Sturm: Sparse Tubal-Regularized Multilinear Regression for fMRI

Wenwen Li[1,4(✉)], Jian Lou[2], Shuo Zhou[1], and Haiping Lu[1,3]

[1] Department of Computer Science, The University of Sheffield, Sheffield S1 4DP, UK
{wenwen.li,szhou20,h.lu}@sheffield.ac.uk
[2] Department of Computer Science, Emory University, Atlanta 30322, USA
jian.lou@emory.edu
[3] Sheffield Institute for Translational Neuroscience, Sheffield S10 2HQ, UK
[4] Centre for Clinical Brain Sciences, The University of Edinburgh, Edinburgh EH16 4SB, UK
wenwen.li@ed.ac.uk

Abstract. While functional magnetic resonance imaging (fMRI) is important for healthcare/neuroscience applications, it is challenging to classify or interpret due to its multi-dimensional structure, high dimensionality, and small number of samples available. Recent sparse multilinear regression methods based on tensor are emerging as promising solutions for fMRI. Particularly, the newly proposed tensor singular value decomposition (t-SVD) sheds light on new directions. In this work, we study t-SVD for sparse multilinear regression and propose a **S**parse **tu**bal-**r**egularized **m**ultilinear regression (**Sturm**) method for fMRI. Specifically, the Sturm model performs multilinear regression with two regularization terms: a tubal tensor nuclear norm based on t-SVD and a standard ℓ_1 norm. An optimization algorithm under the alternating direction method of multipliers framework is derived for solving the Sturm model. We then perform experiments on four classification problems, including both resting-state fMRI for disease diagnosis and task-based fMRI for neural decoding. The results show the superior performance of Sturm in classifying fMRI using just a small number of voxels.

1 Introduction

Brain diseases affect millions of people worldwide and impose significant challenges to healthcare systems. Functional magnetic resonance imaging (fMRI) is a key brain imaging technique for diagnosis, monitoring and treatment of brain diseases. Beyond healthcare, fMRI is also an indispensable tool in neuroscience studies [5].

Facing the challenging "large p (brain voxels) small n (samples)" problem in brain imaging, sparse learning models [9,11] are found to be attractive on

W. Li and J. Lou—These authors contributed equally to this work.

H.-I. Suk et al. (Eds.): MLMI 2019, LNCS 11861, pp. 256–264, 2019.
https://doi.org/10.1007/978-3-030-32692-0_30

fMRI data because they can reveal the direct dependency of a response (e.g., diagnosis outcome or brain states) on a small portion of features i.e., brain voxels. Recently, tensor-based sparse multilinear regression methods are emerging as a promising direction, where *tensor* refers to multidimensional array. For example, a 3D fMRI volume can be seen as a 3D tensor or a third-order tensor.

Tensor-based sparse multilinear regression models relate a feature tensor with a univariate response via a coefficient tensor, generalizing Lasso-based models [4] to tensor data. Regularization that promotes sparsity and *low rankness* is also generalized to the coefficient tensor. For example, the regularized multilinear regression and selection (Remurs) model [12] incorporates a sparse regularization term, via an ℓ_1 norm, and a *Tucker rank*-minimization term [13], via a summation of the nuclear norms (SNN) of unfolded matrices with application to task-based fMRI data. A new Tubal Tensor Nuclear Norm (TNN) [14] has recently been proposed based on the *tubal rank*, which originates from the tensor singular value decomposition (t-SVD) [6]. In this work, we study sparse multilinear regression under the t-SVD framework for fMRI classification. The success of TNN was limited to *unsupervised* learning settings such as completion/recovery and robust PCA [2]. To our knowledge, TNN has not been studied in a *supervised* setting yet, such as multilinear regression for predicting a response based on a set of tensor-structural samples. Moreover, the targeted fMRI classification tasks have the challenge of small sample size (relative to the feature dimension).

Our contributions are twofold: (1) We propose a **S**parse **tu**bal-**r**egularized **m**ultilinear regression (**Sturm**) method that incorporates TNN regularization and a sparsity regularization on the coefficient tensor; (2) We evaluate Sturm and related methods on *both* resting-state and task-based fMRI classification problems, instead of only one of them as in previous work [3].

2 Method

Notations. We use lowercase, bold lowercase, bold uppercase, calligraphic uppercase letters to denote scalar, vector, matrix, and tensor, respectively. A third-order tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ is addressed by three indices $\{i_n\}$, $n = 1, 2, 3$. Each i_n usually addresses the nth mode of $\mathcal{A}$. For example, in a 3D fMRI volume, I_1, I_2 and I_3 could indicate the sagittal, coronal and axial dimension, respectively.

2.1 Tubal Rank and Tubal Tensor Nuclear Norm (TNN)

Tubal rank is derived from the t-SVD as illustrated in Fig. 1, where $\mathcal{U} \in \mathbb{R}^{I_1 \times I_1 \times I_3}$, $\mathcal{V} \in \mathbb{R}^{I_2 \times I_2 \times I_3}$ are orthogonal tensors, and $\mathcal{S} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ is an *f-diagonal* tensor. The number of nonzero singular tubes (along the diagonal direction) of $\mathcal{S}$ is defined as the tubal rank.

t-SVD can be computed efficiently via the discrete Fourier transfer (DFT). Denote the Fourier transformed tensor $\mathcal{A}$ as $\mathcal{A}_\mathcal{F}$, $\mathcal{A}_\mathcal{F} = \texttt{fft}(\mathcal{A}, [\,], 3)$. As a convex relaxation for the tubal rank, the TNN of $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ is defined as $\|\mathcal{A}\|_{TNN} = \frac{1}{I_3} \sum_{i_3=1}^{I_3} \|\mathbf{A}_\mathcal{F}^{(i_3)}\|_*$, where $\|\cdot\|_*$ denotes the matrix nuclear norm. More about t-SVD and TNN can be found in [14].

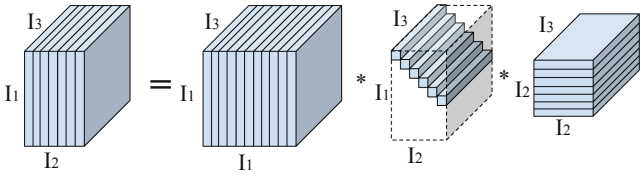


Fig. 1. Illustration of t-SVD $\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^\top$, $*$ means tensor product [6], assuming $I_1 > I_2$. $\mathcal{U}$ and $\mathcal{V}$ are orthogonal tensors. $\mathcal{S}$ is an f-diagonal tensor. Tubal rank is the number of nonzero singular tubes of $\mathcal{S}$. In this example, the tubal rank is 6.

2.2 The Sturm Model

Regularized Multilinear Regression Model. This approach trains a model from M pairs of feature tensor and associated response label, $(\mathcal{X}_m \in \mathbb{R}^{I_1 \times I_2 \times I_3}, y_m)$ with $m = 1, ..., M$, to relate them via a *coefficient tensor* $\mathcal{W} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ as

$$\min_{\mathcal{W}} \frac{1}{M} \sum_{m=1}^{M} L(\langle \mathcal{X}_m, \mathcal{W} \rangle, y_m) + \lambda \Omega(\mathcal{W}), \tag{1}$$

where $L(\cdot)$ is a loss function, $\Omega(\cdot)$ is a regularization term, λ is a balancing hyper-parameter, and $\langle \mathcal{X}, \mathcal{W} \rangle$ denotes the inner product (a.k.a. the scalar product) of two tensors of the same size:

$$\langle \mathcal{X}, \mathcal{W} \rangle := \sum_{i_1} \sum_{i_2} \sum_{i_3} \mathcal{X}(i_1, i_2, i_3) \cdot \mathcal{W}(i_1, i_2, i_3). \tag{2}$$

The State-of-the-Art Model. The Remurs [12] model has been successfully applied to task-based fMRI data. It uses a conventional least square loss function and assumes $\mathcal{W}$ to be both sparse and low rank. The sparsity of $\mathcal{W}$ is regularized by an ℓ_1 norm and the low rank by an SNN norm. However, the SNN requires unfolding $\mathcal{W}$ into matrices, susceptible to losing some higher-order structural information. Moreover, it has been pointed out in [10] that SNN is not a tight convex relaxation of its target rank.

A New Model. The limitation of SNN motivates us to propose a **S**parse **tu**bal-**r**egularized **m**ultilinear regression (**Sturm**) model which replaces SNN in Remurs with TNN. This leads to the following objective function

$$\min_{\mathcal{W}} \frac{1}{2} \sum_{m=1}^{M} (y_m - \langle \mathcal{X}_m, \mathcal{W} \rangle)^2 + \tau \|\mathcal{W}\|_{TNN} + \gamma \|\mathcal{W}\|_1, \tag{3}$$

where τ and γ are hyperparameters, and $\|\mathcal{W}\|_1$ is the ℓ_1 norm of tensor $\mathcal{W}$, defined as $\|\mathcal{W}\|_1 = \sum_{i_1} \sum_{i_2} \sum_{i_3} |\mathcal{W}(i_1, i_2, i_3)|$, which is equivalent to the ℓ_1 norm of its vectorized representation $\mathbf{w}$. Here, the TNN regularization term $\|\mathcal{W}\|_{TNN}$ enforces low tubal rank in $\mathcal{W}$.

Optimization Algorithm for Sturm. ADMM [1] is a standard solver for Problem (3). Thus, we derive an ADMM algorithm to optimize the Sturm objective function. We begin with introducing two auxiliary variables, $\mathcal{A}$ and $\mathcal{B}$ to disentangle the TNN and ℓ_1-norm regularization:

$$\min_{\mathcal{W}} \frac{1}{2}\sum_{m=1}^{M}(y_m - \langle \mathcal{X}_m, \mathcal{A}\rangle)^2 + \tau\|\mathcal{B}\|_{TNN} + \gamma\|\mathcal{W}\|_1, s.t.\ \mathcal{A} = \mathcal{W} \text{ and } \mathcal{B} = \mathcal{W}. \quad (4)$$

Then, we introduce two Lagrangian dual variables $\mathcal{P}$ (for $\mathcal{A}$) and $\mathcal{Q}$ (for $\mathcal{B}$). With a Lagrangian constant ρ, the augmented Lagrangian becomes,

$$\begin{aligned} L_\rho(\mathcal{A}, \mathcal{B}, \mathcal{W}, \mathcal{P}, \mathcal{Q}) &= \frac{1}{2}\sum_{m=1}^{M}(y_m - \langle \mathcal{X}_m, \mathcal{A}\rangle)^2 + \tau\|\mathcal{B}\|_{TNN} + \gamma\|\mathcal{W}\|_1 \\ &+ \left\langle \mathcal{P}, \mathcal{A} - \mathcal{W}\right\rangle + \frac{\rho}{2}\|\mathcal{A} - \mathcal{W}\|_F^2 + \left\langle \mathcal{Q}, \mathcal{B} - \mathcal{W}\right\rangle + \frac{\rho}{2}\|\mathcal{B} - \mathcal{W}\|_F^2. \end{aligned} \quad (5)$$

where $\|\cdot\|_F$ is the Frobenius norm defined as $\|\mathcal{T}\|_F = \sqrt{\langle \mathcal{T}, \mathcal{T}\rangle}$ using Eq. (2). We further introduce two scaled dual variables $\mathcal{P}' = \frac{1}{\rho}\mathcal{P}$ and $\mathcal{Q}' = \frac{1}{\rho}\mathcal{Q}$ only for notational convenience. Next, we derive the update from iteration k to $k+1$ by minimizing one variable with all the other variables fixed.

Updating $\mathcal{A}^{k+1}$:

$$\mathcal{A}^{k+1} = \arg\min_{\mathcal{A}} \frac{1}{2}\sum_{m=1}^{M}(y_m - \langle \mathcal{X}_m, \mathcal{A}\rangle)^2 + \frac{\rho}{2}\|\mathcal{A} - \mathcal{W}^k + \mathcal{P}'^k\|_F. \quad (6)$$

This can be rewritten as a linear-quadratic objective function by vectorizing all the tensors. Specifically, let $\mathbf{a} = \texttt{vec}(\mathcal{A})$, $\mathbf{w}^k = \texttt{vec}(\mathcal{W}^k)$, $\mathbf{p}'^k = \texttt{vec}(\mathcal{P}'^k)$, $\mathbf{y} = [y_1 \cdots y_M]^\top$, $\mathbf{x}_m = \texttt{vec}(\mathcal{X}_m)$, and $\mathbf{X} = [\mathbf{x}_1 \cdots \mathbf{x}_M]^\top$. Then we get an equivalent objective function with the following solution:

$$\mathbf{a}^{k+1} = (\mathbf{X}^\top\mathbf{X} + \rho\mathbf{I})^{-1}(\mathbf{X}^\top\mathbf{y} + \rho(\mathbf{w}^k - \mathbf{p}'^k)), \quad (7)$$

where $\mathbf{I}$ is an identity matrix. $\mathcal{A}^{k+1}$ is obtained by folding (reshaping) $\mathbf{a}^{k+1}$ into a third-order tensor, denoted as $\mathcal{A}^{k+1} = \texttt{tensor}_3(\mathbf{a}^{k+1})$.

Updating $\mathcal{B}^{k+1}$:

$$\mathcal{B}^{k+1} = \arg\min_{\mathcal{B}} \tau\|\mathcal{B}\|_{TNN} + \frac{\rho}{2}\|\mathcal{B} - \mathcal{W}^k + \mathcal{Q}'^k\|_F^2 = \texttt{prox}_{\frac{\tau}{\rho}\|\cdot\|_{TNN}}(\mathcal{W}^k - \mathcal{Q}'^k). \quad (8)$$

The proximal operator for the TNN at tensor $\mathcal{T}$ with parameter μ is denoted by $\texttt{prox}_{\mu\|\cdot\|_{TNN}}(\mathcal{T})$ and defined as

$\texttt{prox}_{\mu\|\cdot\|_{TNN}}(\mathcal{T}) := \arg\min_{\mathcal{W}} \mu\|\mathcal{W}\|_{TNN} + \frac{1}{2}\|\mathcal{W} - \mathcal{T}\|_F^2$. More details of the TNN proximal operator can be found in [14].

Updating $\mathcal{W}^{k+1}$:

$$\mathcal{W}^{k+1} = \texttt{prox}_{\frac{\gamma}{2\rho}\|\cdot\|_1}(\mathcal{A}^{k+1} + \mathcal{P}'^k + \mathcal{B}^{k+1} + \mathcal{Q}'^k)/2. \quad (9)$$

It can be solved by calling the proximal operator of the ℓ_1 norm with parameter $\frac{\gamma}{2\rho}$, which is simply the element-wise soft-thresholding, i.e., $\texttt{prox}_{\mu\|\cdot\|_1}(\mathcal{T}) = sign(\mathcal{T})(|\mathcal{T}| - \mu)_+$.

Updating $\mathcal{P}^{k+1}$ and $\mathcal{Q}^{k+1}$:

$$\mathcal{P}'^{k+1} = \mathcal{P}'^{k} + \mathcal{A}^{k+1} - \mathcal{W}^{k+1}, \text{ and } \mathcal{Q}'^{k+1} = \mathcal{Q}'^{k} + \mathcal{B}^{k+1} - \mathcal{W}^{k+1}. \tag{10}$$

Optimization Algorithm Analysis. *Sturm* is an instance of the ADMM algorithm applied to linear constraint convex optimization problem, which is guaranteed to converge with a $\frac{1}{\epsilon}$ convergence rate [1].

3 Experiments and Discussion

3.1 Classification Problems and Datasets

Resting-State fMRI for Disease Diagnosis. We use two freely available datasets. *Rest 1 – $ABIDE_{NYU\&UM}$*: the two largest subsets NYU and UM from the Autism Brain Imaging Data Exchange (ABIDE)[1], which consists of 101 patients with autism spectrum disorder (ASD) and 131 healthy control subjects. *Rest 2 – $ADHD\text{-}200_{NYU}$*: the NYU subset from the Attention Deficit Hyperactivity Disorder (ADHD) 200 dataset[2] with 118 ADHD patients and 98 healthy controls. The 4D raw fMRI data is reduced to 3D by either taking the average or the amplitude of low frequency fluctuation of voxel values along the time dimension. We perform experiments on both and report the best results.

Task-Based fMRI for Neural Decoding. We consider four datasets from the OpenfMRI[3] in two binary classification problems. *Task 1 – Balloon vs Mixed Gamble*, two gamble-related datasets with 64 subjects in total; and *Task 2 – Simon vs Flanker*, two recognition and response related tasks with overall 94 subjects. The OpenfMRI data is processed with a standard template following [8] to obtain the 3D statistical parametric maps (SPMs) for each brain condition.

3.2 Algorithms and Evaluation Settings

Algorithms. Sturm and Sturm + SVM (support vector machine) are compared against the following four algorithms and three additional algorithms combining with SVM.

- *SVM*: a linear SVM is chosen for both speed and accuracy consideration.
- *Lasso* [4]: a linear regression method with the ℓ_1 norm regularization.
- *Elastic Net (ENet)* [15]: a linear regression method with ℓ_1 and ℓ_2 norm.
- *Remurs* [12]: a multilinear regression model with ℓ_1 norm and Tucker rank-based SNN regularization.

[1] http://fcon_1000.projects.nitrc.org/indi/abide.
[2] http://neurobureau.projects.nitrc.org/ADHD200/Data.html.
[3] https://legacy.openfmri.org.

SVM, Lasso, and ENet take vectorized fMRI data as input while Remurs directly takes 3D fMRI tensors as input. In addition, Lasso, ENet, Remurs, and Sturm can also be used for feature selection. Hence, we can use the selected voxels from each of the above algorithm as input to SVM for classification i.e., Lasso + SVM, ENet + SVM, Remurs + SVM and Sturm + SVM.

Model Hyper-parameter Tuning. For Sturm, we follow the Remurs default setting [12] to set ρ to 1 and use the same set $\{10^{-3}, 5 \times 10^{-3}, 10^{-2}, \ldots, 5 \times 10^{2}, 10^{3}\}$ for τ and γ, additionally scaling the first term in Eq. (3) by a factor $\alpha = \sqrt{(\max(I_1, I_2) \times I_3)}$ to better balance the scales of the loss function and regularization terms [7]. Ten-fold cross validation is applied for tuning hyper-parameters in all the algorithms.

Image Resizing. To improve computational efficiency and reduce the small sample size problem (and overfitting), the input 3D tensors are further re-sized into three different sizes with a factor β, choosing from $\{0.3, 0.5, 0.7\}$. The best accuracy from all three sizes is reported for each algorithm in this paper.

Feature Selection. In Lasso + SVM, ENet + SVM, Remurs + SVM, and Sturm + SVM, we rank the selected features by their associated absolute values of $\mathcal{W}$ in the descending order and feed the top $\eta\%$ of the features to SVM. We study five values of η: $\{1, 5, 10, 50, 100\}$ and report the best accuracy for each algorithm.

Evaluation Metric and Method. The classification accuracy is our primary evaluation metric, and we also examine the sparsity of the obtained solutions for all algorithms except SVM. The sparsity is calculated as the ratio of the number of zeros in the output coefficient tensor $\mathcal{W}$ to its size $I_1 \times I_2 \times I_3$. In general, higher sparsity implies better interpretability [4].

Table 1. Classification accuracy (mean ± standard deviation in %). Rest 1 and Rest 2 denote two disease diagnosis problems on $\text{ABIDE}_{NYU\&UM}$ and ADHD-200, respectively. Task 1 and Task 2 denote two neural decoding problems on OpenfMRI datasets for Balloon vs Mixed gamble and Simon vs Flanker, respectively. The best accuracy among all of the compared algorithms for each column is highlighted in **bold** and the second best is underlined.

Method	Rest 1	Rest 2	Task 1	Task 2	Average		
					Rest	Task	All
SVM	60.78 ± 0.09	63.97 ± 0.09	87.38 ± 0.12	82.56 ± 0.17	62.38	84.97	73.67
Lasso	61.16 ± 0.08	64.84 ± 0.11	87.38 ± 0.12	85.22 ± 0.07	63.00	86.30	74.65
ENet	61.21 ± 0.10	64.38 ± 0.10	81.19 ± 0.15	82.56 ± 0.17	62.80	81.87	72.34
Remurs	60.72 ± 0.08	62.13 ± 0.09	87.14 ± 0.13	84.67 ± 0.15	61.43	85.90	73.67
Sturm	62.05 ± 0.11	63.47 ± 0.07	**89.10 ± 0.09**	**86.89 ± 0.16**	62.76	**88.00**	**75.38**
Lasso + SVM	63.37 ± 0.08	62.56 ± 0.09	74.05 ± 0.20	72.11 ± 0.16	62.97	73.08	68.02
ENet + SVM	64.20 ± 0.07	61.61 ± 0.08	76.43 ± 0.14	72.00 ± 0.14	62.91	74.21	68.56
Remurs + SVM	**64.67 ± 0.10**	60.23 ± 0.10	81.19 ± 0.12	83.56 ± 0.19	62.45	82.37	72.41
Sturm+ SVM	64.66 ± 0.12	**66.24 ± 0.06**	78.10 ± 0.22	82.44 ± 0.16	**65.45**	80.27	72.86

Table 2. Sparsity (mean ± standard deviation) for respective results in Table 1 with the **best** and second best highlighted.

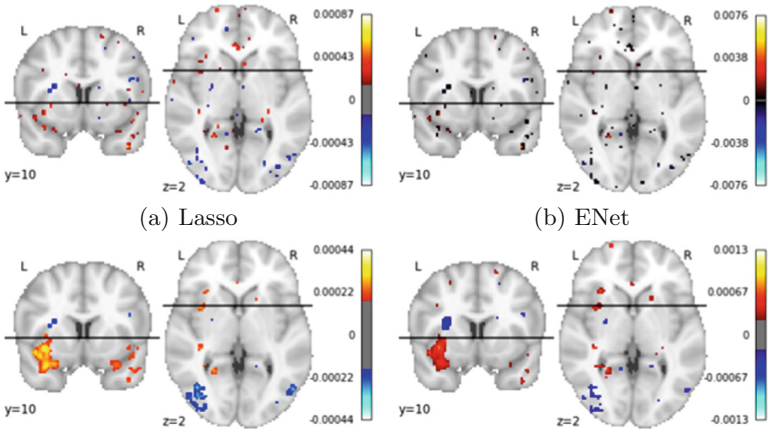
Method	Rest 1	Rest 2	Task 1	Task 2	Average		
					Rest	Task	All
Lasso	0.52 ± 0.09	0.23 ± 0.32	0.74 ± 0.12	0.73 ± 0.01	0.38	0.73	0.55
ENet	0.60 ± 0.01	0.01 ± 0.01	**0.96 ± 0.05**	**0.95 ± 0.03**	0.31	**0.96**	0.63
Remurs	0.69 ± 0.03	0.73 ± 0.17	0.81 ± 0.08	0.81 ± 0.07	0.71	0.81	0.76
Sturm	0.86 ± 0.18	0.86 ± 0.24	0.72 ± 0.24	0.60 ± 0.15	0.86	0.66	0.76
Lasso + SVM	0.57 ± 0.05	0.19 ± 0.40	0.77 ± 0.10	0.75 ± 0.06	0.38	0.76	0.57
ENet + SVM	0.58 ± 0.09	0.02 ± 0.01	**0.96 ± 0.04**	**0.95 ± 0.04**	0.30	**0.96**	0.63
Remurs + SVM	0.70 ± 0.13	0.74 ± 0.17	0.80 ± 0.04	0.79 ± 0.13	0.72	0.79	0.76
Sturm + SVM	**0.87 ± 0.07**	**0.99 ± 0.01**	0.85 ± 0.14	0.56 ± 0.11	**0.93**	0.71	**0.82**

3.3 Result Discussion and Visualization

Classification Accuracy. Table 1 shows the classification accuracy for all algorithms. Over the two resting-state problems, Sturm + SVM has the highest accuracy of 65.45%, and Lasso is the second-best (63.00%). Whereas, on these two task-based problems, Sturm has outperformed all other algorithms in accuracy, with 88.00% accuracy. Lasso is again the second-best in accuracy (86.30%). One interesting observation is that on task-based classification problems, Sturm + SVM has significant drop in accuracy compared with Sturm alone, and Lasso + SVM, ENet + SVM and Remurs + SVM all have lower accuracy compared to without SVM. Overall, Sturm still achieves the highest accuracy.

Model Sparsity. Table 2 presents the respective sparsity values except SVM, which uses all features so the sparsity is zero. Over the two resting-state classification problems, Sturm + SVM and Sturm are the top two algorithms with sparsity of 0.93 and 0.86, respectively. Noticeably, Lasso & ENet fail to select a sparse solution via cross validation on Rest 2. Over the two task-based problems, both ENet and ENet + SVM have the best sparsity (0.96). However, both of them have much lower accuracy than Sturm and Remurs. Over all the problems, Sturm + SVM obtains the most sparse model with the overall sparsity of 0.82, meanwhile, Sturm and Remurs have almost identical sparsity of 0.76.

Weight Map Visualization. Take Task Simon vs Flanker as an example. The selected voxels on two brain slices from the top 5% (4514 voxels, ranked by the absolute weights) of the full-brain voxels are visualized in Fig. 2 using the best performing parameters in cross-validation of Lasso, ENet, Remurs and Sturm. Figure 2 clearly shows that Sturm and Remurs select more spatially connected voxels than Lasso or ENet does, resulting in more robust and easier interpreted models for further analysis. Both Sturm and Remurs choose voxels from highly consistent regions, while Sturm produces weight maps with slightly higher visual contrast than Remurs does.



(a) Lasso (b) ENet

(c) Remurs (d) Sturm

Fig. 2. Weight maps on Task Simon vs Flanker. Sturm and Remurs select clear regions which are easier for further analysis than Lasso and ENet.

4 Conclusion

The proposed multilinear regression model (*Sturm*) performs regression with regularization on the tubal tensor nuclear norm (TNN), demonstrates the superior overall performance of Sturm (and Sturm + SVM, in some cases) over other methods in terms of accuracy, sparsity and weight maps. Thus, it is promising to use TNN and tubal rank regularization in a supervised setting for fMRI.

Acknowledgement. This work was supported by the grant from the UK Engineering and Physical Sciences Research Council (EP/R014507/1).

References

1. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends® Mach. Learn. **3**(1), 1–122 (2011)
2. Candès, E.J., Li, X., Ma, Y., Wright, J.: Robust principal component analysis? J. ACM (JACM) **58**(3), 11 (2011)
3. Eickenberg, M., Dohmatob, E., Thirion, B., Varoquaux, G.: Grouping total variation and sparsity: statistical learning with segmenting penalties. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9349, pp. 685–693. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24553-9_84
4. Hastie, T., Tibshirani, R., Wainwright, M.: Statistical Learning with Sparsity: The Lasso and Generalizations. CRC Press, Boca Raton (2015)
5. Huettel, S.A., Song, A.W., McCarthy, G.: Functional Magnetic Resonance Imaging, vol. 1. Sinauer Associates, Sunderland (2004)

6. Kilmer, M.E., Braman, K., Hao, N., Hoover, R.C.: Third-order tensors as operators on matrices: a theoretical and computational framework with applications in imaging. SIAM J. Matrix Anal. Appl. **34**(1), 148–172 (2013)
7. Lu, C., Feng, J., Liu, W., Lin, Z., Yan, S.: Tensor robust principal component analysis with a new tensor nuclear norm. IEEE Trans. PAMI (2019)
8. Poldrack, R.A., Barch, D.M., Mitchell, J., et al.: Toward open sharing of task-based fMRI data: the OpenfMRI project. Front. Neuroinf. **7**, 12 (2013)
9. Rao, N., Cox, C., Nowak, R., et al.: Sparse overlapping sets lasso for multitask learning and its application to fMRI analysis. In: NeurIPS, pp. 2202–2210 (2013)
10. Romera-Paredes, B., Pontil, M.: A new convex relaxation for tensor completion. In: NeurIPS, pp. 2967–2975 (2013)
11. Ryali, S., Supekar, K., Abrams, D.A., Menon, V.: Sparse logistic regression for whole-brain classification of fmri data. NeuroImage **51**(2), 752–764 (2010)
12. Song, X., Lu, H.: Multilinear regression for embedded feature selection with application to fMRI analysis. In: AAAI Conference on Artificial Intelligence (2017)
13. Tucker, L.R.: Some mathematical notes on three-mode factor analysis. Psychometrika **31**(3), 279–311 (1966)
14. Zhang, Z., Aeron, S.: Exact tensor completion using t-SVD. IEEE Trans. Sig. Process. **65**(6), 1511–1526 (2017)
15. Zou, H., Hastie, T.: Regularization and variable selection via the elastic net. J. Roy. Stat. Soc.: Ser. B (Stat. Methodol.) **67**(2), 301–320 (2005)