$)	0.994±0.001	0.000±0.000	1.000±0.000	-	1
		M-DWD	**0.997±0.000**	0.000±0.000	1.000±0.000	-	1
		Full SDWD	0.727±0.009	0.000±0.000	0.262±0.032	-	0.99
100	Even more (3 × 3 × 3)	M-SDWD	**0.922±0.051**	**0.136±0.056**	0.045±0.009	1.000±0.018	0.09
		M-SDWD ($\lambda_1 = 0$)	0.794±0.064	0.146±0.056	0.064±0.000	0.000±0.000	0.085
		M-DWD	0.784±0.060	0.153±0.042	0.064±0.000	0.000±0.000	0.115
		Full SDWD	0.840±0.037	0.194±0.033	0.022±0.004	1.015±0.001	0.215
	More (5 × 5 × 5)	M-SDWD	**0.935±0.010**	0.048±0.010	0.697±0.038	0.831±0.042	0.905
		M-SDWD ($\lambda_1 = 0$)	0.899±0.012	0.051±0.010	1.000±0.000	0.000±0.000	0.88
		M-DWD	0.926±0.010	**0.042±0.009**	1.000±0.000	0.000±0.000	0.825
		Full SDWD	0.760±0.017	0.060±0.011	0.156±0.016	0.994±0.001	0.765
	Less (10 × 15 × 15)	M-SDWD	0.992±0.001	0.000±0.000	0.890±0.021	0.461±0.060	1
		M-SDWD ($\lambda_1 = 0$)	0.995±0.000	0.000±0.000	1.000±0.000	0.000±0.000	1
		M-DWD	0.996±0.000	0.000±0.000	1.000±0.000	0.000±0.000	1
		Full SDWD	0.818±0.008	0.000±0.000	0.198±0.013	0.955±0.007	0.995
	No (30 × 15 × 15)	M-SDWD	0.995±0.001	0.000±0.000	0.934±0.016	-	1
		M-SDWD ($\lambda_1 = 0$)	0.997±0.001	0.000±0.000	1.000±0.000	-	1

N	Sparsity	Methods	Cor	Mis	TP	TN	Prop. cor >0.5
		M-DWD	0.999±0.000	0.000±0.000	1.000±0.000	-	1
		Full SDWD	0.848±0.005	0.000±0.000	0.342±0.027	-	1

Table 4

Simulation results among simulations with correlation greater than 0.5 under the low dimensional scenario (15 × 4 × 5). In the Sparsity column, the numbers in parentheses indicate the number of non-zero variables in each dimension. "Cor" is the correlation between the estimated linear hyperplane and the true hyperplane. "Mis" is the average misclassification rate. "TP" is the true positive rate, i.e., the proportion of non-zero coefficients that are correctly estimated to be non-zero."TN" is the true negative rate, i.e., the proportion of zero coefficients that are correctly estimated to be zero. The margins of error (2* standard errors across 200 replicates) for each statistic are also listed following the ± symbol.

N	Sparsity	Methods	Cor	Mis	TP	TN	Prop. of Cor >0.5
40	More (5 × 2 × 2)	M-SDWD	**0.829±0.028**	0.159±0.030	0.616±0.077	0.768±0.079	0.31
		M-SDWD ($\lambda_1 = 0$)	0.778±0.030	0.179±0.030	1.000±0.000	0.000±0.000	0.34
		M-DWD	0.815±0.029	**0.149±0.029**	1.000±0.000	0.000±0.000	0.3
		Full SDWD	0.721±0.038	0.159±0.033	0.277±0.052	0.955±0.019	0.22
	Less (5 × 4 × 5)	M-SDWD	0.851±0.019	0.101±0.017	0.684±0.044	0.518±0.057	0.845
		M-SDWD ($\lambda_1 = 0$)	**0.853±0.018**	0.102±0.017	1.000±0.000	0.000±0.000	0.865
		M-DWD	0.877±0.017	0.096±0.016	1.000±0.000	0.000±0.000	0.84
		Full SDWD	0.696±0.021	**0.089±0.018**	0.280±0.033	0.869±0.027	0.54
	No (15 × 4 × 5)	M-SDWD	0.930±0.010	0.030±0.008	0.816±0.036	-	0.95
		M-SDWD ($\lambda_1 = 0$)	0.935±0.011	0.031±0.009	1.000±0.000	-	0.96
		M-DWD	**0.947±0.010**	**0.027±0.009**	1.000±0.000	-	0.94
		Full SDWD	0.724±0.016	0.029±0.008	0.434±0.049	-	0.765
100	More (5 × 2 × 2)	M-SDWD	**0.872±0.024**	0.191±0.025	0.673±0.058	0.772±0.058	0.535
		M-SDWD ($\lambda_1 = 0$)	0.850±0.023	0.186±0.024	1.000±0.000	0.000±0.000	0.515
		M-DWD	0.853±0.026	0.190±0.025	1.000±0.000	0.000±0.000	0.525
		Full SDWD	0.800±0.027	**0.177±0.025**	0.340±0.042	0.953±0.016	0.45
	Less (5 × 4 × 5)	M-SDWD	0.929±0.011	**0.090±0.014**	0.749±0.033	0.536±0.051	0.945
		M-SDWD ($\lambda_1 = 0$)	0.927±0.010	0.094±0.014	1.000±0.000	0.000±0.000	0.95
		M-DWD	**0.939±0.010**	0.093±0.014	1.000±0.000	0.000±0.000	0.95
		Full SDWD	0.768±0.018	0.106±0.016	0.336±0.032	0.838±0.028	0.83
	No (15 × 4 × 5)	M-SDWD	0.962±0.009	0.031±0.010	0.879±0.030	-	0.99
		M-SDWD ($\lambda_1 = 0$)	0.968±0.008	0.033±0.010	1.000±0.000	-	0.995
		M-DWD	**0.975±0.006**	**0.030±0.010**	1.000±0.000	-	0.99
		Full SDWD	0.816±0.013	0.036±0.010	0.437±0.037	-	0.925

Table 5

Simulation results under the high dimensional scenario (30 × 15 × 15) when the true model is rank-2. In the Sparsity column, the numbers in parentheses indicate the number of non-zero variables in each dimension. “Cor” is the correlation between the estimated linear hyperplane and the true hyperplane. “Mis” is the average misclassification rate. “TP” is the true positive rate, i.e., the proportion of non-zero coefficients that are correctly estimated to be non-zero.“TN” is the true negative rate, i.e., the proportion of zero coefficients that are correctly estimated to be zero. The margins of error (2* standard errors across 200 replicates) for each statistic are also listed following the ± symbol.

N	**Sparsity**	**Methods**	**Cor**	**Mis**	**TP**	**TN**
40	More (5 × 5 × 5)	M-SDWD (R=2)	**0.833±0.020**	**0.027±0.011**	0.714±0.038	0.761±0.043
		M-SDWD (λ_1 = 0, R=2)	0.747±0.025	0.040±0.013	1.000±0.000	0.000±0.000
		M-SDWD (R=1)	0.779±0.025	0.035±0.014	0.678±0.039	0.716±0.050
		M-SDWD (λ_1 = 0, R=1)	0.745±0.026	0.043±0.015	1.000±0.000	0.000±0.000
		M-DWD	0.736±0.033	0.055±0.019	1.000±0.000	0.000±0.000
		Full SDWD	0.591±0.024	0.071±0.015	0.150±0.013	0.984±0.007
	Less (10 × 15 × 15)	M-SDWD (R=2)	0.984±0.003	0.000±0.000	0.942±0.016	0.402±0.060
		M-SDWD (λ_1 = 0, R=2)	**0.990±0.002**	0.000±0.000	1.000±0.000	0.000±0.000
		M-SDWD (R=1)	0.813±0.011	0.000±0.000	0.840±0.029	0.488±0.062
		M-SDWD (λ_1 = 0, R=1)	0.821±0.010	0.000±0.000	1.000±0.000	0.000±0.000
		M-DWD	0.786±0.016	0.000±0.000	1.000±0.000	0.000±0.000
		Full SDWD	0.750±0.009	0.000±0.000	0.239±0.018	0.937±0.012
	No (30 × 15 × 15)	M-SDWD (R=2)	**0.996±0.000**	0.000±0.000	1.000±0.000	-
		M-SDWD (λ_1 = 0, R=2)	**0.996±0.000**	0.000±0.000	1.000±0.000	-
		M-SDWD (R=1)	0.790±0.012	0.000±0.000	0.836±0.033	-
		M-SDWD (λ_1 = 0, R=1)	0.799±0.010	0.000±0.000	1.000±0.000	-
		M-DWD	0.763±0.016	0.000±0.000	1.000±0.000	-
		Full SDWD	0.784±0.006	0.000±0.000	0.399±0.036	-
100	More (5 × 5 × 5)	M-SDWD (R=2)	**0.925±0.010**	**0.012±0.004**	0.841±0.026	0.754±0.041
		M-SDWD (λ_1 = 0, R=2)	0.889±0.011	0.016±0.005	1.000±0.000	0.000±0.000
		M-SDWD (R=1)	0.853±0.012	0.015±0.004	0.738±0.031	0.818±0.039
		M-SDWD (λ_1 = 0, R=1)	0.829±0.014	0.019±0.005	1.000±0.000	0.000±0.000
		M-DWD	0.825±0.020	0.027±0.010	1.000±0.000	0.000±0.000
		Full SDWD	0.797±0.014	0.026±0.007	0.236±0.016	0.992±0.002
	Less (10 × 15 × 15)	M-SDWD (R=2)	0.992±0.001	0.000±0.000	0.958±0.012	0.456±0.061
		M-SDWD (λ_1 = 0, R=2)	**0.996±0.000**	0.000±0.000	1.000±0.000	0.000±0.000
		M-SDWD (R=1)	0.807±0.010	0.000±0.000	0.871±0.024	0.526±0.062
		M-SDWD (λ_1 = 0, R=1)	0.807±0.010	0.000±0.000	1.000±0.000	0.000±0.000
		M-DWD	0.797±0.012	0.000±0.000	1.000±0.000	0.000±0.000
		Full SDWD	0.856±0.005	0.000±0.000	0.346±0.019	0.935±0.012

N	**Sparsity**	**Methods**	**Cor**	**Mis**	**TP**	**TN**
	No (30 × 15 × 15)	M-SDWD (R=2)	**0.998±0.000**	0.000±0.000	1.000±0.000	-
		M-SDWD ($\lambda_1 = 0$, R=2)	**0.998±0.000**	0.000±0.000	1.000±0.000	-
		M-SDWD (R=1)	0.786±0.009	0.000±0.000	0.908±0.023	-
		M-SDWD ($\lambda_1 = 0$, R=1)	0.790±0.009	0.000±0.000	1.000±0.000	-
		M-DWD	0.758±0.014	0.000±0.000	1.000±0.000	-
		Full SDWD	0.871±0.004	0.000±0.000	0.466±0.029	-