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 $\ell_1$-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 $x \in \mathbb{R}^I$, 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 $(x_m, y_m)_{m=1}^M$ via a loss function $J(\cdot)$ plus a regularization function $\Omega(\cdot)$ formulated as follows:

$$\min_{w} \sum_{m=1}^{M} J(\langle x_m, w \rangle, y_m) + \Omega(w),$$  

where $w \in \mathbb{R}^I$ 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 $\ell_2$-norm regularization, leading to Lasso (Tibshirani 1996). Lasso gives a sparse $w$, 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

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

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 $\ell_1$-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 (Tomiioka and Aihara 2007) and generalized linear models (Zhou and Li 2014). It was also combined with $\ell_2$-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 $\ell_1$-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; Tomiioka, 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 × modality × 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 regularized multilinear regression and selection (Remurs) for tensor data. It is an extension of Lasso to tensor data using both tensor nuclear norm and ℓ₁-norm regularization. Gaiffas and Lecué (2011) pointed out that a mixture of nuclear norm and ℓ₁-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 ℓ₁-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., ť; matrices by uppercase boldface letters, e.g., À; and tensors by calligraphic letters, e.g., À. We denote their elements with indices in parentheses, and we follow the notations in (Kolda and Bader 2009) to denote vectors by lowercase the uppercase letter of the index, e.g., À. We denote their elements with indices in parentheses, and in (Tan et al. 2012). Thus, our model has a direct control on the unfolded matrices. Unfolding a tensor into matrices and could be explored in future work.

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 preserves 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.

Trade-off in Remurs. The penalty τ||W||∗ + γ||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 ℓ₁-norm with the smoothness of the ℓ₂-norm. Consequently, there is a trade-off between the tensor nuclear norm and the ℓ₁-norm. We analyze three cases of (5) below:

- When τ = 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 À and W, and represent them as ť and w respectively. Then (5) with τ = 0 can be rewritten as the Lasso problem: min_{Ŵ} \frac{1}{2} \sum_{m=1}^{M} (y_m - \langle À_m, Ŵ \rangle)^2 + γ||Ŵ||1 = min_{Ŵ} \frac{1}{2} \sum_{m=1}^{M} (y_m - \langle À_m, Ŵ \rangle)^2 + γ||Ŵ||1.

- When γ = 0, Remurs degenerates to a model enforcing only low rank, denoted as Remursγ=0. The second-order version of Remursγ=0 embodies the same penalty as in (Tomioka and Aihara 2007; Zhou and Li 2014;
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 \( \gamma/\tau \) 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 \( \ell_1 \)-norm first.

**Definition 1** (Singular value thresholding). Consider the SVD of a matrix \( A \in \mathbb{R}^{I_1 \times I_2} \), \( A = U \text{diag} [\sigma_j(A)] V^\top \), 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): \( \text{prox}_{\mu \| \cdot \|_n}(A) = U \text{diag} [\max(\sigma_j(A) - \mu, 0)] V^\top \).

**Definition 2** (Soft thresholding). Consider the \( \ell_1 \)-norm of a tensor \( A \in \mathbb{R}^{I_1 \times \cdots \times I_N} \) defined in Eq. (4). For \( \nu > 0 \), its proximity operator is

\[
\text{prox}_{\nu\| \cdot \|_1}(A) = \begin{cases} 
(A_{i_1, \ldots, i_N} - \nu) & \text{if } A_{i_1, \ldots, i_N} > \nu, \\
0 & \text{if } |A_{i_1, \ldots, i_N}| \leq \nu, \\
(A_{i_1, \ldots, i_N} + \nu) & \text{if } A_{i_1, \ldots, i_N} < -\nu.
\end{cases}
\]  

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

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

\[
\min_{U, V} \frac{1}{2} \sum_{m=1}^{M} (y_m - \langle X_m, U \rangle)^2 + \tau \|W\|_* + \gamma \|W\|_1 \\
s.t. \quad U = W.
\]  

(8)

To solve (8), we need to solve the part containing \( U \) and the part containing \( W \) independently. The summation of the two regularizers \( \tau \|W\|_* + \gamma \|W\|_1 \) on the same \( W \) 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. \( \tau \|V\|_* \) and \( \gamma \|W\|_1 \). However, it is still difficult to solve the part of \( \tau \|V\|_* \), because \( \tau \|V\|_* \) is defined in Eq. (6) as a summation of nuclear norm of interdependent matrices \( \{V_n\} \), which share the same entries and hence cannot be optimized independently. Therefore, we introduce \( N \) auxiliary tensors \( \{V_n, n = 1, \ldots, \} \) into (8) to obtain the following objective function where the proximity operator of each term is available:

\[
\min_{U, \{V_n\}, W} \frac{1}{2} \sum_{m=1}^{M} (y_m - \langle X_m, U \rangle)^2 + \tau \sum_{n=1}^{N} \|V_n\|_1 + \gamma \|W\|_1 \\
s.t. \quad U = W \quad \text{and} \quad V_n = W, n = 1, \ldots, N.
\]  

(9)

**Augmented Lagrangian and further splitting**

Integrating \( \ell_1 \)-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 \( U \) in (9) as the global variable, while we propose to use \( W \) instead for efficiency and sparsity, as \( W \) is sparse and much faster to compute. We provide the solution below.

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

\[
L_{\mu}(U, V_1, \ldots, V_N, W, P, Q_1, \ldots, Q_N) = \frac{1}{2} \sum_{m=1}^{M} (y_m - \langle X_m, U \rangle)^2 + \tau \sum_{n=1}^{N} \|V_{n(n)}\|_1 + \gamma \|W\|_1 \\
+ \langle P, U - W \rangle + \frac{\rho}{2} \|U - W\|_F^2 \\
+ \mu \sum_{n=1}^{N} \left( \langle Q_n, V_n - W \rangle + \frac{\rho}{2} \|V_n - W\|_F^2 \right),
\]  

(10)

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

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

\[
U^{k+1} = \arg \min_U \frac{1}{2} \sum_{m=1}^{M} (y_m - \langle X_m, U \rangle)^2 + \frac{\rho}{2} \|U - W^k + P^k\|_F^2.
\]  

(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., \( x_m = \text{vec}(X_m) \), and denote \( X = (x_1 \cdots x_M)^\top \) to get \( u^{k+1} = (X^\top X + \rho I)^{-1} (X^\top y + \rho (w^k - p^k)) \), where \( I \) is an identity matrix, \( w^k = \text{vec}(W^k) \), and \( p^k = \text{vec}(P^k) \). \( U^{k+1} \) can then be obtained by reshaping \( u^{k+1} \) into a tensor: \( U^{k+1} = \text{tensor}_N(u^{k+1}) \).

**Computing \( V^{k+1} \):** Fixing all the other variables, we calculate \( V^{k+1} \) based on Definition 1 as:

\[
V^{k+1} = \arg \min_{V_n} \frac{\tau}{N} \|V_{n(n)}\|_1 + \frac{\rho}{2} \|V_n - W^k + Q^k_{n(n)}\|_F^2 \\
= \text{fold}_n \left[ \text{prox}_{\frac{\tau}{N}\| \cdot \|_1}(W^k_{n(n)} - Q^k_{n(n)}) \right].
\]  

(12)

**Computing \( W^{k+1} \):** Fixing all the other variables, we express \( W \)-update as an averaging step using Definition 2:

\[
W^{k+1} = \arg \min_W \|W\|_1 + \frac{(N+1)\rho}{2} \|W - Z^{k+1}\|_F^2 = \text{prox}_{\frac{\gamma}{N+1}\| \cdot \|_1}(Z^{k+1}), \quad \text{where} \quad Z^{k+1} = \frac{U^{k+1} + \sum_{n=1}^{N} V^{k+1}_{n(n)}}{N+1} + \frac{\rho}{N+1}\sum_{n=1}^{N} Q'_{n(n)}.
\]

**Computing \( P^{k+1} \):** Fixing all the other variables, we express \( P \)-update as an averaging step using Definition 2:

\[
P^{k+1} = P^k + U^{k+1} - V^{k+1} \quad \text{and} \quad Q^k_{n+1} = Q^k_n + V^{k+1}_{n(n)} - W^{k+1}.
\]
and verify that the unaugmented Lagrangian key idea is to rewrite (9) into a two-block ADMM problem
Proof.

dual variable convergence of (9).

or TRR (Guo, Kotsia, and Patras 2012).

Algorithm 1 The Remurs algorithm based on ADMM.

1: Input: A set of $N$th-order tensor observations $\{X_m \in \mathbb{R}^{I_1 \times \cdots \times I_N}\}$, the corresponding responses $\{y_m\}$ $(m = 1, \ldots, M)$, the maximum number of iterations $K$, and parameters $\rho$, $\tau$, and $\gamma$.
2: Initialize: $U = V = W = p' = Q' = 0$.
3: for $k = 1$ to $K$ do
4: $u^{k+1} = (X^\top X + \rho I)^{-1}(X^\top y + \rho (w^k - p^k))$.
5: $L^{k+1} = \text{tensor}_N(u^{k+1})$.
6: for $n = 1$ to $N$ do
7: $\gamma^{k+1}_n = \text{fold}_n [\text{prox}_{\gamma_2\|\cdot\|_2} (W^{k}_n - Q^{k}_n(n))]$.
8: end for
9: $\rho^{k+1} = \text{prox}_{\gamma_1 + \gamma_2 \|\cdot\|_2} \left(\frac{L^{k+1} + p^k + \sum_{n=1}^N (\gamma^{k+1}_n + Q^{k}_n(n))}{N+1}\right)$.
10: $p^{k+1} = p^k + L^{k+1} - W^{k+1}$.
11: for $n = 1$ to $N$ do
12: $Q^{k+1}_n = Q^{k}_n + \gamma^{k+1}_n - W^{k+1}$.
13: end for
14: end for
15: Output: $W$.

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

Convergence analysis
As both nuclear norm and $\ell_1$-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 $\ell_1$-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): $y = (X, W) + \sigma e$, where $e$ is drawn from a standard normal distribution, $\sigma$ controls the signal-to-noise ratio, and $X \in \mathbb{R}^{I \times I}$.

We generate the true support as $I_k \times I_k \times \cdots \times I_k$ for $k$ dimensions, the maximum number of iterations $K$, the sparsity level of $W$ and $\sigma$, the percentage of zero entries, and the rank of $W$. The predictors are drawn from a multivariate Gaussian with zero mean and a covariance matrix where the correlation between $X(i, j)$ and $X(p, q)$ is $0.5 \sqrt{(1-p)^2 + (q-1)^2}$. We generate the true support as $W = W_d W_d^\top$, where $W_d \in \mathbb{R}^{I \times I}, d = 1, 2, R$ controls the rank of $W$. The sparsity level of $W$ is the percentage of zero entries, controlled by each entry of $W_d$ 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 $W$, both measured by the root-mean-squared error (RMSE). Five factors are varied to study the behaviors of each method: the feature size of $X$ ($I^2$), the rank of $W$ ($R$), the sparsity level of $W$ ($S$), the number of training samples ($M$), and the noise level ($\sigma$). When studying one of the factors, other factors are fixed to $I = 16$, $R = \frac{1}{3} I$, $S = 0.8$, $M = \frac{1}{2} I^2$, 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 $W$ 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 $\ell_1$-norm and the nuclear norm.
- As shown in Fig. 1(b), Lasso and ENet perform much better when the true $W$ 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).
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 \( \tau \) and \( \gamma \) are less than 10. Much lower prediction RMSEs can be achieved when \( \gamma \) for \( \ell_1 \)-norm is slightly smaller than \( \tau \) 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.

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, 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, \( \gamma \) and \( \tau \) 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/theoad-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.

<table>
<thead>
<tr>
<th>ID</th>
<th>Remurs</th>
<th>Remurs$_{\gamma=0}$</th>
<th>ENet</th>
<th>Lasso</th>
<th>orTRR</th>
<th>MMR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>95.00±0.06</td>
<td>92.50±0.07</td>
<td>93.33±0.06</td>
<td>92.50±0.07</td>
<td>80.00±0.16</td>
<td>61.67±0.09</td>
</tr>
<tr>
<td>2</td>
<td>83.33±0.05</td>
<td>84.17±0.07</td>
<td>78.33±0.05</td>
<td>77.50±0.07</td>
<td>69.17±0.13</td>
<td>60.83±0.09</td>
</tr>
<tr>
<td>3</td>
<td>82.50±0.08</td>
<td>82.50±0.08</td>
<td>77.50±0.05</td>
<td>79.17±0.07</td>
<td>74.17±0.13</td>
<td>62.50±0.03</td>
</tr>
<tr>
<td>4</td>
<td>91.67±0.05</td>
<td>90.83±0.09</td>
<td>91.67±0.05</td>
<td>90.00±0.05</td>
<td>70.00±0.12</td>
<td>55.00±0.14</td>
</tr>
<tr>
<td>5</td>
<td>62.50±0.09</td>
<td>65.00±0.08</td>
<td>63.33±0.09</td>
<td>55.83±0.05</td>
<td>52.50±0.12</td>
<td>48.33±0.10</td>
</tr>
<tr>
<td>6</td>
<td>79.17±0.09</td>
<td>72.50±0.07</td>
<td>80.00±0.07</td>
<td>79.17±0.05</td>
<td>68.33±0.13</td>
<td>55.83±0.09</td>
</tr>
<tr>
<td>7</td>
<td>74.17±0.11</td>
<td>76.67±0.09</td>
<td>70.83±0.13</td>
<td>70.00±0.05</td>
<td>67.50±0.07</td>
<td>63.33±0.09</td>
</tr>
<tr>
<td>8</td>
<td>61.67±0.15</td>
<td>62.50±0.17</td>
<td>53.33±0.13</td>
<td>55.83±0.05</td>
<td>50.83±0.06</td>
<td>45.83±0.09</td>
</tr>
<tr>
<td>9</td>
<td>73.33±0.06</td>
<td>76.67±0.05</td>
<td>70.83±0.06</td>
<td>74.17±0.05</td>
<td>72.50±0.17</td>
<td>51.67±0.14</td>
</tr>
</tbody>
</table>

Acc $\pm$ S  
78.15±0.08 78.15±0.09 75.46±0.08 74.91±0.06 68.33±0.12 56.11±0.10

The accuracy of all methods except GLTRM, 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 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 $\ell_1$-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


