Multilinear Regression for Embedded Feature Selection with Application to fMRI Analysis

Xiaonan Song
Department of Computer Science
Hong Kong Baptist University, China
sxnzxz@gmail.com

Haiping Lu
Department of Computer Science
University of Sheffield, UK
h.lu@sheffield.ac.uk

Abstract

Embedded feature selection is effective when both prediction and interpretation are needed. The Lasso and its extensions are standard methods for selecting a subset of features while optimizing a prediction function. In this paper, we are interested in embedded feature selection for multidimensional data, wherein (1) there is no need to reshape the multidimensional data into vectors and (2) structural information from multiple dimensions are taken into account. Our main contribution is a new method called Regularized multilinear regression and selection (Remurs) for automatically selecting a subset of features while optimizing prediction for multidimensional data. Both nuclear norm and the ℓ_1-norm are carefully incorporated to derive a multi-block optimization algorithm with proved convergence. In particular, Remurs is motivated by fMRI analysis where the data are multidimensional and it is important to find the connections of raw brain voxels with functional activities. Experiments on synthetic and real data show the advantages of Remurs compared to Lasso, Elastic Net, and their multilinear extensions.

Introduction

Linear regression is popular in modeling the relationship between a scalar *response* y and a vector of I predictors $\mathbf{x} \in \mathbb{R}^I$, with two objectives: accurate prediction on future data and interpretation of the model (Hastie, Tibshirani, and Friedman 2009). It can be fitted to M training samples $(\mathbf{x}_m, y_m)_{m=1}^M$ via a loss function $J(\cdot)$ plus a regularization function $\Omega(\cdot)$ formulated as follows:

$$\min_{\mathbf{w}} \sum_{m=1}^{M} J\left(\langle \mathbf{x}_m, \mathbf{w} \rangle, y_m\right) + \Omega\left(\mathbf{w}\right), \tag{1}$$

where $\mathbf{w} \in \mathbb{R}^I$ is the *coefficient vector*, and $\langle \cdot, \cdot \rangle$ is the standard Euclidean inner product. A classical linear regression method is the least square loss with ℓ_1-norm regularization, leading to Lasso (Tibshirani 1996). Lasso gives a sparse $\mathbf{w}$, thus having *feature selection* embedded.

However, many real-world data are *multidimensional*, i.e., *tensor* observations. They need to be *vectorized* first for linear regression. This vectorization completely ignores underlying multidimensional structure, e.g., the spatial or temporal coherence. It also tends to create very high-dimensional vectors, leading to severe small sample size (SSS) problem (Kolda and Bader 2009). This motivates *multilinear regression* models which represent an observation as a tensor $\mathcal{X}$ and learn a *coefficient tensor* $\mathcal{W}$ via model fitting (Suzuki 2015). They extend Problem (1) to tensor data as

$$\min_{\mathcal{W}} \sum_{m=1}^{M} J\left(\langle \mathcal{X}_m, \mathcal{W} \rangle, y_m\right) + \Omega\left(\mathcal{W}\right). \tag{2}$$

Existing multilinear regression models for Problem (2) impose a low-rank constraint on $\mathcal{W}$ to leverage the structure information within $\mathcal{X}$ by fixing the CANDECOMP/PARAFAC (CP) rank of $\mathcal{W}$ as a priori. E.g., Su *et al.* (2012) assume $\mathcal{W}$ to be rank-one, which is too restrictive to properly fit the model. Guo *et al.* (2012) and Zhou *et al.* (2013) impose a rank-R constraint via a tensor factorization model, which have many local minima. Moreover, none of them has feature selection embedded as Lasso. Tan *et al.* (2012) also impose a rank-R constraint but they apply ℓ_1-norm regularization to factor matrices (which are multiplied to produce $\mathcal{W}$) to promote sparsity in $\mathcal{W}$ *indirectly*, which hurts the interpretability.

For matrix (second-order tensor) data, the *nuclear norm* (a.k.a., the trace norm) was used as a low-rank constraint on coefficient matrix to solve the second-order version of Problem (2) with various regression models, such as logistic regression (Tomioka and Aihara 2007) and generalized linear models (Zhou and Li 2014). It was also combined with ℓ_2-norm in the hinge-loss regression (Luo et al. 2015). Nonetheless, these regression methods are formulated only for matrix data, and they do not have feature selection embedded either. In addition, the combination of nuclear norm and ℓ_1-norm appears in other problems such as subspace clustering (Wang, Xu, and Leng 2013).

This paper aims to solve the multilinear regression Problem (2) with feature selection embedded. Our work is motivated by the use of the *tensor nuclear norm* in other models, such as *multilinear multitask learning* (Romera-Paredes et al. 2013) and *tensor completion* (Signoretto, De Lathauwer, and Suykens 2012; Tomioka, Hayashi, and Kashima 2010; Gandy, Recht, and Yamada 2011; Richard, Savalle, and Vayatis 2012; Liu et al. 2013; Signoretto et al. 2014). Note that both models solve problems different from the multilinear regression problem (2). Furthermore, multilinear multitask

learning represents each observation as a *vector* and forms a tensor by observation $\times$ modality $\times$ task. In contrast, multilinear regression represents each observation as a *tensor* to preserve the underlying spatial/temporal coherence.

Built on the above, this paper proposes a new method of **re**gularized **mu**ltilinear **r**egression and **s**election (**Remurs**) for tensor data. It is an extension of Lasso to tensor data using both tensor nuclear norm and ℓ_1-norm regularization. Gaiffas and Lecué (2011) pointed out that a mixture of nuclear norm and ℓ_1-norm can make the prediction accuracy less sensitive to the feature size, in contexts of matrix completion and multitask learning. However, this has not been studied for the multilinear regression problem (2). Remurs embodies a tensor version of this mixture. Therefore, we carry out extensive synthetic experiments to study its prediction accuracy when the gap between the feature size and the sample number increases.

The optimization problem for Remurs is convex but non-smooth. Thus, we derive an alternating direction method of multipliers (ADMM) (Boyd et al. 2011) algorithm and provide the convergence guarantee. Integrating ℓ_1-norm and tensor nuclear norm, we need to cope with more auxiliary variables and consider the feature selection capability at the same time. Finally, we apply Remurs to real-world fMRI data, where Remurs can provide stable and accurate classification results with good interpretability, outperforming competing methods on the whole.

Regularized multilinear regression with feature selection embedded

Notations and definitions. We follow the notations in (Kolda and Bader 2009) to denote vectors by lowercase boldface letters, e.g., $\mathbf{a}$; matrices by uppercase boldface letters, e.g., $\mathbf{A}$; and tensors by calligraphic letters, e.g., $\mathcal{A}$. We denote their elements with indices in parentheses, and indices by lowercase letters spanning the range from 1 to the uppercase letter of the index, e.g., $n = 1, \ldots, N$. An Nth-order tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ is addressed by N indices $\{i_n\}$. Each i_n addresses the n-mode of $\mathcal{A}$. The scalar product of two tensors $\langle \mathcal{A}, \mathcal{B} \rangle \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is defined as: $\langle \mathcal{A}, \mathcal{B} \rangle := \sum_{i_1} \cdots \sum_{i_N} \mathcal{A}(i_1, \ldots, i_N) \cdot \mathcal{B}(i_1, \ldots, i_N)$. Unfolding $\mathcal{A}$ along the n-mode is denoted as $\mathbf{A}_{(n)} \in \mathbb{R}^{I_n \times (I_1 \times \cdots \times I_{n-1} \times I_{n+1} \times \cdots \times I_N)}$, where the column vectors of $\mathbf{A}_{(n)}$ are the n-mode vectors of $\mathcal{A}$. Its opposite operation is defined as $\text{fold}_n(\mathbf{A}_{(n)}) := \mathcal{A}$. The operation $\text{tensor}_N(\cdot)$ is the opposite of the vectorization operation $\text{vec}(\cdot)$. The Frobenius norm of a tensor $\mathcal{A}$ is defined as $\|\mathcal{A}\|_{\text{F}} := \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle}$. $\sigma_j(\mathbf{A})$ denotes the jth largest singular value of $\mathbf{A}$. The nuclear norm of matrix $\mathbf{A}$ is denoted as $\|\mathbf{A}\|_* := \sum_j \sigma_j(\mathbf{A})$.

The Remurs model. Given an Nth-order tensor dataset with M observations $\{\mathcal{X}_m \in \mathbb{R}^{I_1 \times \cdots \times I_N}, m = 1, \ldots, M\}$, let $\mathbf{y} = (y_1, \ldots, y_M)^\top$ be the response. After a location and scale transformation, the response is centered and predictors are standardized: $\sum_{m=1}^M y_m = 0$; $\sum_{m=1}^M \mathcal{X}_m(i_1, \ldots, i_N) = 0$ and $\sum_{m=1}^M \mathcal{X}_m^2(i_1, \ldots, i_N)/M = 1$ for $i_n = 1, \ldots, I_n$. The Remurs model assumes the coefficient tensor $\mathcal{W} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ to be both low rank and sparse. For any fixed non-negative parameter γ and rank r, we define the Remurs problem as

$$\min_{\mathcal{W}} \quad \frac{1}{2} \sum_{m=1}^M \left(y_m - \langle \mathcal{X}_m, \mathcal{W} \rangle\right)^2 + \gamma \|\mathcal{W}\|_1 \quad \text{s.t. } \text{rank}(\mathcal{W}) \leq r, \tag{3}$$

where $\|\mathcal{W}\|_1$ is the entrywise ℓ_1-norm defined as

$$\|\mathcal{W}\|_1 = \sum_{i_1} \sum_{i_2} \cdots \sum_{i_N} |\mathcal{W}(i_1, i_2, \ldots, i_N)|. \tag{4}$$

However, the computation of tensor rank is NP-hard (Hillar and Lim 2013). Therefore, we cast the rank constraint into a convex *tensor nuclear norm* penalty (Liu et al. 2013), with a Lagrange multiplier τ. The Remurs problem of (3) can then be written as

$$\min_{\mathcal{W}} \quad \frac{1}{2} \sum_{m=1}^M \left(y_m - \langle \mathcal{X}_m, \mathcal{W} \rangle\right)^2 + \tau \|\mathcal{W}\|_* + \gamma \|\mathcal{W}\|_1, \tag{5}$$

where

$$\|\mathcal{W}\|_* = \frac{1}{N} \sum_{n=1}^N \|\mathbf{W}_{(n)}\|_*. \tag{6}$$

Remark 1. *This definition of tensor nuclear norm is based on the unfolded matrices. Unfolding a tensor into matrices loses some multidimensional structural information but still perserves* mode-wise structure*, which is* completely lost in vectorization*. Note that there is a different definition of tensor nuclear norm in (Yuan and Zhang 2014), as an* atomic norm*, i.e., a convex hull of rank-one tensors, which does not involve unfolding into matrices and could be explored in future work.*

Remark 2. *In Remurs, we apply ℓ_1-norm regularization* directly *on $\mathcal{W}$, rather than* indirectly *on the factor matrices of $\mathcal{W}$ in (Tan et al. 2012). Thus, our model has a direct control of the sparsity of $\mathcal{W}$ and a better interpretability in turn.*

Trade-off in Remurs. The penalty $\tau\|\mathcal{W}\|_* + \gamma\|\mathcal{W}\|_1$ in (5) allows Remurs to benefit from the virtues of both low rank and sparsity, just as the Elastic Net (Zou and Hastie 2005) combines the sparsity-inducing property of the ℓ_1-norm with the smoothness of the ℓ_2-norm. Consequently, there is a trade-off between the tensor nuclear norm and the ℓ_1-norm. We analyze three cases of (5) below:

- When $\tau = 0$, the Remurs degenerates to a linear model, i.e., the Lasso, which enforces sparsity only and enables automatic feature selection. To see this more clearly, we vectorize $\mathcal{X}_m$ and $\mathcal{W}$, and represent them as $\mathbf{x}_m$ and $\mathbf{w}$ respectively. Then (5) with $\tau = 0$ can be rewritten as the Lasso problem: $\min_{\mathcal{W}} \ \frac{1}{2}\sum_{m=1}^M \left(y_m - \langle \mathcal{X}_m, \mathcal{W} \rangle\right)^2 + \gamma\|\mathcal{W}\|_1 = \min_{\mathbf{w}} \ \frac{1}{2}\sum_{m=1}^M \left(y_m - \langle \mathbf{x}_m, \mathbf{w} \rangle\right)^2 + \gamma\|\mathbf{w}\|_1$.
- When $\gamma = 0$, Remurs degenerates to a model enforcing only low rank, denoted as $\text{Remurs}_{\gamma=0}$. The second-order version of $\text{Remurs}_{\gamma=0}$ embodies the same penalty as in (Tomioka and Aihara 2007; Zhou and Li 2014;

Luo et al. 2015). The tensor nuclear norm constraint captures the spatial/temporal coherence in multidimensional structure which helps alleviate the SSS problem.

- In the intermediate case, the ratio γ/τ balances the effects of sparsity and low rank, and also controls the percentage of features selected in turn.

ADMM-based algorithm for Remurs

This section derives an algorithm based on ADMM (Boyd et al. 2011) to solve the Remurs problem (5). We provide the proximity operators of the nuclear norm and the ℓ_1-norm first.

Definition 1 (Singular value thresholding)**.** *Consider the SVD of a matrix* $\mathbf{A} \in \mathbb{R}^{I_1 \times I_2}$, $\mathbf{A} = \mathbf{U}\ diag[\sigma_j(\mathbf{A})]\ \mathbf{V}^\top$, *where* $1 \leq j \leq \min(I_1, I_2)$. *For* $\mu > 0$, *the proximity operator of the nuclear norm is the singular value shrinkage operator (Cai, Candès, and Shen 2010):* $prox_{\mu\|\cdot\|_*}(\mathbf{A}) = \mathbf{U}\ diag\left[\max(\sigma_j(\mathbf{A}) - \mu, 0)\right]\ \mathbf{V}^\top$.

Definition 2 (Soft thresholding)**.** *Consider the* ℓ_1*-norm of a tensor* $\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ *defined in Eq. (4). For* $\nu > 0$, *its proximity operator is*

$$prox_{\nu\|\cdot\|_1}(\mathcal{A}) = \begin{cases} \mathcal{A}(i_1, \ldots, i_N) - \nu & \text{if } \mathcal{A}(i_1, \ldots, i_N) > \nu, \\ 0 & \text{if } |\mathcal{A}(i_1, \ldots, i_N)| \leq \nu, \\ \mathcal{A}(i_1, \ldots, i_N) + \nu & \text{if } \mathcal{A}(i_1, \ldots, i_N) < -\nu. \end{cases} \tag{7}$$

$(2+N)$-block separable convex problem

In (5), $\frac{1}{2}\sum_{m=1}^{M}(y_m - \langle \mathcal{X}_m, \mathcal{W}\rangle)^2$ is convex, differentiable with a Lipschitz gradient, and $\tau\|\mathcal{W}\|_* + \gamma\|\mathcal{W}\|_1$ is convex but not differentiable. We solve this problem via ADMM by first splitting $\mathcal{W}$ into two variables, $\mathcal{U}$ and $\mathcal{W}$:

$$\begin{aligned} \min_{\mathcal{U},\mathcal{W}} \quad & \frac{1}{2}\sum_{m=1}^{M}(y_m - \langle \mathcal{X}_m, \mathcal{U}\rangle)^2 + \tau\|\mathcal{W}\|_* + \gamma\|\mathcal{W}\|_1 \\ \text{s.t.} \quad & \mathcal{U} = \mathcal{W}. \end{aligned} \tag{8}$$

To solve (8), we need to solve the part containing $\mathcal{U}$ and the part containing $\mathcal{W}$ independently. The summation of the two regularizers ($\tau\|\mathcal{W}\|_* + \gamma\|\mathcal{W}\|_1$) on the same $\mathcal{W}$ makes the situation more complicated, since the proximity operator of this sum is non-explicit. An intuitive solution is to further split this sum into two parts, i.e, $\tau\|\mathcal{V}\|_*$ and $\gamma\|\mathcal{W}\|_1$. However, it is still difficult to solve the part of $\tau\|\mathcal{V}\|_*$, because $\tau\|\mathcal{V}\|_*$ is defined in Eq. (6) as a summation of nuclear norm of interdependent matrices $\{\mathbf{V}_{(n)}\}$, which share the same entries and hence cannot be optimized independently. Therefore, we introduce N auxiliary tensors $\{\mathcal{V}_n, n = 1, \ldots, N\}$ into (8) to obtain the following objective function where the proximity operator of each term is available:

$$\begin{aligned} \min_{\mathcal{U},\{\mathcal{V}_n\},\mathcal{W}} \quad & \frac{1}{2}\sum_{m=1}^{M}(y_m - \langle \mathcal{X}_m, \mathcal{U}\rangle)^2 \\ & + \frac{\tau}{N}\sum_{n=1}^{N}\|\mathbf{V}_{n(n)}\|_* + \gamma\|\mathcal{W}\|_1 \\ \text{s.t.} \quad & \mathcal{U} = \mathcal{W} \quad \text{and} \quad \mathcal{V}_n = \mathcal{W}, n = 1, \ldots, N. \end{aligned} \tag{9}$$

Augmented Lagrangian and further splitting

Integrating ℓ_1-norm and tensor nuclear norm, we need to cope with more auxiliary variables and consider feature selection capability. Existing related algorithms (Romera-Paredes et al. 2013) use $\mathcal{U}$ in (9) as the global variable, while we propose to use $\mathcal{W}$ instead for efficiency and sparsity, as $\mathcal{W}$ is sparse and much faster to compute. We provide the solution below.

The augmented Lagrangian associated with (9) is as follows:

$$\begin{aligned} & L_\rho(\mathcal{U}, \mathcal{V}_1, \ldots, \mathcal{V}_N, \mathcal{W}, \mathcal{P}, \mathcal{Q}_1, \ldots, \mathcal{Q}_N) \\ =& \frac{1}{2}\sum_{m=1}^{M}(y_m - \langle \mathcal{X}_m, \mathcal{U}\rangle)^2 + \frac{\tau}{N}\sum_{n=1}^{N}\|\mathbf{V}_{n(n)}\|_* + \gamma\|\mathcal{W}\|_1 \\ & + \langle \mathcal{P}, \mathcal{U} - \mathcal{W}\rangle + \frac{\rho}{2}\|\mathcal{U} - \mathcal{W}\|_{\mathrm{F}}^2 \\ & + \sum_{n=1}^{N}\left(\langle \mathcal{Q}_n, \mathcal{V}_n - \mathcal{W}\rangle + \frac{\rho}{2}\|\mathcal{V}_n - \mathcal{W}\|_{\mathrm{F}}^2\right), \end{aligned} \tag{10}$$

where $\rho > 0$ is the augmented Lagrangian parameter, and $\mathcal{P}$ and $\{\mathcal{Q}_n\}$ are dual variables. For convenience, we further introduce scaled dual variables $\mathcal{P}' = \frac{1}{\rho}\mathcal{P}$ and $\{\mathcal{Q}'_n = \frac{1}{\rho}\mathcal{Q}_n, n = 1, \ldots, N\}$ in the following computation. According to the ADMM framework, we can iteratively update $\mathcal{U}$, $\{\mathcal{V}_n\}$, $\mathcal{W}$, $\mathcal{P}'$, and $\{\mathcal{Q}'_n\}$ as follows:

Computing $\mathcal{U}^{k+1}$: Fixing all the other variables, we can calculate $\mathcal{U}^{k+1}$ below:

$$\begin{aligned} \mathcal{U}^{k+1} = \arg\min_{\mathcal{U}} \quad & \frac{1}{2}\sum_{m=1}^{M}(y_m - \langle \mathcal{X}_m, \mathcal{U}\rangle)^2 \\ & + \frac{\rho}{2}\left\|\mathcal{U} - \mathcal{W}^k + \mathcal{P}'^k\right\|_{\mathrm{F}}^2. \end{aligned} \tag{11}$$

This is equivalent to a linear-quadratic objective function, to which existing acceleration tricks can be well applied. We vectorize all the tensors in Eq. (11), e.g., $\mathbf{x}_m = \mathrm{vec}(\mathcal{X}_m)$, and denote $\mathbf{X} = (\mathbf{x}_1|\cdots|\mathbf{x}_M)^\top$ to get $\mathbf{u}^{k+1} = \left(\mathbf{X}^\top\mathbf{X} + \rho\mathbf{I}\right)^{-1}\left(\mathbf{X}^\top\mathbf{y} + \rho\left(\mathbf{w}^k - \mathbf{p}'^k\right)\right)$, where $\mathbf{I}$ is an identity matrix, $\mathbf{w}^k = \mathrm{vec}(\mathcal{W}^k)$, and $\mathbf{p}'^k = \mathrm{vec}(\mathcal{P}'^k)$. $\mathcal{U}^{k+1}$ can then be obtained by reshaping $\mathbf{u}^{k+1}$ into a tensor: $\mathcal{U}^{k+1} = \mathrm{tensor}_N\left(\mathbf{u}^{k+1}\right)$.

Computing $\mathcal{V}_n^{k+1}$: Fixing all the other variables, we calculate $\mathcal{V}_n^{k+1}$ based on Definition 1 as:

$$\begin{aligned} \mathcal{V}_n^{k+1} &= \arg\min_{\mathcal{V}_n} \frac{\tau}{N}\|\mathbf{V}_{n(n)}\|_* + \frac{\rho}{2}\left\|\mathcal{V}_n - \mathcal{W}^k + \mathcal{Q}_n'^k\right\|_{\mathrm{F}}^2 \\ &= \mathrm{fold}_n\left[\mathrm{prox}_{\frac{\tau}{N\rho}\|\cdot\|_*}\left(\mathbf{W}_{(n)}^k - \mathbf{Q}'^k_{n(n)}\right)\right]. \end{aligned} \tag{12}$$

Computing $\mathcal{W}^{k+1}$: Fixing all the other variables, we express $\mathcal{W}$-update as an averaging step using Definition 2: $\mathcal{W}^{k+1} = \arg\min_{\mathcal{W}}\ \gamma\|\mathcal{W}\|_1 + \frac{(N+1)\rho}{2}\left\|\mathcal{W} - \mathcal{Z}^{k+1}\right\|_{\mathrm{F}}^2 = \mathrm{prox}_{\frac{\gamma}{(N+1)\rho}\|\cdot\|_1}\left(\mathcal{Z}^{k+1}\right)$, where $\mathcal{Z}^{k+1} = \frac{\mathcal{U}^{k+1} + \sum_{n=1}^{N}\mathcal{V}_n^{k+1}}{N+1} + \frac{\mathcal{P}'^k + \sum_{n=1}^{N}\mathcal{Q}'^k_n}{N+1}$.

Computing $\mathcal{P}'^{k+1} = \mathcal{P}'^k + \mathcal{U}^{k+1} - \mathcal{W}^{k+1}$ and $\mathcal{Q}_n'^{k+1} = \mathcal{Q}_n'^k + \mathcal{V}_n^{k+1} - \mathcal{W}^{k+1}$.

Algorithm 1 The Remurs algorithm based on ADMM.

Input: A set of Nth-order tensor observations $\{\mathcal{X}_m \in \mathbb{R}^{I_1 \times \cdots \times I_N}\}$, the corresponding responses $\{y_m\}$ $(m = 1, \ldots, M)$, the maximum number of iterations K, and parameters ρ, τ, and γ.
Initialize: $\mathcal{U} = \mathcal{V} = \mathcal{W} = \mathcal{P}' = \mathcal{Q}' = 0$.
for $k = 1$ **to** K **do**
  $\mathbf{u}^{k+1} = \left(\mathbf{X}^\top \mathbf{X} + \rho \mathbf{I}\right)^{-1} \left(\mathbf{X}^\top \mathbf{y} + \rho \left(\mathbf{w}^k - \mathbf{p}'^k\right)\right)$.
  $\mathcal{U}^{k+1} = \text{tensor}_N\left(\mathbf{u}^{k+1}\right)$.
  for $n = 1$ **to** N **do**
    $\mathcal{V}_n^{k+1} = \text{fold}_n\left[\text{prox}_{\frac{\tau}{N\rho}\|\cdot\|_*}\left(\mathbf{W}_{(n)}^k - \mathbf{Q}'^k_{n(n)}\right)\right]$.
  end for
  $\mathcal{W}^{k+1} = \text{prox}_{\frac{\gamma\|\cdot\|_1}{(N+1)\rho}}\left(\frac{\mathcal{U}^{k+1} + \mathcal{P}'^k + \sum_{n=1}^N (\mathcal{V}_n^{k+1} + \mathcal{Q}'^k_n)}{N+1}\right)$.
  $\mathcal{P}'^{k+1} = \mathcal{P}'^k + \mathcal{U}^{k+1} - \mathcal{W}^{k+1}$.
  for $n = 1$ **to** N **do**
    $\mathcal{Q}'^{k+1}_n = \mathcal{Q}'^k_n + \mathcal{V}_n^{k+1} - \mathcal{W}^{k+1}$.
  end for
end for
Output: $\mathcal{W}$.

Algorithm 1 summarizes the algorithm for solving the Remurs problem (5). A large ρ tends to reduce primal residuals more while a small ρ tends to reduce the dual residuals more. Considering this trade-off, we fix ρ to 1 in implementation.

Convergence analysis

As both nuclear norm and ℓ_1-norm are nonsmooth and (9) is split into more than five parts (for $N \geq 3$), the convergence property of Algorithm 1 can not be directly obtained from existing results on the convergence of ADMM. Thus, we prove its convergence in terms of the objective function in the following theorem.

Theorem 1. *For any $\rho > 0$, the iterations in Algorithm 1 satisfy the residual convergence, objective convergence, and dual variable convergence of (9).*

Proof. Here, we provide a sketch only to save space. The key idea is to rewrite (9) into a two-block ADMM problem and verify that the unaugmented Lagrangian L_0 has a saddle point (Boyd et al. 2011). □

Experiments

We carry out experiments on *synthetic matrix data* to study the behaviors of Remurs against five factors and then on *real fMRI data* to study its classification performance and interpretability. We consider the following existing methods in the evaluation of *Remurs*:

- Linear regression: *Lasso* and Elastic Net (*ENet*).
- Multilinear regression with only tensor nuclear norm constraint: Remurs with only the nuclear norm (*Remurs*$_{\gamma=0}$), which can represent other multilinear regression methods (Tomioka and Aihara 2007; Zhou and Li 2014; Luo et al. 2015) with the same least-square loss function.
- Fixed-CP-rank multilinear regression: multivariate multilinear regression (*MMR*) (Su et al. 2012), and rank-R generalized linear tensor regression model (*GLTRM*) (Zhou, Li, and Zhu 2013).
- Adaptive-CP-rank multilinear regression: optimal-rank tensor ridge regression (*orTRR*) (Guo, Kotsia, and Patras 2012).

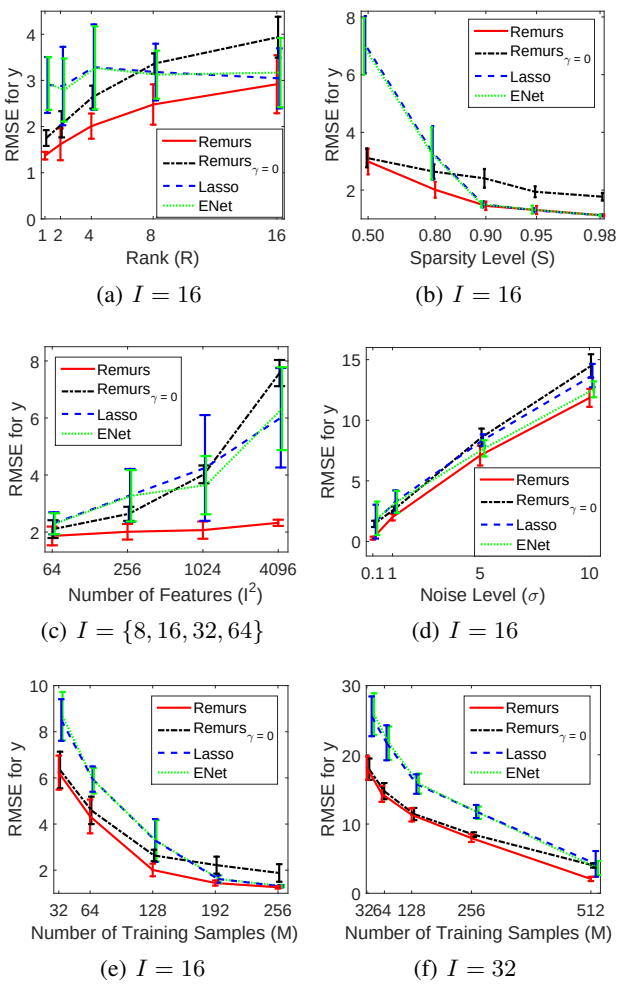
Prediction and sensitivity on synthetic data

Data generation. To verify the proof in (Gaiffas and Lecué 2011) that *a mixture of nuclear norm and ℓ_1-norm can make the prediction accuracy not sensitive to the feature size*, we study synthetic matrix data here. Each dataset (run) is generated from the following model guided by (Tibshirani 1996): $y = \langle \mathbf{X}, \mathbf{W} \rangle + \sigma\epsilon$, where ϵ is drawn from a standard normal distribution, σ controls the signal-to-noise ratio, and $\mathbf{X} \in \mathbb{R}^{I \times I}$. The predictors are drawn from a multivariate Gaussian with zero mean and a covariance matrix where the correlation between $\mathbf{X}(i,j)$ and $\mathbf{X}(p,q)$ is $0.5^{\sqrt{(i-p)^2+(j-q)^2}}$. We generate the true support as $\mathbf{W} = \mathbf{W}_1 \mathbf{W}_2^\top$, where $\mathbf{W}_d \in \mathbb{R}^{I \times R}, d = 1, 2$. R controls the rank of $\mathbf{W}$. The *sparsity level* S of $\mathbf{W}$ is the percentage of zero entries, controlled by each entry of $\mathbf{W}_d$ drawn from a Bernoulli distribution with probability of 1 equal to $\sqrt{\{1 - (1-S)^{(1/R)}\}}$.

Experiment settings. We compare *Remurs* against *Lasso*, *ENet*, and *Remurs*$_{\gamma=0}$. Results of other methods are not reported here, due to their poor results in SSS scenario even with rank known as a priori. We use two metrics to evaluate the performance: *prediction error* of y and *estimation error* of $\mathbf{W}$, both measured by the root-mean-squared error (RMSE). *Five* factors are varied to study the behaviors of each method: the feature size of $\mathbf{X}$ (I^2), the rank of $\mathbf{W}$ (R), the sparsity level of $\mathbf{W}$ (S), the number of training samples (M), and the noise level (σ). When studying one of the factors, other factors are fixed to $I = 16$, $R = \frac{1}{4}I$, $S = 0.8$, $M = \frac{1}{2}I^2$, and $\sigma = 1$, as a *standard case*. For each scenario, our simulated data consist of a training set of M samples and an independent test set of 1000 samples. Hyperparameters of all the methods are determined via *fourfold cross validation* on the training set, with range $\{10^{-3}, 5 \times 10^{-3}, 10^{-2}, 5 \times 10^{-2}, \ldots, 5 \times 10^2, 10^3\}$. We report results averaged over 10 runs with standard deviation.

Results and discussions. We show the prediction error varying with respect to the five factors in Fig. 1. Estimation errors of $\mathbf{W}$ show similar trends. We have the following observations.

- On the whole, Remurs outperforms Lasso and ENet in both true support estimation and prediction. This indicates that leveraging structure information in multidimensional data, Remurs is a better regression and embedded feature selection model.
- Remurs always outperforms its special case, Remurs$_{\gamma=0}$, demonstrating the robustness to noise (e.g., Fig. 1(d)) and better model fitting with a mixture of the ℓ_1-norm and the nuclear norm.
- As shown in Fig. 1(b), Lasso and ENet perform much better when the true $\mathbf{W}$ is sparser.
- Both Remurs and Remurs$_{\gamma=0}$ perform better when the true rank of support is smaller. Lasso and ENet are almost insensitive to rank changes, as shown in Fig. 1(a).



(a) $I = 16$ (b) $I = 16$

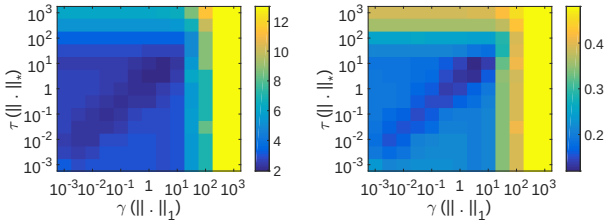
(c) $I = \{8, 16, 32, 64\}$ (d) $I = 16$

(e) $I = 16$ (f) $I = 32$

Figure 1: Results on synthetic data averaged over 10 runs with error bars indicating standard deviations. When one factor is studied, other factors are fixed as $I = 16$ ($I = 32$ in (f)), $R = \frac{1}{4}I$, $S = 0.8$, $M = \frac{1}{2}I^2$, and $\sigma = 1$.

- For severe SSS cases in Figs. 1(e) and 1(f), Remurs and Remurs$_{\gamma=0}$ have similar superior performance over Lasso and ENet. This indicates that the nuclear norm can better alleviate the overfitting problem.
- The prediction error of Remurs is stable and the standard deviation decreases with respect to the increased feature size I^2 (and $M = \frac{1}{2}I^2$), as in Fig. 1(c). In contrast, RMSE and standard deviation of other methods increase more with the growth of I^2. This case is largely consistent with the proof in (Gaiffas and Lecué 2011) that *a mixture of nuclear norm and ℓ_1-norm can make the prediction accuracy less sensitive to the feature size*.

Parameter sensitivity. Figure 2 shows the hyperparameter sensitivity in the *standard case* reported above. In terms of the prediction error (Fig. 2(a)), Remurs does not change too much as long as both τ and γ are less than 10. Much lower prediction RMSEs can be achieved when γ for ℓ_1-norm is slightly smaller than τ for nuclear norm, i.e., $1 \leq \gamma/\tau \leq 0.1$. While for estimation of the true support $\mathbf{W}$ (Fig. 2(b)), it is also better to keep $1 \leq \gamma/\tau \leq 0.1$. These observations provide useful guidelines for hyperparameter determination, and are consistent with those of real data experiments below.



(a) Prediction RMSE of y (b) Estimation RMSE of $\mathbf{W}$

Figure 2: Sensitivity of hyperparameters, τ and γ, on synthetic data: (a) prediction error on 1000 test samples and (b) estimation error of true $\mathbf{W}$, with fixed $I = 16$, $R = \frac{1}{4}I$, $S = 0.8$, $M = \frac{1}{2}I^2$, and $\sigma = 1$, all averaged over 10 runs.

Classification and interpretability for fMRI data

Dataset. We perform real-world fMRI classification on the CMU2008 dataset (Mitchell et al. 2008), with 3D fMRI data of size $51 \times 61 \times 23$ (71,553 voxels). It aims to predict human brain activity associated with the meanings of nouns. The data acquisition experiments had nine right-handed subjects who viewed 60 different word-picture stimuli from 12 semantic categories, with 5 exemplars per category and 6 runs per stimulus. The numbers of valid brain voxels range from 19,750 to 21,764. Data were preprocessed with the SPM software and we use the preprocessed 3D data available online,[1] where each voxel feature is the respective mean percent signal change (PSC) value over time. We focus on binary classification of "animals" vs. "tool". Following (Kampa et al. 2014), the class "animals" combines observations from "animal" and "insect", and the class "tools" combines "tool" and "furniture" in the CMU dataset. Thus, there are 120 observations for each class.

Experiment settings. We study *Lasso*, *ENet*, *orTRR*, *MMR*, *GLTRM*, and *Remurs*$_{\gamma=0}$, where Lasso/ENet takes only valid (19,750 to 21,764) brain voxels as input. We follow (Kampa et al. 2014) to arrange the test, validation and training sets in the format of (1:1:4) for the six runs in all the experiments of Table 1, and report the average results. Hyperparameters are selected by *fivefold cross validation* with the same range in synthetic experiments. In addition, γ and τ of Remurs are constrained by reasonable feature numbers (5% to 50% of brain voxels) in cross validation. Note that for fMRI, other regularizations based on total variation (Michel et al. 2011) or graph (Grosenick et al. 2013) can also be incorporated for further improvement in future work.

Classification accuracy. Table 1 shows the classification

[1] http://www.cs.cmu.edu/afs/cs/project/theo-73/www/science2008/data.html

Table 1: The classification accuracy for nine subjects (ID 1-9) and their average (Acc), and the average sparsity (S in the last row) for fMRI data. In each row, the best results are highlighted with bold font and the second best with underline.

ID	Remurs	Remurs$_{\gamma=0}$	ENet	Lasso	orTRR	MMR
1	**95.00±0.06**	92.50±0.07	93.33±0.06	92.50±0.07	80.00±0.16	61.67±0.09
2	83.33±0.05	**84.17±0.07**	78.33±0.05	77.50±0.07	69.17±0.13	60.83±0.09
3	**82.50±0.08**	**82.50±0.08**	77.50±0.05	79.17±0.07	74.17±0.13	62.50±0.03
4	**91.67±0.05**	90.83±0.09	**91.67±0.05**	90.00±0.05	70.00±0.12	55.00±0.14
5	62.50±0.09	**65.00±0.08**	63.33±0.09	55.83±0.05	52.50±0.12	48.33±0.10
6	79.17±0.09	72.50±0.07	**80.00±0.07**	79.17±0.05	68.33±0.13	55.83±0.09
7	74.17±0.11	**76.67±0.09**	70.83±0.13	70.00±0.05	67.50±0.07	63.33±0.09
8	61.67±0.15	**62.50±0.17**	53.33±0.13	55.83±0.05	50.83±0.06	45.83±0.09
9	73.33±0.06	**76.67±0.05**	70.83±0.06	74.17±0.05	72.50±0.17	51.67±0.14
Acc	**78.15±0.08**	**78.15±0.09**	75.46±0.08	74.91±0.06	68.33±0.12	56.11±0.10
S	0.78	0.00	**0.85**	0.61	0.00	0.00

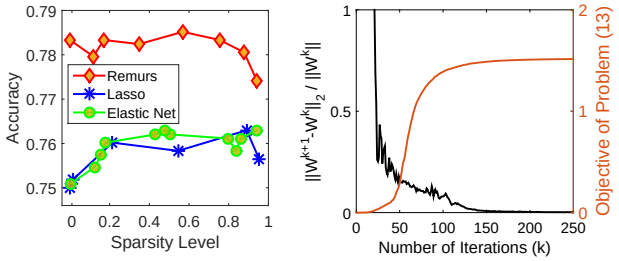
accuracy of all methods except *GLTRM*,[2] with the best results highlighted with bold font and the second best with underline. The two linear methods, ENet and Lasso, have similar accuracy. ENet is slightly better. Remurs and Remurs$_{\gamma=0}$ achieve the best overall performance, 2.69% higher than ENet. This indicates that the tensor nuclear norm penalty has good capability of modeling low-rank structure in 3D real data. OrTRR and MMR give the worst accuracy due to their local minima problem and fixed-rank assumption.

Accuracy versus sparsity. Figure 3(a) illustrates the averaged classification accuracy versus sparsity on fMRI data. Every point is obtained by fixed hyperparameters (without cross validation). For each method, grid search on hyperparameters determines the sparsity with close points removed. Remurs achieves stable and superior accuracy on different levels of sparsity. Note that the accuracy of Remurs does not decrease as much as Lasso and ENet with low sparsity level. This implies the benefit of proper modeling of spatial coherence via the tensor nuclear norm.

Convergence analysis. Figure 3(b) shows the convergence of $\mathcal{W}$ in (9), which is important for feature selection. Convergence of the objective function value of (9) is also shown in the figure, which is consistent with Theorem 1. Both have a fast convergence on big fMRI data with only about 200 iterations needed.

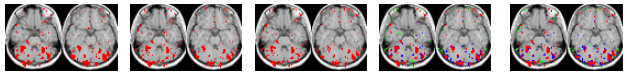
Feature selection. The last row in Table 1 shows the average sparsity, indicating the feature selection capability. The preprocessed 3D fMRI data have only 28.85% of meaningful voxels, with other voxels filled with zeros. Note that for a fair comparison, in all experiments above, we feed only the valid/meaningful voxels (about 20,640 on average) to linear models, while the whole 3D volume (71,553 voxels) is fed to multilinear models. Note that each voxel is considered as a feature here. When calculating sparsity, we use the

[2]The best result of *GLTRM* (for various ranks) is an average accuracy of 54.44%, which is poorer than all methods included in the table. Results in (Kampa et al. 2014) are not included in the table, because its best result of 72.13% (obtained with logistic regression plus ENet) is poorer than the ENet result of 75.46% reported above with a more recent implementation (Liu, Ji, and Ye 2009). We also studied PCA plus SVM, and its average accuracy is 71.39%.



(a) Accuracy vs. sparsity (b) Convergence analysis

Figure 3: Studies on fMRI data: (a) classification accuracy versus sparsity, and (b) the convergence of $\mathcal{W}$ (in black) and that of the objective function value for (9) (in red).



(a) Remurs (b) ENet (c) Lasso (d) (a) & (b) (e) (a) & (c)

Figure 4: Visualization of selected voxels for the 7-8th slices of Subject 1 by Remurs, ENet, and Lasso in red in (a), (b), and (c). In (d)/(e), voxels selected by both Remurs and ENet/Lasso are in red, those selected by Remurs only are in blue, and those selected by ENet/Lasso only are in green. This figure is best viewed in color.

mean number of meaningful/valid voxels (20,640) as the denominator. Although ENet achieves the best sparsity in Table 1 with its best cross validation accuracy, Fig. 3(a) shows Remurs can achieve better accuracy with the same sparsity level of 0.85. In Table 1, Remurs gives the second best sparsity, with only 4,549 features but has the same average accuracy as Remurs$_{\gamma=0}$ with 20,640 features. The methods with low rank constraint only (Remurs$_{\gamma=0}$, orTRR, and MMR) have no feature selection capability (zero sparsity).

Visualization of selected voxels. Finally, we study the voxels selected by Remurs, ENet, and Lasso with parameters set as in Table 1 on Subject 1 (the best performing subject). Because the number of selected voxels varies for different methods, we choose the fourth run where at least 1,700 voxels are selected by each method. We rank the importance of voxels by their weights (absolute values) and study the top 1,700 voxels in the comparison below.

We first perform a *quantitative study* by computing the number of 26-connected components (3D connectivity with 26-connected neighborhood) formed by these 1,700 voxels for each method. There are *208*, *276*, and *291* components for Remurs, ENet, and Lasso, respectively. Thus, the same number of voxels selected by Remurs formed *68* (*83*) fewer components than those by ENet (Lasso), indicating that the regions determined by Remurs have better spatial coherence.

Next, we do a *qualitative study* by visualizing the 1,700 voxels of each method in the 7-8th slices highlighted in red in Figs. 4(a), 4(b), and 4(c). We further overlay the selected voxels of Remurs and ENet (Lasso) in Fig. 4(d) (4(e)), where the common voxels are in red, Remurs-only voxels in blue, and ENet-only (Lasso-only) voxels in green. We can see that

the regions selected by the three methods enjoy large consistency, with most selected regions overlapped. However, the voxels selected by ENet and Lasso are more dispersed than those by Remurs, demonstrating the better interpretability of Remurs.

Conclusions

We proposed Remurs, a regularized multilinear regression model with feature selection embedded for tensor data. It incorporates the tensor nuclear norm and the ℓ_1-norm to preserve the spatial/temporal coherence. We developed an optimization algorithm with feature selection capability based on ADMM to solve a multiple-block separable and nonsmooth problem, with proved convergence. We evaluated Remurs on synthetic data and real-world fMRI data. The results show its robust prediction accuracy against feature sizes, and good classification accuracy and interpretability.

Acknowledgments

This research was supported by Research Grants Council of the Hong Kong SAR (Grant 22200014 and the Hong Kong PhD Fellowship Scheme). We thank Jian Lou for helpful discussions on ADMM.

References

Boyd, S.; Parikh, N.; Chu, E.; Peleato, B.; and Eckstein, J. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning* 3(1):1–122.

Cai, J.-F.; Candès, E. J.; and Shen, Z. 2010. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization* 20(4):1956–1982.

Gaiffas, S., and Lecué, G. 2011. Sharp oracle inequalities for high-dimensional matrix prediction. *IEEE Trans. on Information Theory* 57(10):6942–6957.

Gandy, S.; Recht, B.; and Yamada, I. 2011. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems* 27(2):025010 (19pp).

Grosenick, L.; Klingenberg, B.; Katovich, K.; Knutson, B.; and Taylor, J. E. 2013. Interpretable whole-brain prediction analysis with graphnet. *NeuroImage* 72:304–321.

Guo, W.; Kotsia, I.; and Patras, I. 2012. Tensor learning for regression. *IEEE Trans. on Image Processing* 21(2):816–827.

Hastie, T.; Tibshirani, R.; and Friedman, J. 2009. *The Elements of Statistical Learning*. Springer, 2nd edition.

Hillar, C. J., and Lim, L.-H. 2013. Most tensor problems are NP-hard. *Journal of the ACM* 60(6):45:1–45:39.

Kampa, K.; Mehta, S.; Chou, C.; Chaovalitwongse, W.; and Grabowski, T. 2014. Sparse optimization in feature selection: application in neuroimaging. *Journal of Global Optimization* 59(2–3):439–457.

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

Liu, J.; Musialski, P.; Wonka, P.; and Ye, J. 2013. Tensor completion for estimating missing values in visual data. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 35(1):208–220.

Liu, J.; Ji, S.; and Ye, J. 2009. *SLEP: Sparse Learning with Efficient Projections*. Arizona State University.

Luo, L.; Xie, Y.; Zhang, Z.; and Li, W.-J. 2015. Support matrix machines. In *Proc. Int. Conf. on Machine Learning (ICML)*, 938–947.

Michel, V.; Gramfort, A.; Varoquaux, G.; Eger, E.; and Thirion, B. 2011. Total variation regularization for fMRI-based prediction of behavior. *IEEE Trans. on Medical Imaging* 30(7):1328–1340.

Mitchell, T. M.; Shinkareva, S. V.; Carlson, A.; Chang, K.-M.; Malave, V. L.; Mason, R. A.; and Just, M. A. 2008. Predicting human brain activity associated with the meanings of nouns. *Science* 320(5880):1191–1195.

Richard, E.; Savalle, P.-A.; and Vayatis, N. 2012. Estimation of simultaneously sparse and low rank matrices. In *Proc. Int. Conf. on Machine Learning*, 1351–1358.

Romera-Paredes, B.; Aung, M. H.; Bianchi-Berthouze, N.; and Pontil, M. 2013. Multilinear multitask learning. In *Proc. Int. Conf. on Machine Learning (ICML)*, 1444–1452.

Signoretto, M.; Dinh, Q. T.; De Lathauwer, L.; and Suykens, J. A. 2014. Learning with tensors: a framework based on convex optimization and spectral regularization. *Machine Learning* 94(3):303–351.

Signoretto, M.; De Lathauwer, L.; and Suykens, J. A. 2012. Nuclear norms for tensors and their use for convex multilinear estimation. Technical report.

Su, Y.; Gao, X.; Li, X.; and Tao, D. 2012. Multivariate multilinear regression. *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics* 42(6):1560–1573.

Suzuki, T. 2015. Convergence rate of Bayesian tensor estimatior and its minimax optimality. In *Proc. Int. Conf. on Machine Learning (ICML)*, 1273–1282.

Tan, X.; Zhang, Y.; Tang, S.; Shao, J.; Wu, F.; and Zhuang, Y. 2012. Logistic tensor regression for classification. In *Intelligent Science and Intelligent Data Engineering*. Springer. 573–581.

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

Tomioka, R., and Aihara, K. 2007. Classifying matrices with a spectral regularization. In *Proc. Int. Conf. on Machine Learning (ICML)*, 895–902.

Tomioka, R.; Hayashi, K.; and Kashima, H. 2010. Estimation of low-rank tensors via convex optimization. *arXiv preprint arXiv:1010.0789*.

Wang, Y.-X.; Xu, H.; and Leng, C. 2013. Provable subspace clustering: When LRR meets SSC. In *Advances in Neural Information Processing Systems (NIPS)*, 64–72.

Yuan, M., and Zhang, C.-H. 2014. On tensor completion via nuclear norm minimization. *Foundations of Computational Mathematics* 1–38.

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

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

Zou, H., and Hastie, T. 2005. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67(2):301–320.