Classification Based on Neuroimaging Data by Tensor Boosting

Bo Zhang*, Hua Zhou†, Liwei Wang‡ and Chul Sung§
*IBM, Durham, NC 27703, USA
Email: bozhang@us.ibm.com
†University of California, Los Angeles, CA 90095, USA
‡Pharmaceutical Product Development Inc., Morrisville, NC 27560, USA
§IBM, Austin, TX 78758, USA

***Abstract*—Recent advances in medical imaging technologies generate a high volume of imaging data. Classification of cognitive outcome and disease status based on brain images is one of the most important tasks in neuroimaging studies. However it poses great challenge to the current classification methods due to the extremely high dimensionality and low signal to noise ratio in brain image data. In this article we propose a tensor boosting algorithm for classification based on neuroimaging data. The method is off-the-shelf, computationally simple and amenable to various modalities of neuroimaging data. The method is applied to an EEG data set from an alcoholism study and an MRI data set from an ADHD Global Competition and shows significantly improved classification performance.**

I. Introduction

Modern technology is producing an enormous amount of medical image data, such as electroencephalography (EEG), anatomical magnetic resonance imaging (MRI) and functional magnetic resonance imaging (fMRI). A common task in neuroimaging studies is to classify subjects into different cognitive outcomes and disease status based on brain images. There is much hope that the huge amount of information inherent in these images brings discriminant power in classification, although it still remains an active research area how this can be achieved. Medical imaging data are often in the form of multidimensional arrays, also known as tensors. This imposes great challenge to the traditional classifiers. For instance, typical anatomical MRI image of size $128\times128\times128$ contains $128^3 = 2,097,152$ voxels. Both computability and performance of commonly used classifiers are compromised by this ultra-high dimensionality. More seriously, traditional classifiers take a vector of features as input. Vectorizing an array before applying classification algorithm destroys the inherent spatial structure of the image that possesses wealth of information.

Typical classification methods in the current neuroimaging literature consist of three components: feature extraction, feature dimensionality reduction and feature-based classification. A few or more effective features are first extracted from the original image data. This step often requires substantial domain knowledge. Once the features have been defined and extracted, dimension reduction may be necessary to further reduce the dimensionality, by principal component analysis (PCA) [1] [2] [3], independent component analysis (ICA) [4] [5], or their variants [6]. The final summary features are then fed into commonly used classifiers, such as Fisher linear discrimination analysis (LDA) [1] [2], quadratic discriminant analysis (QDA) [6] [7], logistic regression [8], or Projection Pursuit [5].

Although these methods have been widely used in practice, a feature selection step needs to be conducted first. Combining the feature selection and feature-based classification in a unified framework could avoid the step of selecting the right set of features. In this paper, we propose a tensor LogitBoost method, which organically combines two methods. The first one is boosting, which was called the "best off-the-shelf classifier in the world" [9]. The second one is the recently proposed tensor regression method [10], which handles the tensor-valued data efficiently in the regression framework. One advantage of the proposed tensor LogitBoost method lies in its ability to keep the tensor structure during the classification stage. Also, respecting tensor structure in the tensor regression step, tensor LogitBoost improves the performance of regular LogitBoost. It is "off-the-shelf" and thus applies to various modalities of image data, which are all in the form of matrix or tensor. Also, it is computationally simple due to an efficient estimation algorithm for the tensor regression.

The rest of the paper is organized as follows. In Section II, we first introduce some notations. In Section III, we briefly review two essential components of our algorithm, boosting and tensor regression. We then present our new method of tensor LogitBoost in detail. Tensor PCA and tuning strategy for tensor boosting are also discussed in this section. In Section IV and Section V, tensor LogitBoost is applied to the analysis of the EEG data and MRI data example for further illustration. We conclude the chapter in Section VI with a discussion on future extensions.

II. Notation

Given two matrices $\boldsymbol{A} = [\boldsymbol{a}_1 \dots \boldsymbol{a}_n] \in \mathbb{R}^{m\times n}$ and $\boldsymbol{B} = [\boldsymbol{b}_1 \dots \boldsymbol{b}_q] \in \mathbb{R}^{p\times q}$, if $\boldsymbol{A}$ and $\boldsymbol{B}$ have the same number of columns $n = q$, then the *Khatri-Rao* product is defined as the $mp \times n$ columnwise *Kronecker product*

$$\boldsymbol{A} \odot \boldsymbol{B} = [\boldsymbol{a}_1 \otimes \boldsymbol{b}_1 \ \dots \ \boldsymbol{a}_n \otimes \boldsymbol{b}_n].$$

978-1-5090-6182-2/17/$31.00 ©2017 IEEE

The *mode-d matricization*, $\boldsymbol{B}_{(d)} \in \mathbb{R}^{p_d \times \prod_{d' \neq d} p_{d'}}$, is a matrix with columns the mode-d fibers of $\boldsymbol{B}$. More precisely, the $(i_1, \ldots, i_D)$ element of the tensor $\boldsymbol{B}$ maps to the (i_d, j) element of the matrix $\boldsymbol{B}_{(d)}$, where $j = 1 + \sum_{d' \neq d}(i_{d'} - 1) \prod_{d'' < d', d'' \neq d} p_{d''}$. With $d = 1$, we observe that $\mathrm{vec}\, \boldsymbol{B}$ is the same as vectorizing the mode-1 matricization $\boldsymbol{B}_{(1)}$, where vec is the vectorization operator that stacks the columns of a matrix below one another into a vector.

An *outer product* of D vectors $\boldsymbol{b}_d \in \mathbb{R}^{p_d}$, $d = 1, \ldots, D$, $\boldsymbol{b}_1 \circ \boldsymbol{b}_2 \circ \cdots \circ \boldsymbol{b}_D$ is a $p_1 \times \cdots \times p_D$ tensor with entries $(\boldsymbol{b}_1 \circ \boldsymbol{b}_2 \circ \cdots \circ \boldsymbol{b}_D)_{i_1 \cdots i_D} = \prod_{d=1}^{D} b_{d i_d}$.

We say a tensor $\boldsymbol{B} \in \mathbb{R}^{p_1 \times \cdots \times p_D}$ admits a CANDECOMP/PARAFAC decomposition if

$$\begin{aligned} \boldsymbol{B} &= \sum_{r=1}^{R} \boldsymbol{\beta}_1^{(r)} \circ \cdots \circ \boldsymbol{\beta}_D^{(r)} \quad (1) \\ &= [\![\boldsymbol{B}_1, \ldots, \boldsymbol{B}_D]\!], \quad (2) \end{aligned}$$

where $\boldsymbol{\beta}_d^{(r)} \in \mathbb{R}^{p_d}, d = 1, \ldots, D, r = 1, \ldots, R$, are all column vectors and $\boldsymbol{B}_d = [\boldsymbol{\beta}_d^{(1)}, \ldots, \boldsymbol{\beta}_d^{(R)}] \in \mathbb{R}^{p_d \times R}$, $d = 1, \ldots, D$.

The mode-d matricization of $\boldsymbol{B}$ admitting CANDECOMP/PARAFAC decomposition (2) can be expresses as

$$\boldsymbol{B}_{(d)} = \boldsymbol{B}_d (\boldsymbol{B}_D \odot \cdots \odot \boldsymbol{B}_{d+1} \odot \boldsymbol{B}_{d-1} \odot \cdots \odot \boldsymbol{B}_1).$$

Also note that

$$\mathrm{vec}\boldsymbol{B} = \mathrm{vec}\boldsymbol{B}_{(1)} = (\boldsymbol{B}_D \odot \cdots \odot \boldsymbol{B}_1)\mathbf{1}_R.$$

III. Method

There are multiple versions of boosting methods developed for different application scenarios. The point of view of the gradient boosting machine by [11] is particularly productive and drives derivations of many variants of boosting algorithm. See [12] for a good review.

The LogitBoost algorithm can be treated as a "statistical" version of the well known AdaBoost [13] because it learns the set of regression functions $\{f_l(\boldsymbol{x})\}_{l=1,\ldots,L}$ by minimizing the negative binomial log-likelihood instead of the exponential loss. Consider a two-class classification problem with the vector-valued predictors $\boldsymbol{x}_i \in \mathbb{R}^p$ and class labels $y_i \in \{0, 1\}$. The probability of $\boldsymbol{x}$ being in class 1 is represented by

$$p(\boldsymbol{x}) = \frac{e^{F(\boldsymbol{x})}}{1 + e^{F(\boldsymbol{x})}},$$

where $F(\boldsymbol{x}) = \frac{1}{2} \sum_{l=1}^{L} f_l(\boldsymbol{x})$. The LogitBoost algorithm uses Newton steps for fitting a logistic model by maximizing binomial log-likelihood of the data

$$l(y, p(\boldsymbol{x})) = \sum_{i=1}^{n} [y_i \log(p(\boldsymbol{x}_i)) + (1 - y_i) \log(1 - p(\boldsymbol{x}_i))].$$

The detail of the LogitBoost algorithm is presented in Algorithm 1. The ν in (3) is the shrinkage parameter. The natural value is 1, but a smaller value might be a better choice. This makes the algorithm slower, since more iterations are needed, but more stable, since the steps taken are smaller. Also, an implementation protection is necessary by enforcing thresholds on the weights w_i and working responses z_i. In our setting, we follow the suggestion in [11] and use the following thresholds: $w_i = \max\{w_i, 10^{-10}\}$ and $z_i = \max\{\max\{z_i, -3\}, 3\}$.

Algorithm 1 LogitBoost algorithm for two-class classification problem

Initialize: $F(\boldsymbol{x}_i) = 0$ and $p(\boldsymbol{x}_i) = 1/2$, $i = 1, \ldots, n$
for $l = 1, \ldots, L$ **do**

Compute the working responses and weights

$$\begin{aligned} z_i &= \frac{y_i - p(\boldsymbol{x}_i)}{p(\boldsymbol{x}_i)[1 - p(\boldsymbol{x}_i)]}, \\ w_i &= p(\boldsymbol{x}_i)[1 - p(\boldsymbol{x}_i)]; \end{aligned}$$

Fit a weighted least square regression $f_l(\boldsymbol{x})$ of training points $\boldsymbol{x}_i$ to response values z_i with weights w_i;
Denote the fitted values by $\hat{f}_l$;
Update the regression function by the fitted values

$$\begin{aligned} F &\leftarrow F + \nu \hat{f}_l, \quad (3) \\ p &\leftarrow \frac{1}{1 + e^{-F}}; \end{aligned}$$

end for
Output the classifier $F(\boldsymbol{x}) = \mathrm{sgn}[\sum_{l=1}^{L} \hat{f}_l(\boldsymbol{x})]$.

An essential ingredient of the LogitBoost algorithm is the weighted least squares solver. For image data, a naive way is to vectorize voxels and then apply the regular weighted least squares solver. Two potential pitfalls prevent this strategy. First, the number of voxels far exceeds the number of observations and there is no unique solution. Second, vectorizing voxels destroys the tensor structure of images which themselves possess huge amount of information. Ideally the procedure should respect the tensor structure to retain as much spatial information as possible. Our numerical results show that respecting tensor structure significantly increases the classification performance of the boosting algorithm.

The recent tensor regression model developed in [10] supplies a natural regression method for tensor-valued predictors. Consider the special case of weighted least squares criterion

$$\frac{1}{2} \sum_{i=1}^{n} w_i \left(y_i - \langle \boldsymbol{X}_i, \boldsymbol{B} \rangle\right)^2,$$

where $y_i \in \mathbb{R}$ are scalar responses, $\boldsymbol{X}_i \in \mathbb{R}^{p_1 \times \cdots \times p_D}$ are tensor-valued predictors, and $\boldsymbol{B}$ is a tensor regression parameter. Limited sample size prevents estimation of the full tensor parameter $\boldsymbol{B}$ which has the same size as $\boldsymbol{X}$.

In the rank-R tensor regression introduced in [10], we consider the criterion

$$\begin{aligned} & L(\boldsymbol{B}_1, \ldots, \boldsymbol{B}_D) \\ = & \frac{1}{2} \sum_{i=1}^{n} w_i \left(y_i - \sum_{r=1}^{R} \boldsymbol{X}_i \times_1 \boldsymbol{\beta}_1^{(r)} \cdots \times_D \boldsymbol{\beta}_D^{(r)} \right)^2 \\ = & \frac{1}{2} \sum_{i=1}^{n} w_i \left(y_i - \langle \boldsymbol{X}_i, (\boldsymbol{B}_D \odot \cdots \odot \boldsymbol{B}_1) \mathbf{1}_R \rangle \right)^2, \quad (4) \end{aligned}$$

where $\boldsymbol{B}_d = [\boldsymbol{\beta}_d^{(1)}, \ldots, \boldsymbol{\beta}_d^{(R)}] \in \mathbb{R}^{p_d \times R}$, $\odot$ denotes the Khatri-Rao product, and $\mathbf{1}_R$ is the vector of R ones.

Dimensionality of this tensor regression model is $R \sum_p p_d$, which is usually substantially smaller than that of the full model $\prod_d p_d$. In the ADHD-200 global competition, for example, dimensionality of the MRI data is reduced from $128^3 = 2,097,152$ to $128 \times 3 = 384$ by a rank $R = 1$ model. More importantly, the multi-linear form in (4) generalizes the linear form in the classical linear regression while still respecting the tensor structure in $\boldsymbol{X}$.

The parameter $(\boldsymbol{B}_1, \ldots, \boldsymbol{B}_D)$ can be efficiently estimated by an alternating least squares (ALS) algorithm, summarized in Algorithm 2, for solving the weighted tensor least squares problem (4).

Algorithm 2 Alternating least squares (ALS) algorithm for maximizing the weighted tensor least squares criterion (4)

Initialize: $\boldsymbol{B}_d^{(0)} \in \mathbb{R}^{p_d \times R}$ for $d = 1, \ldots, D$
Repeat
for $d = 1, \ldots, D$ **do**
$\boldsymbol{B}_d^{(t+1)}$ is obtained by minimizing

$$\frac{1}{2} \sum_i w_i (y_i - \langle \boldsymbol{B}_d, \boldsymbol{X}_{i(d)} (\boldsymbol{B}_D^{(t)} \odot \cdots \odot \boldsymbol{B}_{d+1}^{(t)} \odot \boldsymbol{B}_{d-1}^{(t+1)} \odot \cdots \odot \boldsymbol{B}_1^{(t+1)}) \rangle)^2$$

end for
Until: $L(\boldsymbol{\theta}^{(t)}) - L(\boldsymbol{\theta}^{(t+1)}) < \epsilon$

Note that, when updating the block $\boldsymbol{B}_d$, it is a regular weighted least squares with Rp_d parameters, which admits an explicit solution. Therefore, the ALS algorithm is extremely simple to implement. The ALS algorithm iterates monotonically decrease the objective value and, under mild regularity conditions, converge to a stationarity point of L.

For classification with tensor-valued predictors, our proposal is to replace the regular weighted least squares by the weighted tensor least squares in Algorithm 1. Let $\{\boldsymbol{X}_i, y_i\}_{i=1,\ldots,n}$ be the training set, where $\boldsymbol{X}_i \in \mathbb{R}^{p_1 \times \cdots \times p_D}$ and $y_i \in \{0, 1\}$. The tensor LogitBoost is summarized in Algorithm 3.

In practice we often need to scale down image data such that the effective model size, $R(p_l + p_r) - R^2$ for $D = 2$ or $R(\sum_d p_d - D + 1)$ for $D > 2$, of tensor least squares is less than the sample size. Success of tensor LogitBoost lies in its ability to retain the tensor structure during the classification stage. Therefore in the preprocessing step we would like to keep the tensor structure too. Particularly we use tensor PCA to reduce sizes of image data.

Algorithm 3 Tensor LogitBoost

Initialize: $F(\boldsymbol{X}_i) = 0$ and $p(\boldsymbol{X}_i) = 1/2$, $i = 1, \ldots, n$
for $l = 1, \ldots, L$ **do**
Compute responses and weights

$$\begin{aligned} z_i &= \frac{y_i - p(\boldsymbol{X}_i)}{p(\boldsymbol{X}_i)[1 - p(\boldsymbol{X}_i)]}, \\ w_i &= p(\boldsymbol{X}_i)[1 - p(\boldsymbol{X}_i)]; \end{aligned}$$

Fit a weighted tensor linear regression with responses z_i, predictors $\boldsymbol{X}_i$ with weights w_i;
Denote the fitted values by $\hat{f}_l$;
Update the regression function by the fitted values

$$\begin{aligned} F &\leftarrow F + \hat{f}_l, \\ p &\leftarrow \frac{1}{1 + e^{-F}}; \end{aligned}$$

end for
Output the classifier $F(\boldsymbol{X}) = \text{sgn}[\sum_{l=1}^{L} \hat{f}_l(\boldsymbol{X})]$.

Define the first two sample moments of d-matricization of data as

$$\begin{aligned} \bar{\boldsymbol{X}}_{(d)} &= \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{X}_{i(d)}, \\ \boldsymbol{S}_{(d)} &= \frac{1}{n} \sum_{i=1}^{n} (\boldsymbol{X}_{i(d)} - \bar{\boldsymbol{X}}_{(d)})(\boldsymbol{X}_{i(d)} - \bar{\boldsymbol{X}}_{(d)}). \end{aligned}$$

Suppose the symmetric matrix $\boldsymbol{S}_{(d)} \in \mathbb{R}^{p_d \times p_d}$ admits eigen-decomposition $\boldsymbol{S}_{(d)} = \boldsymbol{U}_d \Lambda_d \boldsymbol{U}_d$ for $d = 1, \ldots, D$ and the target dimension is $q_1 \times \cdots \times q_D$, where $q_d \le p_d$ for all d. Then the reduced image for each subject is

$$\tilde{\boldsymbol{X}}_i = \boldsymbol{X}_i \times_1 \tilde{\boldsymbol{U}}_1 \cdots \times_D \tilde{\boldsymbol{U}}_D \in \mathbb{R}^{q_1 \times \cdots q_D},$$

where $\tilde{\boldsymbol{U}}_d \in \mathbb{R}^{p_d \times q_d}$ contains the top q_d eigenvectors of $\boldsymbol{S}_{(d)}$ in its columns. Then the follow-up tensor LogitBoost is performed on reduced tensor data $\tilde{\boldsymbol{X}}_i$, $i = 1, \ldots, n$. Note that, for $D = 1$, tensor PCA reduces to the classical PCA.

Also in practice the tensor LogitBoost algorithm has to be tuned on a training data set for better performance on the testing data. The parameters subject to tuning include the number of boosting steps L, the rank of tensor regression R, and the shrinkage parameter ν in boosting algorithm [11]. In following examples, we apply cross validation on the training data to choose the best combination among the following range:

- Boosting step L between 1 and 1000.

- Rank $R \in \{1, 2, 3\}$
- Shrinkage $\nu \in \{0.01, 0.05, 0.1\}$.

IV. EEG Data Analysis

We now apply the proposed tensor LogitBoost to analyze an EEG images dataset. Our proposed method is compared to the regular LogitBoost with the vectorized predictors (regular LogitBoost). The study examines EEG correlates of genetic predisposition to alcoholism and involved two groups of subjects: an alcoholic group of 77 subjects and a control group of 45 subjects. Each subject was exposed to either one stimulus or two stimuli. During an exposure, the voltage values were measured from 64 channels of electrodes and for 256 time points. The 64 electrodes are placed at different locations on the subject's scalp. The stimuli were pictures chosen from a picture set. When two pictures were shown, they were displayed in either a matched condition, where two pictures were identical, or a unmatched condition, where they were different. Each subject had 120 trials under these three conditions: single stimulus, two matched stimuli and two unmatched stimuli. The primary interest was to study the association between alcoholism and the pattern of voltage values over times and channels.

To keep matters simple, in this paper, we used only part of the data set: we included only the single stimulus condition and, for each subject, we took the average of all the trials under that condition. That is, the portion of the data we used consists of $(\boldsymbol{X}_1, Y_1), ..., (\boldsymbol{X}_{122}, Y_{122})$, where $\boldsymbol{X}_i$ is a 256×64 matrix with each entry representing the mean voltage value of subject i at a combination of a time point and a channel, averaged over all trials under the single stimulus condition, and Y_i is a binary random variable indicating whether the ith subject is alcoholic ($Y_i = 1$) or nonalcoholic ($Y_i = 0$). To visualize and illustrate the results in a meaningful way, we set $p_l = d_r = 2$.

Instead of directly applying the classification method to the original EEG data, we preprocessed the data by tensor PCA to reduce the dimension. We partition the data into $k \in \{5, 10, 122\}$ portions. Each portion is held out once as the testing data set and the remaining $k-1$ portions together form the training data set. For the training set, a further five-fold cross-validation is applied to choose the tuning parameters, including the number of boosting steps L, the rank of tensor regression R and the shrinkage parameter ν. We experimented several scales of downsizing and obtained the similar results.

Table I shows the superior performance of tensor LogitBoost compared to the regular LogitBoost under all scenarios. The best results are highlighted in boldface.

Table I: Testing error for EEG data

Method	Leave-one-out	Five-fold	Ten-fold
Tensor LogitBoost	**0.167**	**0.221**	**0.157**
Regular LogitBoost	0.288	0.287	0.290

V. MRI Data Analysis

Attention deficit hyperactivity disorder (ADHD) is one of the most commonly diagnosed behavioral disorders among children. The primary symptoms for ADHD could be classified into three groups: developmentally inappropriate inattention, impulsive behavior and hyperactivity. Children with these symptoms do not know how to control their behaviors or have trouble organizing their thoughts. About 5% of U.S. children aged 6-17 have been affected with ADHD. The levels of these primary symptoms are widely used as the diagnosis and treatment evaluation of ADHD. Ranking of these primary symptoms is often evaluated by the teachers or parents of the children, which is inherently subjective. Therefore, more objective methods are greatly desired.

The goal of the ADHD-200 global competition is to develop novel strategies for predicting ADHD diagnostic status based on an individual's MRI data or fMRI data. The ADHD-200 initiative organized the public release of the MRI data and fMRI data for 776 children (285 children with ADHD and 491 controls) across 8 independent sites. A testing set of 195 unlabeled children then will be utilized to measure the performance of the classifiers developed by the teams. Competition results have revealed that the prediction accuracy varied from 43.08% to 61.54% with mean $= 56.02\%$. The mean accuracy of predicting control subjects is 71.77%, which is larger than the one of predicting children with ADHD (37.44%).

We only included the MRI data in our analysis. The MRI data set that we used is the preprocessed version of the ADHD-200 Global Competition MRI data set, which is released by the Neuro Bureau. SPM8 is used for the preprocessing of the MRI data.

The preprocessed MRI data comprises of 776 labeled children and 195 unlabelled children. The labeled training set contains 285 children with ADHD and 491 typical developed children (TDC). Not all the children in the training set and testing set were used for our analysis. For some children, the MRI data are fragmentary and of low quality. Moreover, the diagnostic labels for the 26 participants from the Brown University are unavailable. It turns out that 770 training children and 171 testing children were used in our study. The number of children from each site is listed in Table II.

Prior to classification, we performed the tensor PCA to downsize the MRI data. The original size of the MRI data is $121 \times 145 \times 121$. We experimented four different scales of PCA. Then we applied the tensor LogitBoost and regular LogitBoost to classify ADHD status based on the subject's downsized MRI data. For the 770 training data, we employed a five-fold cross-validation to tune the number

Table II: Training and testing subjects from different sites

Site Name	Training	Testing
Peking University	194	51
Brown University	0	26
Kennedy Krieger Institute	83	11
NeuroImage	48	25
New York University	217	41
Oregon Health & Science University	79	34
University of Pittsburgh	89	9
Washington University	60	0
Total	770	171

Table III: Testing error for MRI Data

Reduced dimension	Regular LogitBoost	Tensor LogitBoost
$25 \times 29 \times 25$	0.37	0.33
$31 \times 37 \times 31$	0.39	0.32
$37 \times 44 \times 37$	0.39	0.33
$43 \times 51 \times 43$	0.45	0.31

of boosting steps L, the rank of tensor regression R, and the shrinkage parameter ν in the tensor LogitBoost (upper-left panel). Also, L and R in the regular LogitBoost were tuned using the same strategy (upper-right panel). We then applied the tuned model to the testing data and evaluated the misclassification error rate (lower-right panel).

The classification errors on the testing set are reported in Table III. Again, we observe superior performance of tensor LogitBoost compared to the regular LogitBoost. The best prediction accuracy of 69.00% can be achieved when the $43 \times 51 \times 43$ downsized MRI is used as the input of the tensor LogitBoost. Recall that the ADHD-200 global competition results show that the prediction accuracy varied from 43.08% to 61.54% with mean 56.02%. Hence, the results demonstrate convincingly the advantage of our new classification procedure.

VI. Conclusion

Modern brain image data have posed various challenges to the traditional classification methods due to their high dimension and complex structure. To address these challenges, we propose the tensor LogitBoost algorithm as an "off-the-shelf" classification tool based on tensor-valued brain image data.

Compared to the current existing classification tools in brain image literature, tensor LogitBoost is straightforward to implement and applies to a variety of brain image data, such as MRI, DTI, fMRI, PET, EEG and MEG. By maintaining the tensor structure along the pipeline, it is able to retain the rich structural information in the image data and improves the performance of the regular boosting machines. Our applications to the EEG data and MRI data demonstrate its competence.

We concentrated on a two-class classification problem. However, extensions to multi-class are worth further research. The multi-class LogitBoost algorithm would be a good starting point. The multi-class tensor LogitBoost algorithm would be similar to the original multi-class LogitBoost, except for a few differences at the level of weak learners. The regression functions should be learned using weighted tensor least squares.

Algorithm 4 LogitBoost algorithm for multi-class classificaiton problem

Initialize: $w_{ij} = 1/n$, $i = 1, \ldots, n$, $j = 1, \ldots, J$, $F_j(\boldsymbol{x}_i) = 0$ and $p_j(\boldsymbol{x}_i) = 1/J$, $\forall j$

Repeat for $m = 1, \ldots, M$:

1) Repeat for $j = 1, \ldots, J$:

a) Compute the working responses and weights

$$\begin{aligned} z_{ij} &= \frac{y_{ij} - p_j(\boldsymbol{x}_i)}{p_j(\boldsymbol{x}_i)[1 - p_j(\boldsymbol{x}_i)]}, \\ w_{ij} &= p_j(\boldsymbol{x}_i)[1 - p_j(\boldsymbol{x}_i)]; \end{aligned}$$

b) Fit a weighted least square regression, $f_{mj}(\boldsymbol{x})$ of training points $\boldsymbol{x}_i$ to response values z_{ij} with weights w_{ij}, and denote the fitted values by $\hat{f}_l$;

2)

$$\begin{aligned} f_{mj} &\leftarrow \frac{J-1}{J}[f_{mj}(\boldsymbol{x}) - \frac{1}{J}\sum_{k=1}^{J} f_{mk}(\boldsymbol{x})], \\ F_j(\boldsymbol{x}) &\leftarrow F_j(\boldsymbol{x}) + f_{mj}(\boldsymbol{x}); \end{aligned}$$

3) Update p_j by

$$p_j = \frac{e^{F_j(\boldsymbol{x})}}{\sum_{k=1}^{J} e^{F_k(\boldsymbol{x})}};$$

Output the classifier $\text{argmax}_j F_j(\boldsymbol{x})$.

Acknowledgment

The authors would like to thank Professor Lexin Li from University of California, Berkeley for his constructive and inspiring comments and suggestions on our research on Tensor LogitBoosting with images as predictors.

References

[1] P. J. Pardo, A. P. Georgopoulos, J. T. Kenny, T. A. Stuve, R. L. Findling, and S. C. Schulz, "Classification of adolescent psychotic disorders using linear discriminant analysis," *Schizophrenia Research*, vol. 87, pp. 297–306, 2006.

[2] J. Ford, H. Farid, F. Makedon, L. A. Flashman, T. W. McAllister, V. Megalooikonomou, and A. J. Saykin, "Patient classification of fmri activation maps," in *in Proc. of the 6th Annual International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI'03*, 2003, pp. 58–65.

[3] C. Sung, J. Woo, M. Goodman, T. Huffman, and Y. Choe, "Scalable, incremental learning with mapreduce parallelization for cell detection in high-resolution 3d microscopy data," in *Neural Networks (IJCNN), The 2013 International Joint Conference on*. IEEE, 2013, pp. 1–7.

[4] V. D. Calhoun, P. K. Maciejewski, G. D. Pearlson, G. D. Pearlson, and K. A. Kiehl, "Temporal lobe and default hemodynamic brain modes discriminate between schizophrenia and bipolar disorder," *Human Brain Mapping*, vol. 29, pp. 1265–1275, 2008.

[5] O. Demirci, V. Clark, and V. D. Calhoun, "A projection pursuit algorithm to classify individuals using fmri data: Application to schizophrenia," *NeuroImage*, vol. 39, pp. 1774–1782, 2008.

[6] B. Li, M. K. Kim, and N. Altman, "On dimension folding of matrix- or array-valued statistical objects," *Annals of Statistics*, vol. 38, pp. 1094–1121, 2010.

[7] B. Li, A. Artemiou, and L. Li, "Principal support vector machines for linear and nonlinear sufficient dimension reduction," *Annals of Statistics*, vol. 39, no. 6, pp. 3182–3210, 2011.

[8] B. Zhang and L. Wang, "Structure preserving dimension reduction with 2d images as predictors," in *Big Data (Big Data), 2016 IEEE International Conference on*. IEEE, 2016, pp. 3619–3624.

[9] L. Breiman *et al.*, "Arcing classifier (with discussion and a rejoinder by the author)," *The annals of statistics*, vol. 26, no. 3, pp. 801–849, 1998.

[10] H. Zhou, L. Li, and H. Zhu, "Tensor regression with applications in neuroimaging data analysis," *Journal of the American Statistical Association*, vol. 108, no. 502, pp. 540–552, 2013.

[11] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, 2001.

[12] G. Ridgeway, "The state of boosting," *Computing Science and Statistics*, vol. 31, pp. 172–181, 1999.

[13] Y. Freund and R. E. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," in *European conference on computational learning theory*. Springer, 1995, pp. 23–37.