Voxel Selection in fMRI Data Analysis Based on Sparse Representation

Yuanqing Li*, *Member, IEEE*, Praneeth Namburi, Zhuliang Yu, *Member, IEEE*, Cuntai Guan, *Senior Member, IEEE*, Jianfeng Feng, and Zhenghui Gu, *Member, IEEE*

***Abstract*—Multivariate pattern analysis approaches toward detection of brain regions from fMRI data have been gaining attention recently. In this study, we introduce an iterative sparse-representation-based algorithm for detection of voxels in functional MRI (fMRI) data with task relevant information. In each iteration of the algorithm, a linear programming problem is solved and a sparse weight vector is subsequently obtained. The final weight vector is the mean of those obtained in all iterations. The characteristics of our algorithm are as follows: 1) the weight vector (output) is sparse; 2) the magnitude of each entry of the weight vector represents the significance of its corresponding variable or feature in a classification or regression problem; and 3) due to the convergence of this algorithm, a stable weight vector is obtained. To demonstrate the validity of our algorithm and illustrate its application, we apply the algorithm to the Pittsburgh Brain Activity Interpretation Competition 2007 functional fMRI dataset for selecting the voxels, which are the most relevant to the *tasks* of the subjects. Based on this dataset, the aforementioned characteristics of our algorithm are analyzed, and a comparison between our method with the univariate general-linear-model-based statistical parametric mapping is performed. Using our method, a combination of voxels are selected based on the principle of effective/sparse representation of a task. Data analysis results in this paper show that this combination of voxels is suitable for decoding tasks and demonstrate the effectiveness of our method.**

***Index Terms*—Functional MRI (fMRI), prediction, sparse representation, statistical parametric mapping (SPM), voxel selection.**

I. Introduction

IN FUNCTIONAL MRI (fMRI), an fMRI scanner measures the blood-oxygenation-level-dependent (BOLD) signal at all points in a 3-D grid, or image of the brain. Each grid of the 3-D image is known as a voxel. A typical fMRI dataset is composed of time series (BOLD signals) of tens of thousand voxels. High volume is a characteristic of fMRI data. Therefore, voxel selection plays an important role in fMRI data analysis because of the following: 1) heavy computation burden and 2) uncorrelation (or redundancy) of a large number of voxel time series with respect to the stimulus/task presented to the subject. Much of current fMRI research such as identifying brain regions activated in response to a task or stimulus is related to the issue of voxel selection.

General linear model (GLM) is a classical univariate approach toward detection of task-related activation in the brain. A typical example is the statistical parametric mapping (SPM) based on the GLM. SPM is a powerful tool for the analysis of fMRI data including voxel selection [2]–[4]. Correlation-based methods, which are univariate approaches, are also useful for detection of task-related activations in the brain [5].

Recent multivariate approaches draw from pattern classification and machine learning theory, including classifier-based method [6], multiple regressor model [7], as well as least square regression with L (ridge) and L (Lasso) regularization [8]. Recently, an elastic net regression technique was proposed in [9]. This technique achieves both sparsity and clustering effect by using a weighted combination of 1-norm and 2-norm penalties on top of the least-squares problem, was applied to the analysis of the fMRI dataset of Pittsburgh Brain Activity Interpretation Competition (PBAIC) 2007. Considering sparsity and clustering effect simultaneously, the authors demonstrated the distributed nature of neural activities and the importance of localized clusters of activity.

In this paper, we present a novel sparse-representation-based method for voxel selection in fMRI data.

The sparse representation of signals can be modeled by

$$\mathbf{y} \quad \mathbf{Aw} \tag{1}$$

where $\mathbf{y} \in R^N$ is a given signal vector and $\mathbf{A} \in R^{N \times M}$ ($N < M$) is a basis matrix. In the context of fMRI data analysis, in model (1), $\mathbf{A}$ represents a data matrix of which each column is a time series of a voxel, and $\mathbf{y}$ is the stimulus/task function convolved with a hemodynamic response function to translate expected task related neural activity to expected BOLD response.

The task of sparse representation is to find a solution $\mathbf{w} \in R^M$ for (1) such that this solution is as sparse as possible. In many references, such as [11], a basis pursuit (BP) algorithm was presented, in which a sparse solution (i.e., -norm solution) can be found by solving the following optimization problem:

$$\|\mathbf{w}\| , \qquad \text{s.t. } \mathbf{Aw} \quad \mathbf{y} \tag{2}$$

where 1-norm $\|\mathbf{w}\|$ is defined as $\sum_i^M |w_i|$.

Manuscript received August 19, 2008; revised April 7, 2009. First published June 26, 2009; current version published September 16, 2009. The work of Y. Li and Z. Yu was supported by the National Natural Science Foundation of China under Grant 60825306 and Grant 60802068. *Asterisk indicates corresponding author.*

*Y. Li is with the School of Automation Science and Engineering, South China University of Technology, Guangzhou 510640, China (e-mail: auyqli@scut.edu.cn).

P. Namburi is with the Cognitive Neuroscience Laboratory, DUKE-NUS Graduate Medical School, Singapore 169857, Singapore, and also with the Institute for Infocomm Research, Singapore 119613, Singapore.

Z. Yu and Z. Gu are with the School of Automation Science and Engineering, South China University of Technology, Guangzhou 510640, China.

C. Guan is with the Institute for Infocomm Research, Singapore 119613, Singapore.

J. Feng is with the Department of Computer Science and Mathematics, Warwick University, Coventry CV4 7AL, UK, and also with Centre for Computational Systems Biology, Fudan University, Shanghai 200433, China.

Digital Object Identifier 10.1109/TBME.2009.2025866

Setting $\mathbf{w}$ $\mathbf{u}-\mathbf{v}$, where $\mathbf{u}, \mathbf{v} \in R^M$ are nonnegative, (2) can be converted to the following equivalent linear programming problem:

$$\sum_{i}^{M} u_i \quad v_i \ , \qquad . \ . \ \mathbf{A}, -\mathbf{A} \ \ \mathbf{u}^T, \mathbf{v}^{T\ T} \qquad \mathbf{y}, \mathbf{u}, \mathbf{v} \geq \ . \tag{3}$$

The solution of a linear programming problem is generally unique [19], which can be obtained by standard software packages. In this paper, all linear programming problems are solved using the MATLAB function "linprog."

Sparse representation of signals has received a great deal of attention in recent years (e.g., see [11]–[16]). For instance, Donoho and Elad discussed optimal sparse representation in general (nonorthogonal) dictionaries via l minimization [17]. In practical applications, sparse representation can be used in underdetermined blind source separation (BSS), which is difficult to deal with using a standard independent component analysis (ICA) method [18]–[23]. BP is also an important application of sparse representation [11], [17]. Recently, it has been found that model (2) has applications in feature selection and detection tasks. Equation (2) was successfully used in [22] for cross-modal localizations of sound-related region in the video, where $\mathbf{A}$ was constructed from the video and $\mathbf{y}$ from the accompanying audio.

A related method is -norm support vector machine (SVM). Similar to the BP algorithm, 1-norm SVM solves a linear programming problem to obtain a sparse solution. Thus, it is also called sparse SVM [24]–[29]. 1-norm SVM has potential applications in feature selection including dimension reduction [24], detection of region of interest of images [25], detection of machine damage, or highlighting abnormal features (localization) in medical data [29]. There are differences between the models (2) and -norm SVM. For instance, when 1-norm SVM and model (2) are used for the same dataset, there are more variables and constraints for 1-norm SVM than for (2). This implies a heavier computational burden is required for 1-norm SVM. Another related method is the so-called Lasso regularization, which is also used in the potential SVM [30], [31]. Compared with Lasso method or the potential SVM with quadratic objective functions, (2) can be converted into a standard linear programming problem and has computational advantage especially when the number of the variables is extremely large as in fMRI data.

In the following, we compare the model (1) with the GLM model, and analyze the difference between the two models. GLM model is represented by

$$\mathbf{x}_i \quad \mathbf{G}\beta_i \quad \mathbf{e}_i \tag{4}$$

where $\mathbf{x}_i \in R^N$ is a time series of the ith voxel, $\mathbf{G} \in R^{N\times K}$ is called a design matrix, $\beta_i \in R^K$ is an unknown parameter vector to be estimated for each voxel, $\mathbf{e}_i \in R^N$ is an error (noise) vector, i $, \ldots, M$. Each column of $\mathbf{G}$ corresponds to an explanatory variable related to the specific experimental conditions under which the data were collected, β_i represents the weights of the explanatory variables (columns) of $\mathbf{G}$.

Considering all voxels, the matrix form of (4) becomes

$$\mathbf{X} \quad \mathbf{G}\beta \quad \mathbf{E} \tag{5}$$

where $\mathbf{X} \in R^{N\times M}$ is the data matrix, which is the same as $\mathbf{A}$ in (1), $\beta \in R^{K\times M}$, $\mathbf{E} \in R^{N\times M}$.

Multiplying both sides of (5) by the Moor–Penrose inverse β of β, we have

$$\mathbf{G} \quad \mathbf{X}\beta \quad -\mathbf{E}\beta \ . \tag{6}$$

Furthermore, considering each column $\mathbf{g}_j$ of $\mathbf{G}$ and letting the noise vector be included implicitly in the coefficient vector, (6) can be rewritten as

$$\mathbf{g}_j \quad \mathbf{X}\ \beta_j \quad \mathbf{e}_j \tag{7}$$

where $\mathbf{e}_j$ $-\mathbf{X}\ \ \mathbf{E}\beta_j$.

Since $\mathbf{g}_j$, representing a specific experimental condition, is the convolution of a stimulus/task function and a hemodynamic response function (HRF), it is $\mathbf{y}$ in (1). Furthermore, in view of $\mathbf{X}$ in (7) and $\mathbf{A}$ in (1) representing the same data matrix, (7) is equivalent to the model in (1). The previous analysis shows how the model in (1) is related to the GLM model in (4). The main differences between these two models are as follows: 1) in model (1), a transformed stimulus/task function is linearly represented by the time series of a set of voxels. The assumption of sparse representation implies that the number of voxels used in this representation is as small as possible. Note that although only a small number of voxels are needed in sparse representation, they are generally representative voxels distributed in different activated brain areas. In contrast, in (7), the time series of each voxel is linearly represented by the columns of a design matrix, of which each column is a transformed stimulus/task function, or a function related to noise, etc., and 2) model (1) is a multivariate approach because the relationship of different voxels is extracted using (1). Conversely, all the stimulus/task functions are considered simultaneously in the GLM model (4). It is the relationship of different stimulus/task functions that is extracted in (4) rather than the connection of different voxels. Thus, model (4) is a univariate approach. The relationship between different voxels is generally taken into account (e.g., by random field method) in later analysis of SPM.

In this paper, we develop a sparse representation algorithm based on the linear programming problem (2) for voxel selection in fMRI data analysis. The aim of sparse representation is to find a coefficient vector $\mathbf{w}$ of model (1) such that $\mathbf{w}$ is as sparse as possible. The motivations for using sparse representation here are:

1) Considering a huge number of voxels of the brain, only a small number of voxels are useful for representing a stimulus/task function $\mathbf{y}$ in (1). This is reflected by the sparsity of $\mathbf{w}$.
2) Through sparse representation, we obtain a combination of voxels. This combination of voxels can represent the stimulus/task function $\mathbf{y}$ with a high efficiency since it contains a small number of voxels ($\mathbf{w}$ is sparse). Thus, the links between those voxels in the combination are emphasized through an effective/sparse representation of $\mathbf{y}$.

3) The voxels picked by sparse representation can be categorized into two sets: one where the times series are correlated with $\mathbf{y}$, and the other set of voxels whose time series are not significantly correlated to $\mathbf{y}$, but still contain important information necessary to represent $\mathbf{y}$. The first set of voxels can be identified using routine statistic parametric methods, e.g., GLM–SPM. However, the second set of voxels are difficult to be identified using these statistic parametric methods (see Fig. 7).
4) In the model (1), the combination of voxels are selected based on the principle of effective/sparse representation of a task. As shown in our data analysis of this paper, the voxel selection based on model (1) could be more suitable for decoding tasks than GLM model in which the representation of a task is not considered.

To elaborate the advantages of our method as summarized earlier, we apply it to the fMRI dataset of PBAIC 2007. These data were collected for a prediction task. Stimuli in the experiments performed to obtain these datasets are rich and nonrepetitive. Therefore, it is difficult to perform voxel selection satisfactorily using typical methods such as Pearson-correlation-based methods. After voxel selection with our method, we perform the prediction of experience-based cognitive tasks from the fMRI dataset of PBAIC 2007 as in [8] and [10]. The prediction results will be used in evaluation of our method. In our data analysis, we also compare our method with the benchmark approach: the GLM–SPM method.

The remaining part of this paper is organized as follows. Our detection algorithm is presented in Section II. The analysis of convergence and effectiveness of this algorithm are also included. In Section III, we use our algorithm for voxel selection in fMRI data analysis. Additional discussions related to fMRI data analysis are included in Appendixes I and II. Finally, conclusions are presented in Section IV to review our method.

II. Materials and Methods

A. Algorithm

In this section, we describe our algorithm for voxel selection. The algorithm includes a few more steps than just solving (2) because of the following three aspects: 1) the number of nonzero entries of $\mathbf{w}$ generally equals to N [19]. This means that the sparsity of $\mathbf{w}$ decreases with the increasing of N. This leads to a situation where increase in the amount of training data may not lead to any improvement in the feature selection by (2); 2) equation (2) is not suitable for the overdetermined case in which $N > M$; 3) when N is not sufficiently large, $\mathbf{w}$ obtained by a single optimization iteration (see shortly) may not reflect the important data features well. Even if N is sufficiently large, this problem still exists because of the presence of noise. In order to address these three problems, we extend (2) and present an iterative detection algorithm in this paper.

Suppose that each row of $\mathbf{A}$ in (2) represents a data sample, $\mathbf{y}$ can be speech signal, stimulus, labels etc.

In this paper, $\mathbf{A}$ is an fMRI data matrix of which each column is the time series of a voxel (and hence each row represented many voxels at one point in time), $\mathbf{y}$ is the stimulus/task function convolved with a hemodynamical response function. The following algorithm is designed to detect the parts in the rows of $\mathbf{A}$ (e.g., pixels or voxels) relevant to $\mathbf{y}$.

Algorithm 1:

Step 1: For k , . . ., do the following Steps 1.1 to 1.4.

Step 1.1: Randomly choose L rows from $\{\mathbf{a} , \ldots , \mathbf{a}_N\}$ to construct a $L \times M$ matrix denoted as $\mathbf{A}_k$, the corresponding L entries of $\mathbf{y}$ form a column vector denoted as $\mathbf{y}_k \in R^L$.

Step 1.2: Solve the following optimization problem. Similar to (2), this optimization problem can be converted to a standard linear programming problem

$$\|\mathbf{w}\| , \qquad \text{s.t. } \mathbf{A}_k\mathbf{w} \quad \mathbf{y}_k. \tag{8}$$

The optimal solution of (8) is denoted by $\mathbf{w}^{\,k}$.

Step 1.3: Let

$$\mathbf{w}^{\,k} \quad \overline{k}\sum_{i}^{k}\mathbf{w}^{\,i} . \tag{9}$$

Step 1.4: If $d\ k$ $\|\mathbf{w}^{\,k} - \mathbf{w}^{\,k-}\| < \alpha$ or k K , where α is a predefined small positive constant and K is a predefined limiting upper bound for the number of iterations (e.g., K), set $\mathbf{w}$ $\mathbf{w}^{\,k}$ and go to Step 2. Otherwise go to Step 1.1.

Step 2: For a given positive θ, define R $\{j| |w_j| > \theta, j$ $, \cdots , M\}$. Then, R is our detected part of interest in all rows of $\mathbf{A}$.

Remark 1: 1) Note that each $\mathbf{w}^{\,i}$ in (9) is obtained by solving a linear programming problem in the ith iteration. Suppose that there exists error vector denoted as $\mathbf{e}_i$ associated with $\mathbf{w}^{\,i}$, then the error vector with respect to the output of k iterations is the mean of $\mathbf{e}_i$, i.e. $/k \sum_i^k \mathbf{e}_i$. Thus, from the viewpoint of statistics, the error does not increase as increasing number of iterations and 2) although noise is not explicitly expressed in (8), the weight vector $\mathbf{w}$ is generally affected by noise [see (7)]. Through the average operation in (9), the effect of noise can be reduced (see Appendix III: Robustness analysis). Furthermore, through randomly selecting L rows from $\mathbf{A}$, $\mathbf{w}^{\,k}$ obtained in (8) can be seen as a random sample of $\mathbf{w}$, while $\mathbf{w}^{\,k}$ in (9) can be thought of as an approximation to the mean of $\mathbf{w}$. Suppose that Algorithm 1 terminates after K iterations. This implies that we obtain K random samples of $\mathbf{w}$ and then their mean is obtained. The number of iterations (i.e., the number of random samples) can be easily determined because of the convergence of Algorithm 1, as shown later.

In the following, we discuss the setting of three parameters L, α, and θ in Algorithm 1. Note that each $w^{\,k}$ is obtained through solving a standard linear programming problem, thus Algorithm 1 is not involved in setting the initial vector $\mathbf{w}$.

To select the parameter L, following two aspects should be taken into account: 1) L should not be very small since the columns of $\mathbf{A}_k$ and $\mathbf{y}_k$ in (8) contain temporal evolution

information and should not be lost and 2) L should not be very large so that the computational burden for solving the optimization problem (8) is not heavy. It will be explained in Appendix I that the data analysis results are not sensitive to the value of L provided that L is not very small. In this paper, we generally choose L . N. Other choices, e.g., L . N, . N, and . N, are also acceptable.

As will be proved later, Algorithm 1 is convergent, we can easily choose a small α (e.g., $\alpha <$.) to obtain a stable $\mathbf{w}$.

The threshold parameter θ can be chosen in various ways depending on the applications. One way is the cross-validation method, which is elaborated in Appendix II. Here, we present a probability method. Considering the entries of $\mathbf{w}$ are sparse, we assume that the probability distribution of the entries of $\mathbf{w}$ is Laplacian. Using all entries of $\mathbf{w}$ as samples, we estimate the mean, the variance, and the inverse cumulative distribution function F^{-} of this Laplacian distribution. We then define R $\{i|\ |w_i| > \theta, i$ $, \cdots, M\}$, where θ is chosen as F^{-} p , p is a given probability (e.g., . in this paper). As will be shown in Section III, this method for determining θ is acceptable. The values of the parameter θ determined by the cross-validation method and the probability method are generally different. We suggest the use of cross-validation method in a case where there is a lot of training data available and decoding is the main purpose and we suggest the use of probability based method if the aim is localization, i.e., detection of localized regions in the brain that contain task-relevant information.

We now analyze the convergence and effectiveness of Algorithm 1.

Convergence: Suppose that in kth iteration of Algorithm 1, we have the output

$$\mathbf{w}^{\,k} \quad \frac{}{k}\sum_{i}^{k}\mathbf{w}^{\,i}\,. \tag{10}$$

Let

$$d\ k \qquad \|\mathbf{w}^{\,k} \quad -\mathbf{w}^{\,k}\|\,. \tag{11}$$

From (10)

$$\begin{aligned}\mathbf{w}^{\,k} \quad -\mathbf{w}^{\,k} \quad & \frac{}{k}\sum_{i}^{k}\mathbf{w}^{\,i} \;-\; \frac{}{k}\sum_{i}^{k}\mathbf{w}^{\,i} \\ & \frac{k\sum_{i}^{k}\mathbf{w}^{\,i} \;-\; k \qquad \sum_{i}^{k}\mathbf{w}^{\,i}}{k\ k} \\ & \frac{k\mathbf{w}^{\,k} \quad -\sum_{i}^{k}\mathbf{w}^{\,i}}{k\ k}.\end{aligned} \tag{12}$$

In Algorithm 1, for given data matrix $\mathbf{A}$ and parameter L, there are in total C_N^L $N\ N- \cdots N-L$ $/L$ choices of the pairs $(\mathbf{A}_k,\ \mathbf{y}_k)$ in (8). For each pair $(\mathbf{A}_k,\ \mathbf{y}_k)$, there is a weight vector $\mathbf{w}^{\,k}$, which is the solution of (8). Let γ $\{\|\mathbf{w}^{\,k}\|\ , k$ $, \ldots, C_N^L\}$, then $\|\mathbf{w}^{\,k}\| \ < \gamma$.

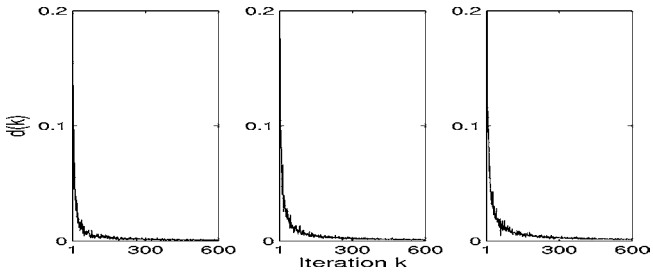

Fig. 1. Three iterative curves demonstrating the convergence of Algorithm 1 with parameter L set to be , , and , respectively, where $d\ k$ is the convergence index of Algorithm 1. The execution time for the three cases are , , and s, respectively.

From (12), we have

$$\|\mathbf{w}^{\,k} \quad -\mathbf{w}^{\,k}\| \ \le \frac{k\ \gamma}{k\ k} \qquad \frac{\gamma}{k}. \tag{13}$$

Therefore, $_{k\to\infty}\, d\ k$, i.e., Algorithm 1 is convergent.

In fact, the convergence of Algorithm 1 originates from the fact that there are finite number of weight vectors $\mathbf{w}^{\,k}$, which are bounded [each $\mathbf{w}^{\,k}$ corresponding to a pair $(\mathbf{A}_k,\ \mathbf{y}_k)$]. Although the number of $\mathbf{w}^{\,k}$ is huge, generally (C_N^L), Algorithm 1 converges in several hundred of iterations (see Fig. 1 in our data analysis section).

Although $\mathbf{w}^{\,k}$ in (10) is generally not so sparse as $\mathbf{w}^{\,i}$, our simulations and data analysis results show that a large fraction of entries of $\mathbf{w}^{\,k}$ are close to zero. Thus, we say that $\mathbf{w}^{\,k}$ is still sparse.

Effectiveness: Regarding the effectiveness of Algorithm 1, we have the following explanation. Here, we only consider the underdetermined case where $N < M$. First, we define a set of M-dimensional vectors U such that $\forall \mathbf{w}\ \dot{\mathbf{A}}, \dot{\mathbf{y}}\ \in U$, $\mathbf{w}\ \dot{\mathbf{A}}, \dot{\mathbf{y}}$ is the 1-norm solution of the equations $\dot{\mathbf{A}}\mathbf{w} \quad \dot{\mathbf{y}}$, where $\dot{\mathbf{A}}$ is composed by L rows randomly taken from $\mathbf{A}$, $\dot{\mathbf{y}}$ is a vector composed by L corresponding entries of $\mathbf{y}$. Note that there are C_N^L vectors in U. Next, we define a M-dimensional random vector $\mathbf{v}$ $v\ , \ldots, v_M\ ^{T}$, where $\mathbf{v}$ randomly takes values in U.

For a sample $\mathbf{w}\ \dot{\mathbf{A}}, \dot{\mathbf{y}}$ of $\mathbf{v}$ in U, it is sparse since it has at most L nonzeros. Generally, the magnitude of the ith entry of $\mathbf{w}\ \dot{\mathbf{A}}, \dot{\mathbf{y}}$ reflects the significance of the ith column of $\dot{\mathbf{A}}$ for the constraint $\dot{\mathbf{A}}\mathbf{w} \quad \dot{\mathbf{y}}$. Obviously, the output of Algorithm 1 satisfies $\mathbf{w} \qquad /k\ \sum_i^k \ \mathbf{w}^{\,i} \approx E\ \mathbf{v}$, i.e., the expected value of $\mathbf{v}$. Now, we show that $E\ v_i$ can reflect the significance of the ith column of $\mathbf{A}$ for the regression between $\mathbf{A}$ and $\mathbf{y}$.

For $\mathbf{w}^{\,i}$ obtained in the ith iteration Algorithm 1, we have

$$\mathbf{A}\mathbf{w}^{\,i} \quad \mathbf{y} \quad \mathbf{n}^{\,i} \tag{14}$$

where $\mathbf{n}^{\,i}$ $n^{\,i}\ , \cdots, n_N^{\,i}\ ^{T}, n_j^{\,i}$ if $j \in \mathrm{Ind}_i$ (Ind_i is the set of indexes of the L rows of $\mathbf{A}_i$ in $\mathbf{A}$), otherwise $n_j^{\,i}$ $y_j - \mathbf{a}_j \cdot \mathbf{w}^{\,i}$.

Furthermore, we have

$$\mathbf{A}\mathbf{w} \quad \mathbf{A}\left(\frac{}{k}\sum_{i}^{k}\mathbf{w}^{\ i}\right) \quad \mathbf{y} \quad \frac{}{k}\sum_{i}^{k}\mathbf{n}^{\ i}\ . \qquad (15)$$

Note that $n_j^{\ i}$ can be positive, negative, or zero. In many cases, especially when the parameter L in Algorithm 1 is not small, the expectation of $n_j^{\ i}$ can be assumed to be close to zero, i.e., $/k\ \sum_i^k \ \mathbf{n}^{\ i} \approx \mathbf{0}$ for sufficiently large k. Thus,

$$\mathbf{A}\mathbf{w} \approx \mathbf{A}E\ \mathbf{v}\ \approx \mathbf{y}. \qquad (16)$$

Therefore, the correlation between $\mathbf{A}\mathbf{w}$ and $\mathbf{y}$ is close to and $\mathbf{w}$ is close to a regression coefficient vector between training data matrix $\mathbf{A}$ and $\mathbf{y}$.

From the previous analysis and the definition of $\mathbf{w}$ in Algorithm 1, we can see the following: 1) the magnitude of $\mathbf{w}$ [i.e., $E\ v_i$] reflects the significance of the ith column of $\mathbf{A}$ to the satisfaction of (16); 2) if L in Algorithm 1 is fixed, we can obtain a consistent $\mathbf{w}$. This is due to the convergence of Algorithm 1; 3) more importantly, $\mathbf{w}$ is still sparse. This will be demonstrated in our data analysis examples. Based on the sparsity of $\mathbf{w}$, the voxels that are the most correlated to the stimulus/task function can be selected.

B. Voxel Selection in Functional MRI Data

In this section, we apply Algorithm 1 to the fMRI data of PBAIC 2007 [34] for voxel selection. The fMRI data were collected by Siemens 3T Allegra scanner with imaging parameters TR and TE being . s and ms, respectively. Three subjects' data were available in the competition. Each subject's data consist of three *runs*. Each run consists of volumes of fMRI data (704 volumes if you include fixation), of which each volume contain $\times \quad \times$ voxels. The size of a voxel is $.\ \times\ .\ \times\ .$ mm . The preprocessed data provided by the competition are used in this paper. The data preprocessing attempted to remove some standard artifacts that occur in fMRI data that may hinder data analysis. The functional and structural data were preprocessed with analysis of functional NeuroImage (AFNI) and NeuroImage software (NIS) in the following steps: slice time correction, motion correction and detrending. The feature data were preprocessed by convolving the raw feature vectors with the double gamma hemodynamic response filter (HRF) produced by the SPM (see http://www.fil.ion.ucl.ac.uk/spm/).

Upon applying a threshold on the BOLD signal intensity to mask out the nonbrain voxels, the total number of voxels in the brain was $\approx$. Thus, the fMRI data for each run is represented by a matrix of $\approx$ columns (voxels) and rows (time points). Each column of the matrix is the time series of a voxel. When the scans were obtained, the subject was performing several tasks (e.g., listen to instructions, pick up fruits) in a virtual reality (VR) world. The *ratings* for these tasks were computed by considering the delay of hemodynamic responses and form the task functions. tasks were considered in the competition. Only the tasks for the first two runs were distributed at www.braincompetition.org. Therefore, here, we use only data from the first two runs for analysis. We present detailed results mainly for the following four tasks.

1) The *hits* task, times when subject correctly picked up fruit or weapon or took picture of a pierced person.
2) The *instructions* task, which represents the task of listening to instructions from a cell phone in the virtual world.
3) The *faces* task, times when subject looked at faces of a pierced or unpierced person.
4) The *velocity* task, times when subject was moving but not interacting with an object.

For more detailed description of the data, see [34]. The goal of the competition was to predict the task functions of the third run using the fMRI data. Our final submission based on Algorithm 1 to this competition was ranked the tenth based on the average score of the features. As pointed out in [9], a fair comparison with other methods cannot be made, as postprocessing had decisive effect on performance.

Since the number of voxels among which we are looking for task related information is large, voxel selection plays an important role in fMRI data analysis. Using the instructions task as an example, we describe our data analysis method in detail.

The preprocessed fMRI data obtained from the competition website (www.braincompetition.org) is first filtered temporally and spatially. The temporal filter is $-\ ,\ ,\ ,$ while the spatial filter is a cube with $,\ ,\quad ,\ ,\quad ,\ ,\ ,\quad ,\ ,\quad ,\ ,\quad ,\ ,\ ,\ ,\ ,\ ,\quad ,\ ,\quad ,\ ,$ (similar to a 3-D Gaussian kernel). We then perform twofold cross-validation as follows. In the first fold, we use run 1 data to calculate the Pearson correlation between the time series of each voxel and the transformed instructions task function. The voxels with high absolute value of this correlation are chosen to form a set of voxels, $\mathcal{N}$. Then, our algorithm is used for a second selection of voxels to obtain $R \subset \mathcal{N}$. In Algorithm 1, $\mathbf{A} \in R^{\ \times|\mathcal{N}|}$, of which each column is a time series of a voxel in $\mathcal{N}$, $\mathbf{y} \in R$ is a transformed task function. The parameters in Algorithm 1 are chosen as follows. The number of iterations is fixed to to evaluate the details of algorithm convergence, L is , and θ can be chosen as described in Section II-A (or using a cross-validation method presented in Appendix II). Ridge regression is used on the time series of voxels $\in R$ to predict the transformed instructions task function of Run 2. Prediction accuracy is measured as the Pearson correlation between the actual transformed task and the predicted task. In the second fold, we use Run 2 data for training and predict the transformed instructions task function for Run 1.

For the purpose of comparison, we use the GLM–SPM method to replace our method for the selection of voxels and perform the twofold cross-validation, as described earlier. Note that when the GLM–SPM method is used for voxel selection, all transformed/convolved task functions provided by PBAIC 2007 are used to construct the design matrix. For each voxel and a task, a *t*-statistics is calculated as in [4]. For a task, those voxels with high absolute values of *t*-statistics are selected.

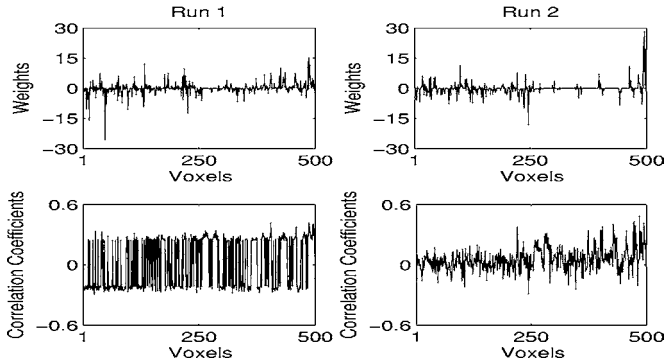

Fig. 2. (First row) Weights of voxels in $\mathcal{N}$ obtained by Algorithm 1 for subject 1 and instructions task. (Second row) Correlation coefficients for the same subject and task, each is calculated using the time series of a voxel in $\mathcal{N}$ and the transformed task function. The two columns correspond to two runs (i.e., twofolds in cross-validation), respectively.

III. Results and Discussions

In this section, we present our data analysis results to illustrate the convergence of Algorithm 1 and the sparsity of the weight vector $\mathbf{w}$. By comparing our method with the GLM–SPM method, we demonstrate the advantages of our algorithm for voxel selection.

Convergence of Algorithm 1: As an example, we show our convergence analysis result for instructions task in run 1 of subject 1. In the initial selection of voxels, we fix $|\mathcal{N}|$ to . Three cases are considered, in which the parameter L [the number of constraints in (8)] of Algorithm 1 is set to be 25, 50, and 75, respectively. The three subplots in Fig. 1 show three iterative curves of the convergence index $d\ k \quad \|\mathbf{w}^{\ k} - \mathbf{w}^{\ k-}\|$ of Algorithm 1, which correspond to the three cases, respectively. From Fig. 1, we can see that Algorithm 1 converges after $\approx$ iterations in the three cases. However, the execution time in our PC computer (2.3 GHz CPU, 3 G RAM) for the three cases are , , and s, respectively. Therefore, computational burden increases rapidly with increasing L.

Sparsity of weights: Fig. 2 plots two weight vectors $\mathbf{w}$ obtained by Algorithm 1 (top) and correlation coefficients (bottom) for the voxels in $\mathcal{N}$ obtained in the twofold cross-validation for subject 1 and instructions task. From this figure, we can see the sparsity in weights when compared to correlation coefficients. Therefore, we conclude that the weight vector obtained from our algorithm is more suitable for localization of voxels than Pearson correlation.

Effectiveness: First, we check the correlation between $\mathbf{Aw}$ and $\mathbf{y}$, where $\mathbf{A}$ and $\mathbf{y}$ are fMRI data matrix and a transformed task vector, respectively. As mentioned in Section II-A, the weight vector $\mathbf{w}$ can be seen as regression coefficients between $\mathbf{A}$ and $\mathbf{y}$. This means that the correlation between $\mathbf{Aw}$ and $\mathbf{y}$ is big. Fig. 3 shows the iterative curves of this correlation for the instructions task of three subjects for fold 1 (run 1 used for training). As the number of iterations increases, the three correlation curves increase and tend to three limits that are larger than 0.9. This confirms our analysis. When sparse representation approach is used for voxel selection, a voxel whose fMRI time series is highly correlated to $\mathbf{y}$ can generally be selected.

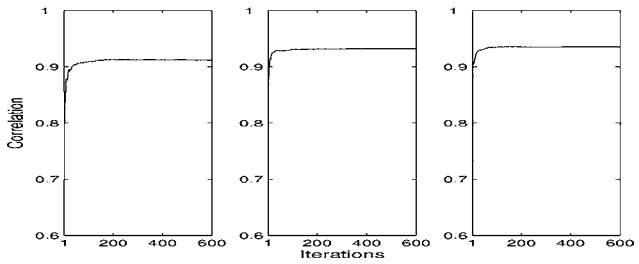

Fig. 3. Iterative curves of correlation between $\mathbf{Aw}$ and $\mathbf{y}$ (instructions task) for three subjects and run 1, each subplot corresponds to one subject.

TABLE I
Average Prediction Accuracy Rates Over Twofolds Obtained With One Voxel for Three Subjects, Four Tasks, and Two Methods (the accuracy rates obtained by GLM–SPM method are in brackets)

	Sub 1	Sub 2	Sub 3
Hits	0.267 (0.131)	0.182 (0.214)	0.26 (0.145)
Instructions	0.323 (0.223)	0.552 (0.362)	0.534 (0.142)
Faces	-0.006 (-0.319)	0.099 (0.147)	0.437 (0.103)
Velocity	0.271 (0.030)	0.198 (0.052)	0.149 (0.143)

However, if there are a set of voxels, e.g., belonging to the same brain area of which the fMRI time series are highly correlated to each other, then only a small part of representative voxels are selected. The iterative Algorithm 1 of which each iteration uses different time points may alleviate the loss of these voxels that are highly correlated to $\mathbf{y}$.

Next, we analyze the prediction accuracy. We compare Algorithm 1 with GLM–SPM method for voxel selection. First, we compare the ability of each method in choosing the *most relevant* voxel. In Table I, we present the prediction accuracies (averaged over twofolds) for N_R for three subjects, four transformed tasks. Hereafter, N_R denotes the number of voxels of R, the set of selected voxels. From Table I, we can see that the voxels selected by Algorithm 1 is more correlated to the transformed task functions in most of cases than those selected by GLM–SPM method.

Furthermore, we test if Algorithm 1 is consistently better than GLM–SPM method for voxel selection. Let $\mathbf{b}$, , ,..., . For each i (i ,...,), we set N_R b_i, and predict the four transformed task functions for all subjects and average the results over twofolds of cross-validation. Fig. 4 shows the plots of average prediction accuracy with respect to $\mathbf{b}$ for the four methods and three subjects and four tasks. In several cases, e.g., shown in the subplot in the first row and the second column of Fig. 4, the performance of Algorithm 1 is comparative to that of GLM–SPM method, while in the other cases, e.g., shown in the subplot in the second row and the third column, the performance of Algorithm 1 is significantly better than that of GLM–SPM method.

We also analyze the effectiveness of choosing R using θ as described in Section II-A. Example, for instructions task, the number of voxels in R obtained using Algorithm 1, N are: *subject 1,* 12 (fold 1), 29 (fold 2); *subject 2*, 24 (fold 1), 21 (fold 2); *subject 3*, 28 (fold 1), 20 (fold 2). The corresponding prediction accuracy (averaged over twofolds) for the three subjects is 0.8151, 0.7469, and 0.8591, respectively. The prediction

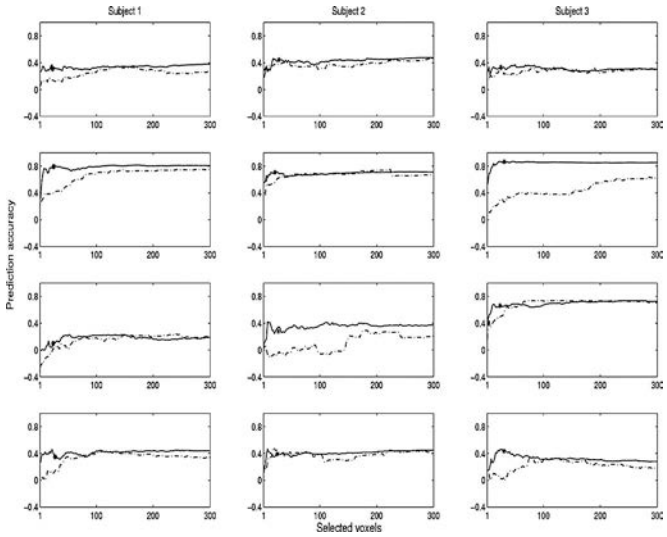

Fig. 4. Prediction accuracy curves obtained by two methods. In each subplot, solid line: Algorithm 1; dash-dotted line: GLM–SPM method. The four rows correspond to four tasks (hits, instructions, faces, velocity), respectively. In each subplot, the average prediction accuracy marked by "*" is determined by θ as in Section II-A.

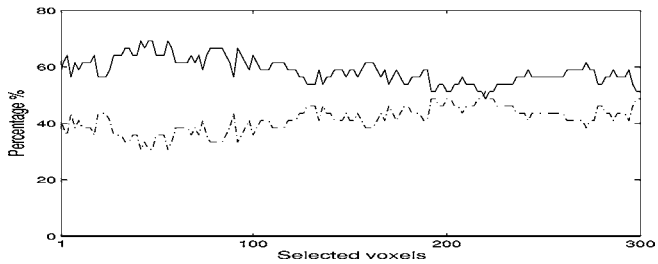

Fig. 5. Two percentage curves ( ) showing the performance of two methods in cases (three subjects and 13 tasks). Solid line (top): Algorithm 1; dash-dotted line: GLM–SPM.

accuracy (averaged over twofolds) for the four tasks are marked with a "*" in Fig. 4. Even though N does not achieve the best prediction accuracy, it had led to a satisfactory result. From this analysis, we conclude that the method described in Section II-A for selecting θ is acceptable, except for the task 3 of subject 1 in which the correlation between fMRI data and this task is always low.

Until now, we have presented our analysis results for four tasks. Considering that tasks are available for the three subjects in the dataset, we have cases (one case corresponds to one task and one subject). We analyzed each case as described earlier for the four tasks. After obtaining the prediction accuracy averaged over twofolds of cross-validation by Algorithm 1, we count the number of times, r_i that Algorithm 1 shows the best prediction accuracy among the four methods for each N_R b_i, $\mathbf{b}$, , ,..., . Performing the similar counting for GLM–SPM method, we obtain r_i . Next, we calculate two ratios (percentages) for the two methods: $r_i/$ $\times$, r_i / $\times$. Fig. 5 shows the two ratio curves, from which we can see Algorithm 1 has higher ratios.

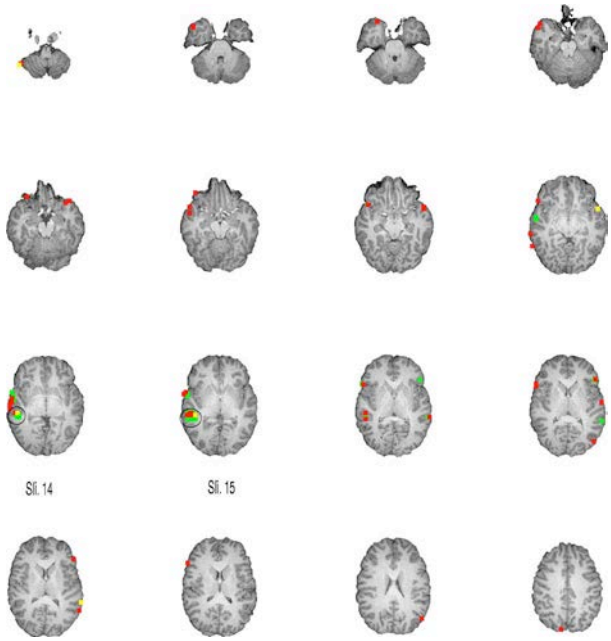

Fig. 6. Distribution of 100 selected voxels (radiological view, L R), of which 50 voxels (highlighted in red) correspond to the first 50 highest weights calculated by Algorithm 1, while the other 50 voxels (highlighted in green) correspond to the first highest values of t-statistics calculated by GLM–SPM method for task 2 (instructions), run 1, and subject 3. If two voxels obtained by two methods are the same, then it is highlighted in yellow. Slices are numbered from inferior to the superior parts of the brain.

Localization: Now, we analyze the effectiveness of Algorithm 1 in selecting voxels from the perspective of information-based detection of task-relevant brain regions rather than the perspective of predictive ability. Each of the four tasks evaluated here (hits, instructions, faces, velocity), can be related to activity in specific region(s) of the brain. For example, we should expect motor cortex (involved in movement) and supplementary motor area (involved in planning actions) to contain information relevant to the hits and velocity events, auditory cortex is expected to contain information relevant to the instructions event, and the fusiform face area (FFA) located on the ventral surface of the temporal lobe is expected to contain information for events involving faces.

Now, we choose two representative cases to show our results of localization. The first case is for task 2 (instructions), run 1, and subject 3, while the second case is for task 3 (faces), run 1, and subject 2. In this two cases, Algorithm 1 has better performance of prediction than GLM–SPM method. For the other cases in which our algorithm has better performance of prediction, we have similar conclusion presented in the following.

For instructions task, Fig. 6 shows 100 voxels, of which 50 voxels are selected by Algorithm 1, while the other 50 voxels are selected by GLM–SPM method. The brain slices are in radiological space (L R). It follows from this figure that there are several voxels that can be selected by both Algorithm 1 and GLM–SPM method. Furthermore, many voxels selected by the two methods are close in locations although they are not overlapped. However, the voxels selected by GLM–SPM method are mainly located in slices 14 and 15, which form a

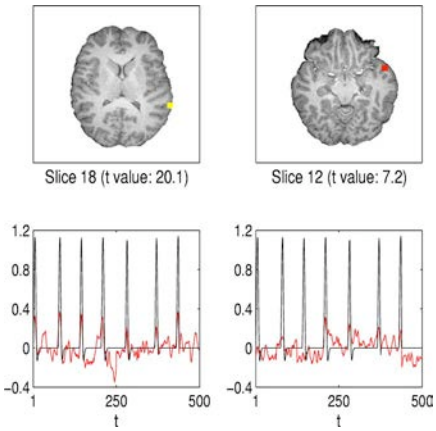

Fig. 7. (First row) Two voxels of which the first one is selected by both Algorithm 1 and GLM–SPM method and the second one is selected only by Algorithm 1. The t values in the brackets are obtained in the SPM detection. (Second row) fMRI signals corresponding to the two voxels, respectively.

TABLE II
SIX VOXELS WITH THEIR TALAIRACH COORDINATES AND CORRESPONDING BRAIN AREAS (NEUROLOGICAL SPACE, L L) SELECTED BY ALGORITHM 1 (THE NUMBERS IN THE VOXEL COLUMN: THE INDEXES OF THE SELECTED VOXELS IN THE COORDINATE DATA PROVIDED BY PBAIC 2007; ABBREVIATIONS—R: RIGHT, L: LEFT, S: SUPERIOR, M: MIDDLE, I: INFERIOR, F: FRONTAL, T: TEMPORAL, O: OCCIPITAL, FF: FUSIFORM, G: GYRUS)

Voxels	Subjects	Tasks	x	y	z	Description
10684	3	Instr.	53.5	-25.8	-10.5	R.M.T.G. (BA 21)
10776	3	Instr.	49.5	-15.8	-5.9	R.S.T.G. (BA 22)
22849	3	Instr.	-51.5	18.5	19.3	L.I.F.G. (BA 45)
178	2	Faces	-47.5	-35.8	-16.7	L.FF.G. (BA 37)
2804	2	Faces	53.5	-58.5	-3.8	R.M.O.G. (BA 19)
15087	2	Faces	45.5	25.3	39.3	R.M.F.G. (BA 8)

cluster (highlighted with circles in Fig. 6). The voxels selected by Algorithm 1, which do not form clusters, are distributed in more slices than those selected by GLM–SPM method.

There exist several voxels that can be selected by Algorithm 1, but not by GLM–SPM method, and useful for prediction/decoding. The two subplots in the first row of Fig. 7 show two voxels, in which the first one with the highest t value in SPM detection is selected by both Algorithm 1 and GLM–SPM method, the second one is selected only by Algorithm 1 other than GLM–SPM method. The second row show the corresponding fMRI signals of the two voxels. Obviously, the fMRI signals of the two voxels contain useful information related to the task.

We can see that most of these voxels selected by the two methods are in the *appropriate* areas of the brain. For instance, rows 2–4 in Table II show three typical voxels with their coordinates in the normalized Talairach space selected by Algorithm 1. These voxels correspond, as expected, to language areas, including Brodmann areas (BA) 21, 22, 45 (Broca's area) [9].

For faces task, Fig. 8 shows the distribution of the voxels, of which voxels are selected by Algorithm 1, while the other voxels are selected by GLM–SPM method. Similarly as in Fig. 6, we can see that several voxels (highlighted in yellow) selected by both Algorithm 1 and GLM–SPM method are common or close. However, the voxels selected by GLM–SPM method form two clusters, one is located in slice 11 (highlighted

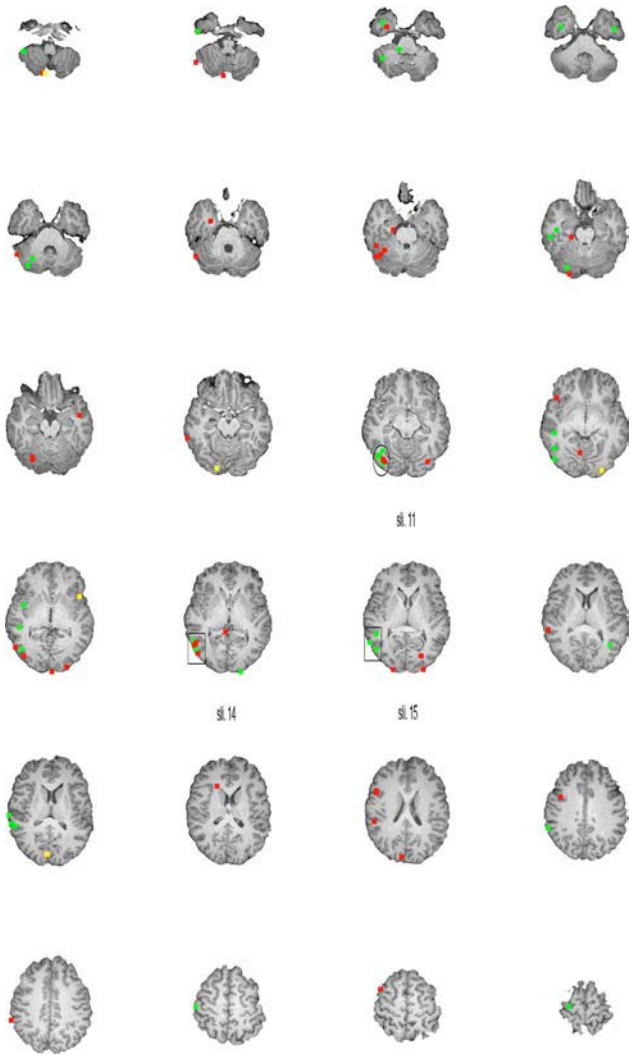

Fig. 8. Distribution of selected voxels (radiological view, L R), of which voxels (highlighted in red) correspond to the first highest weights calculated by Algorithm 1, while the other voxels (highlighted in green) correspond to the first highest values of t-statistics calculated by GLM–SPM method for task 3 (faces), run 1, and subject 2. If two voxels obtained by two methods are the same, then it is highlighted in yellow. Slices are numbered from inferior to the superior parts of the brain.

with a circle in Fig. 8), the other one is located in slices 14 and 15 (highlighted with rectangles in Fig. 8). Most of these voxels selected by the two methods are in the *appropriate* areas of the brain. For instance, rows 4–7 in Table II show three typical voxels with their coordinates in the normalized Talairach space, which are selected by Algorithm 1. These voxels correspond, as expected, to visual functional areas including BAs 37 (Fusiform area), 8 (including frontal eye fields), 19.

Algorithm 1 based on sparse representation selects a combination of voxels that are generally distributed in wider brain areas than those selected by GLM–SPM method. Since this combination of voxels is used to represent the corresponding task function, Algorithm 1 is more suitable for prediction/decoding of tasks in many cases than GLM–SPM method. Conversely, since it is easily for GLM–SPM method to show cluster effect, this method is suitable for localization of active brain areas.

Parameters setting: In the following, we present our parameter settings of Algorithm 1 in this data analysis section.

1) From our analysis, the number of initially selected voxels does not make any significant contributions to the results. This parameter is set to 500 voxels in this paper. We have tested our methods with 1000 and 2000 voxels for initial selection and confirmed the insensitivity of the results to this parameter.
2) The number of iterations corresponding to the parameter α in Algorithm 1 is fixed to 600. From the fact that Algorithm 1 is convergent (Section II-A) and that the changes in weights after 300 iterations are not significant (see Fig. 3), the obtained results should be consistent as long as the number of iterations is sufficiently large.
3) The number of constraints L in (8) is set to except that it is specifically pointed out. The results of Algorithm 1 are not very sensitive to L provided that L is not too small. In Appendix I, we analyze the sensitivity of the results to L and provide supports for the previous statement. Another approach toward selecting L would be to choose its value randomly in each iteration. Although this is valid, we concluded from our analysis that the results are not significantly better. Therefore, the value of L is chosen to be small, but big enough for Algorithm 1 to be valid so that the computational burden is minimized.
4) The parameter θ in Algorithm 1 (corresponding to N_R) determines the number of voxels selected by Algorithm 1. If the objective is to localize important voxels, it can be set as in Section II-A. If the objective is to predict tasks as described previously, this number can be chosen from a wide range (see Fig. 4). Typically, it can be set to a number around . Another method for setting θ is cross-validation, which is described in Appendix II.

IV. Conclusion

In this paper, we presented an iterative detection algorithm based on sparse representation. Then, we analyzed its convergence and effectiveness. This algorithm can be used for feature selection, localization, novelty detection, etc., as 1-norm SVM. Here, we presented one application for voxel (feature) selection in fMRI data analysis. The results demonstrate that this method can be used for voxel selection in the cases of both repeated stimulus/tasks (e.g., instructions task) and nonrepeated stimulus/tasks (e.g., faces task). The sparse representation model used in our algorithm can be seen as the opposite of the GLM model widely used for fMRI analysis; however, there exists significant difference between the two models. In the sparse representation model, many voxels are considered simultaneously, but the task/stimulus conditions are considered separately. Using our algorithm, a combination of voxels are selected. This combination of voxels plays an important role in effective/sparse representation of a task/stimulus function. Conversely, voxels are considered separately, but the task/stimulus conditions are considered simultaneously in GLM model. The validity of our method was shown through the comparison with GLM–SPM method in our data analysis.

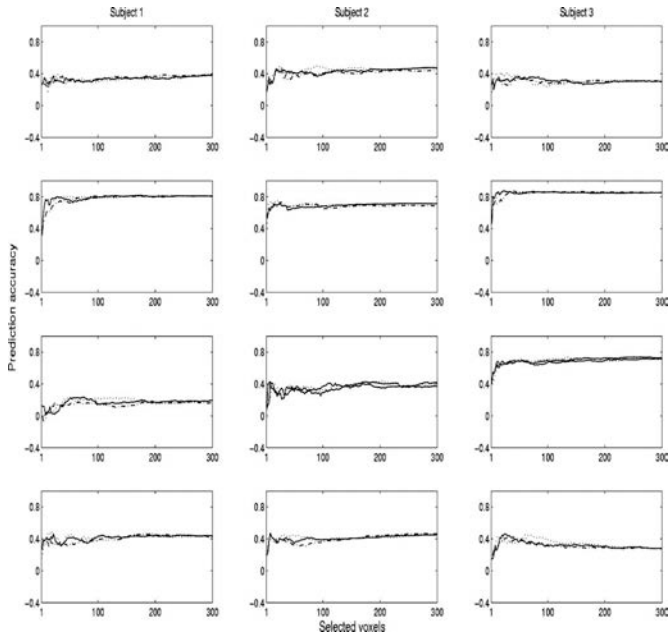

Fig. 9. Three prediction accuracy curves obtained by Algorithm 1 with the numbers of constraints in (8) set to be 25 (dotted line), 50 (solid line), and 75 (dashed-dotted line), respectively, in Appendix I. The four rows correspond to four tasks (hits, instructions, faces, velocity), respectively.

In addition to voxel selection as shown in this paper, the sparse representation approach also can be used for decoding a task based on the information contained in the fMRI BOLD signal. A simple implementation strategy can be the following: first, a sparse regression model is trained using model (2), where $\mathbf{A}$ is a known fMRI data matrix constructed from selected voxels and $\mathbf{y}$ is a measured task function. Then, task function for the same task can be decoded during future acquisitions using model (1), where $\mathbf{A}$ is a new fMRI data matrix of selected voxels, $\mathbf{w}$ is a sparse coefficient vector obtained in the training phase. Our initial analysis results has shown the effectiveness of this decoding method in several cases. However, investigation is required in order to improve the performance of this decoding method since the constraint equation in model (2) is underdetermined and noise is neglected.

Appendix I

On Setting the Number of Constraints in Algorithm 1

In this appendix, we first show our results obtained by Algorithm 1 with the numbers L of constraints in (8) set to be L , L , and L , respectively.

Let $\mathbf{b}$, , ,..., as in Section III. For each pair of i (i ,...,) and L (L , ,), we set N_R b_i and obtain average prediction accuracies by Algorithm 1 with parameter L (averaged over twofolds of cross-validation) for the tasks, and subjects. Each subplot of Fig. 9 shows three prediction accuracy curves (with respect to b_i) for L , , and , respectively. Note that the prediction accuracy curves for L (solid lines) are the same as those in Fig. 4. By comparing these curves in each subplot of Fig. 9, we can see that the prediction results are not sensitive to the parameter L.

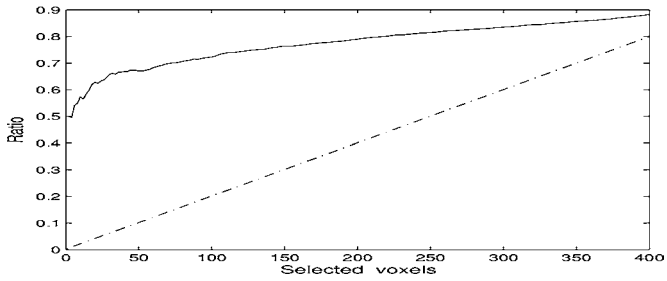

Fig. 10. Ratio curves showing the degree that the two selected voxel sets are overlapped in Appendix I. (Solid line) $r\ b_l$ obtained by Algorithm 1; (dashed-dotted line) $r\ b_l$ obtained by randomly taking two sets of voxels.

Next, we compare the sets of voxels selected by Algorithm 1 with its parameter L set as L and L, respectively. For the ith task (i $,\ldots,$), the jth subject (j , ,), the kth run (k ,), we calculate weight vectors $\mathbf{w}_{L\ ,i,j,k}$ and $\mathbf{w}_{L\ ,i,j,k}$ using Algorithm 1. For a given number b_l (b_l , , $,\ldots,$), we choose two sets of voxels $R_{L\ ,b_l}\ i,j,k$ and $R_{L\ ,b_l}\ i,j,k$ from $\mathcal{N}$ (the initially selected voxels) using $\mathbf{w}_{L\ ,i,j,k}$ and $\mathbf{w}_{L\ ,i,j,k}$, respectively. Furthermore, we calculate the number $q\ i,j,k,b_l$ of voxels $\in R_{L\ ,b_l}\ i,j,k\ \cap R_{L\ ,b_l}\ i,j,k$ and the ratio ratio i,j,k,b_l $q\ i,j,k,b_l\ /b_l$. Averaging ratio i,j,k,b_l across i,j,k, we obtain the mean of ratio i,j,k,b_l denoted as $r\ b_l$. Fig. 10 shows the curve of $r\ b_l$ with b_l (solid line).

For each b_l (b_l , , $,\ldots,$), we also randomly take two subsets of $\mathcal{N}$. Denote the two subsets as $R\ _{,b_l}$ and $R\ _{,b_l}$, each of which contains b_l voxels. We also calculate the ratio $r\ b_l$ with which $R\ _{,b_l}$ and $R\ _{,b_l}$ are overlapped. The curve of $r\ b_l$ is shown as the dashed line in Fig. 10.

From Fig. 10, we can see that if $b_l > \approx$, $r\ b_l >$. Furthermore, the ratio $r\ b_l \gg r\ b_l$. Therefore, the two voxel sets $R_{L\ ,b_l}\ i,j,k$ and $R_{L\ ,b_l}\ i,j,k$ determined by Algorithm 1 with two constraint parameters L and L, respectively, are overlapped to a high degree, i.e., most of the voxels selected by Algorithm 1 with different parameters L are the same. This is possibly why the results obtained by Algorithm 1 are not sensitive to L.

APPENDIX II

CROSS-VALIDATION METHOD FOR SETTING THE NUMBER OF VOXELS FOR PREDICTION OF TASKS IN MRI DATA

In this section, we show a cross-validation method for setting θ in Algorithm 1, which determines the number of voxels used to predict the tasks. First, for each subject and task, the dataset (including fMRI data and task data) of run 1 and run 2 is equally divided into four parts, each consisting of 250 time points. The first three parts are used for threefold cross-validation, while the fourth part is used as an independent test dataset. In one of the threefold cross-validation, we use two parts for training, and predict the task of the left part. For each value of N_R (1–500, the number of selected voxels), a prediction accuracy of the validation feature is obtained. After the threefold cross-validation is performed, three the prediction accuracy curves (with respect to N_R) are obtained. An average prediction accuracy curve is then obtained by taking the mean of the three ones. Suppose that this average prediction accuracy curve has maximum at N, i.e., if N voxels are selected, the average prediction accuracy is the maximum.

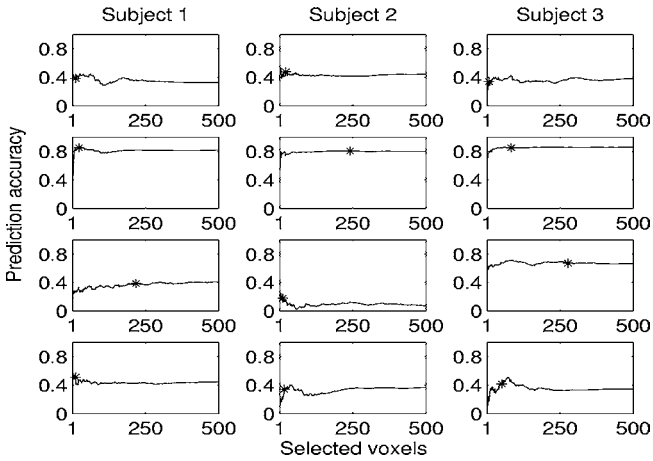

Fig. 11. Average prediction accuracy curves for independent test set for three subjects and four tasks in Appendix II, where the four rows correspond to the four tasks (hits, instructions, faces, velocity), respectively. In each subplot, "*": average prediction accuracy determined by θ.

Next, we use any two of the first three parts for training, and obtain a weight vector $\mathbf{w}$. If we rearrange the vector $|\mathbf{w}|$ (the absolute value vector) in descending order, and denote it as $|w_i\ |,\ldots,|w_i\ |$, then θ $|w_{i_N}\ |$. We now predict the task of the independent test set using the N voxels with weights $\{w_i\ ,\ldots,w_{i_N}\ \}$. Three prediction accuracies are obtained using different combinations of two parts for training, i.e. the three folds of cross-validation. Similarly as before, we also obtain an average prediction accuracy curve with respect to N_R (1–500) for the task of this independent test set. Note that the value of N is the point at which the average of the prediction curves of the threefolds is maximum. In each subplot of Fig. 11, an average prediction accuracy curve for the independent test set is presented for one task of one subject and the "*" represents the average prediction accuracy determined by θ. From this figure, we can see that this cross-validation method for determining the parameter θ is acceptable.

APPENDIX III

ROBUSTNESS ANALYSIS

In this appendix, we analyze the robustness of Algorithm 1 to noise. Consider the following noisy model corresponding to (2):

$$\|\mathbf{w}\|\ , \qquad \text{s.t.}\ \ \mathbf{A}\quad \mathbf{V}\ \mathbf{w}\quad \mathbf{y} \tag{17}$$

where $\mathbf{V} \in R^{N\times M}$ is a noise matrix. The optimal solution of (17) is denoted by $\mathbf{w}_v$. In this paper, $\mathbf{A}$ and $\mathbf{y}$ are a data matrix and a transformed task function, thus we add noise to $\mathbf{A}$ other than $\mathbf{y}$.

Denote all $N \times N$-dimensional submatrices of $\mathbf{A}$ and $\mathbf{V}$ as $\mathbf{A}^{\ j}$ and $\mathbf{V}^{\ j}$, respectively, where j $,\ldots,C_M^N$. Since $\mathbf{A}$ is a real data matrix, we can assume that all submatrices $\mathbf{A}^{\ j}$ are

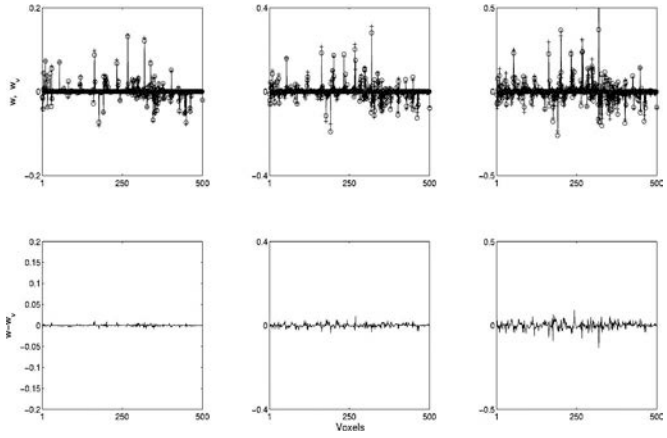

Fig. 12. Each of the three subplots of the first row shows the weight vectors $\mathbf{w}$ (with "o") and $\mathbf{w}_v$ (with " ") obtained by Algorithm 1 after iterations in noiseless case and noise case, respectively, in Appendix III, where $\mathbf{A}$ and $\mathbf{y}$ are the same fMRI data matrix and transformed task function as in Fig. 1, $\mathbf{w}_v$ corresponds to dB simulated zero mean Gaussian noise. Each of the three subplots of the second row shows the error $\mathbf{w} - \mathbf{w}_v$. The three columns of this figure correspond to three parameter settings of Algorithm 1: L , , , respectively.

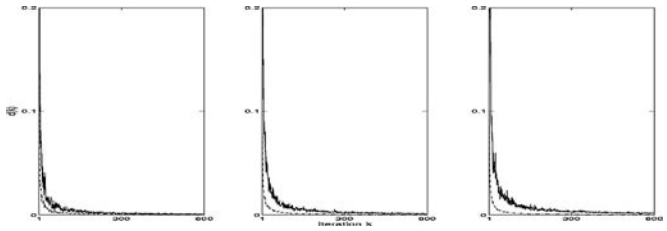

Fig. 13. Each subplot shows two iterative curves of Algorithm 1 obtained in noiseless case (solid curve) and dB noise case (dashed-dotted curve) in Appendix III, where $\mathbf{A}$ and $\mathbf{y}$ are the same fMRI data matrix and transformed task function, as shown in Fig. 1. The three subplots correspond to the three settings , , and of the parameter L in Algorithm 1, respectively.

full of rank. In this case, it follows from linear programming theory [19] that there is a submatrix say $\mathbf{A}^{j}$, such that the 1-norm solution $\mathbf{w}$ of (2) satisfies

$$\mathbf{w} \quad \mathbf{A}^{j} \quad \mathbf{y} \tag{18}$$

i.e.,

$$\| \mathbf{A}^{j} \quad \mathbf{y}\| \qquad \{\| \mathbf{A}^{j} \quad \mathbf{y}\| , j \quad , \ldots, C_M^N\}. \tag{19}$$

Note that

$$\|\mathbf{V}\| \to \| \mathbf{A}^{j} \quad \mathbf{V}^{j} \quad \mathbf{y}\| \quad \| \mathbf{A}^{j} \quad \mathbf{y}\| , j \quad , \ldots, C_M^N. \tag{20}$$

It follows from (19) and (20) that when $\|\mathbf{V}\|$ is sufficiently small

$$\| \mathbf{A}^{j} \quad \mathbf{V}^{j} \quad \mathbf{y}\| \qquad \{\| \mathbf{A}^{j} \quad \mathbf{V}^{j} \quad \mathbf{y}\| \quad j \quad , \ldots, C_M^N\}. \tag{21}$$

i.e., $\mathbf{A}^{j} \quad \mathbf{V}^{j} \quad \mathbf{y}$ is the 1-norm solution $\mathbf{w}_v$ of (17) with sufficiently small noise. Furthermore, since $\mathbf{w}_v$ is close to $\mathbf{w}$ in (18), $\mathbf{w}_v$ can be represented by

$$\mathbf{w}_v \quad \mathbf{w} \quad \triangle\mathbf{w}_v \tag{22}$$

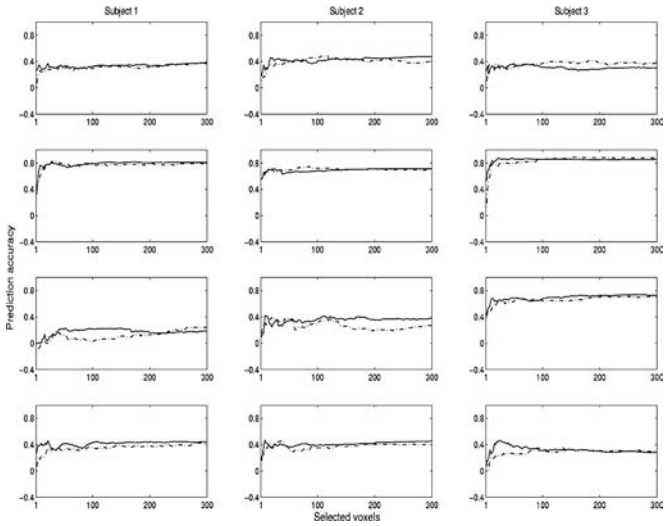

Fig. 14. Each subplot shows two prediction accuracy curves obtained by Algorithm 1 in noiseless case (solid curve) and dB noise case (dashed-dotted curve), respectively, in Appendix III. The four rows correspond to four tasks (hits, instructions, faces, velocity), respectively.

where $\triangle\mathbf{w}_v$ is a disturbance vector resulted by the noise matrix $\mathbf{V}$. It follows from (20) that $\triangle\mathbf{w}_v$ is small if $\|\mathbf{V}\|$ is sufficiently small.

In the following, we consider the following noisy model corresponding to the model (8) in Algorithm 1,

$$\|\mathbf{w}\| , \qquad \text{s.t.} \quad \mathbf{A}_k \quad \mathbf{V}_k \ \mathbf{w} \quad \mathbf{y}_k \tag{23}$$

where $\mathbf{V}_k \in R^{L\times M}$ is a noise matrix. The optimal solution of (23) is denoted by $\mathbf{w}_v^{\,k}$. From the previous analysis, $\mathbf{w}_v^{\,k}$ can be represented by

$$\mathbf{w}_v^{\,k} \quad \mathbf{w}^{\,k} \quad \triangle\mathbf{w}^{\,k} \tag{24}$$

where $\mathbf{w}^{\,k}$ is the solution of (8).

Thus, the output of Algorithm 1 after K iterations in noise case is

$$\mathbf{w}_v^{\,K} \quad \overline{K} \sum_i^K \mathbf{w}^{\,i} \quad \overline{K} \sum_i^K \triangle\mathbf{w}_v^{\,i} . \tag{25}$$

When $\|\mathbf{V}_k\|$ $(k \quad , \ldots, K)$ are sufficiently small, $/K$ $\sum_i^K \triangle\mathbf{w}_v^{\,i}$ is close to zero. This is due to the following reasons: 1) $\|\triangle\mathbf{w}_v^{\,i}\|$ is small, their mean is still small and 2) the mean of each entry of $\triangle\mathbf{w}_v^{\,i}$ is generally zero. Thus, we have

$$\mathbf{w}_v^{\,K} \approx \mathbf{w}^{\,K} \tag{26}$$

where $\mathbf{w}^{\,K}$ is the output of Algorithm 1 after K iterations in noiseless case, i.e., the weight vector obtained by Algorithm 1 is robust to noise at least to some degree.

Fig. 12 shows three pairs of $\mathbf{w}$ (black) and $\mathbf{w}_v$ obtained by Algorithm 1 in noiseless case and dB noise case, respectively. When the noise is sufficiently small, we can see that $\mathbf{w}_v$ is close to $\mathbf{w}$.

Now, we enlarge the additive noise in model (23) to 25 dB, data analysis results (see Figs. 13 and 14) show that the weight

vector obtained by Algorithm 1 is still effective for voxel selection.

ACKNOWLEDGMENT

The authors are grateful to the organizers of Pittsburgh Brain Activity Interpretation Competition 2007 for providing the fMRI data. The authors are grateful to Dr. P. Sun (Brain Science Institute, RIKEN, Japan) for his contribution in the fMRI data analysis part of this paper. The authors are also grateful to the anonymous reviewers for their insightful and constructive suggestions.

REFERENCES

[1] Y. Kamitani and F. Tong, "Decoding the visual and subjective contents of the human brain," *Nat. Neurosci.*, vol. 8, no. 5, pp. 679–685, 2005.

[2] K. J. Friston, P. Jezzard, and R. Turner, "Analysis of functional MRI time series," *Hum. Brain Mapp.*, vol. 1, pp. 153–171, 1994.

[3] K. J. Friston, A. P. Holmes, J. B. Poline, P. J. Grasby, S. C. Williams, R. S. Frackowiak, and R. Turner, "Analysis of fMRI time-series revisited," *NeuroImage*, vol. 2, pp. 45–53, 1995.

[4] K. J. Friston, A. P. Holmes, K. Worsley, J. B. Poline, D. Frith, and R. S. Frackowiak, "Statistical parameter maps in functional imaging: A general linear approach," *Hum. Brain Mapp.*, vol. 2, pp. 189–210, 1995.

[5] D. D. Cox and R. L. Savoy, "Functional magnetic resonance imaging (fMRI) brain reading: Detecting and classifying distributed patterns of fMRI activity in human visual cortex," *Neuroimage*, vol. 19, pp. 261–270, 2003.

[6] T. M. Mitchell, R. Hutchinson, R. S. Niculescu, F. Pereira, X. Wang, M. Just, and S. Newman, "Learning to decode cognitive states from brain images," *Mach. Learning*, vol. 57, pp. 145–175, 2004.

[7] J. V. Haxby, M. I. Gobbini, M. L. Furey, A. Ishai, J. L. Schouten, and P. Pietrini, "Distributed and overlapping representations of faces and objects in ventral temporal cortex," *Science*, vol. 293, pp. 2425–2430, 2001.

[8] A. Battle, G. Chechik, and D. Koller, "Temporal and cross-subject probabilistic models for fMRI prediction task," in *Advances in Neural Information Processing Systems*. vol. 19, Cambridge, MA: MIT Press, 2006, pp. 146–153.

[9] M. K. Carroll, G. A. Cecchi, I. Rish, R. Garg, and A. R. Rao, "Prediction and interpretation of distributed neural activity with sparse models," *NeuroImage*, vol. 44, no. 1, pp. 112–122, 2009.

[10] F. Meyer and G. J. Stephens, "Locality and low-dimensions in the prediction of natural experience from fMRI," presented at the 21st Annu. Conf. Neural Inf. Process. Syst., Vancouver, Canada, 2007.

[11] S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 33–61, 1998.

[12] B. A. Olshausen, P. Sallee, and M. S. Lewicki, "Learning sparse image codes using a wavelet pyramid architecture," in *Advances in Neural Information Processing Systems*. vol. 13, Cambridge, MA: MIT Press, 2001, pp. 887–893.

[13] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1?," *Vis. Res.*, vol. 37, pp. 3311–3325, 1997.

[14] M. S. Lewicki and T. J. Sejnowski, "Learning overcomplete representations," *Neural Comput.*, vol. 12, no. 2, pp. 337–365, 2000.

[15] R. Gribonval and M. Nielsen, "Sparse decompositions in unions of bases," *IEEE Trans. Inf. Theory*, vol. 49, no. 12, pp. 3320–3325, Dec. 2003.

[16] J. A. Tropp, A. C. Gilbert, S. Muthukrishnan, and M. J. Strauss, "Improved sparse approximation over quasi-incoherent dictionaries," in *Proc. 2003 IEEE Int. Conf. Image Process.*, Barcelona, Spain, Sep., vol. 1, pp. I-37–I-40.

[17] D. L. Donoho and M. Elad, "Maximal sparsity representation via l minimization," *Proc. Nat. Acad. Sci. U.S.A.*, vol. 100, pp. 2197–2202, Mar. 2003.

[18] M. Girolami, "A variational method for learning sparse and overcomplete representations," *Neural Comput.*, vol. 13, no. 11, pp. 2517–1532, 2001.

[19] Y. Q. Li, A. Cichocki, and S. Amari, "Analysis of sparse representation and blind source separation," *Neural Comput.*, vol. 16, pp. 1193–1234, 2004.

[20] Y. Li, S. I. Amari, A. Cichocki, and C. Guan, "Probability estimation for recoverability analysis of blind source separation based on sparse representation," *IEEE Trans. Inf. Theory*, vol. 52, no. 7, pp. 3139–3152, Jul. 2006.

[21] Y. Li, S. I. Amari, A. Cichocki, D. W. C. Ho, and S. Xie, "Underdetermined blind source separation based on sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 2, pp. 423–437, Feb. 2006.

[22] E. Kidron, Y. Y. Schechner, and M. Elad, "Cross-modal localization via sparsity," *IEEE Trans. Signal Process.*, vol. 55, no. 4, pp. 1390–1404, Apr. 2007.

[23] M. Zibulevsky and B. A. Pearlmutter, "Blind source separation by sparse decomposition," *Neural Comput.*, vol. 13, no. 4, pp. 863–882, 2001.

[24] J. Bi, P. Bennett, M. Embrechts, C. M. Breneman, and M. Song, "Dimensionality reduction via sparse support vector machines," *J. Mach. Learning Res.*, vol. 3, pp. 1229–1243, 2003.

[25] J. Bi, Y. Chen, and J. Wang, "A sparse support vector machine approach to region-based image categorization," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recogn. (CVPR 2005)*, vol. 1, pp. 1121–1128.

[26] J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani, "1-norm support vector machines," in *Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2003, p. 16.

[27] A. J. Smola, B. Scholkopf, and G. Gatsch, "Linear programs for automatic accuracy control in regression," in *Proc. Int. Conf. Artif. Neural Netw.*. Berlin, Germany: Springer-Verlag, 1999, pp. 575–580.

[28] K. P. Bennett, "Combining support vector and mathematical programming methods for classification," in *Advances in Kernel Methods C Support Vector Machines*, B. Scholkopf, C. Burges, and A. Smola, Eds. Cambridge, MA: MIT Press, 1999, pp. 307–326.

[29] C. Campbell and K. P. Bennett, "A linear programming approach to novelty detection," *Neural Inf. Process. Syst.*, vol. 13, pp. 395–401, 2000.

[30] R. Tibshirani, "Regression selection and shrinkage via the LASSO," *J. R. Stat. Soc., Ser. B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[31] S. Hochreiter and K. Obermayer, "Support vector machines for dyadic data," *Neural Comput.*, vol. 18, no. 6, pp. 1472–1510, 2006.

[32] Y. Li, P. Namburi, C. Guan, and J. Feng, "A sparse representation based algorithm for voxel selection in fMRI data analysis," presented at the 14th Annu. Meet. Org. Hum. Brain Mapp., Melbourne, Vic., Australia, 2008.

[33] I. V. Tetko and A. E. P. Villa, "A comparative study of pattern detection algorithm and dynamical system approach using simulated spike trains," *Lect. Notes Comput. Sci.*, vol. 1327, pp. 37–42, 1997.

[34] W. Schneider and G. Siegle. (2007). *Pittsburgh Brain Activity Interpretation Competition 2007 Guide Book: Interpreting subject-driven actions and sensory experience in a rigorously characterized virtual world* [Online]. Available: http://www.braincompetition.org

Yuanqing Li (M'06) was born in Hunan Province, China, in 1966. He received the B.S. degree in applied mathematics from Wuhan University, Wuhan, China, in 1988, and the M.S. degree in applied mathematics and the Ph.D. degree in control theory and applications from South China Normal University, Guangzhou, China, in 1994 and 1997, respectively.

Since 1997, he has been with South China University of Technology, where he became a Full Professor in 2004. During 2002–2004, he was a Research Fellow at the Laboratory for Advanced Brain Signal Processing, RIKEN Brain Science Institute, Saitama, Japan. During 2004–2008, he was a Research Scientist at the Laboratory for Neural Signal Processing, Institute for Infocomm Research, Singapore. His current research interests include blind signal processing, sparse representation, machine learning, brain–computer interface, EEG, and fMRI data analysis. He is the author or coauthor of more than 60 scientific papers in journals and conference proceedings.

Praneeth Namburi received the Bachelor's degree in 2008 from Nanyang Technological University, Singapore, Singapore.

In 2008, he joined DUKE-NUS Graduate Medical School, Singapore, as a Research Assistant. He is also with Institute for Infocomm Research, Singapore. His current research interests include decoding information from functional MRI data and cognitive neuroscience of attention.

Zhuliang Yu (M'06) received the B.S.E.E. degree in 1995 and the M.S.E.E. degree in 1998, both in electronic engineering from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, and the Ph.D. degree in 2006 from Nanyang Technological University, Singapore, Singapore.

From 1998 to 2000, he was a Software Engineer in Shanghai BELL Company Ltd. In 2000, he joined the Center for Signal Processing, Nanyang Technological University, as a Research Engineer, and then as a Research Fellow. In 2008, he joined the College of Automation Science and Engineering, South China University of Technology, as an Associate Professor. His current research interests include array signal processing, acoustic signal processing, adaptive signal processing and their applications in communications, biomedical engineering, etc.

Cuntai Guan (S'91–M'92–SM'03) received the Ph.D. degree in electrical and electronic engineering from Southeast University, Nanjing, China, in 1993.

From 1993 to 1994, he was at the Southeast University, where he was engaged in speech vocoder, speech recognition, and text-to-speech. During 1995, he was a Visiting Scientist at the Centre de Recherche en Informatique de Nancy (CRIN)/Centre National de la Recherche Scientifique (CNRS) Institut National de Recherche en Informatique et en Automatique (INRIA), Paris, France, where he was involved in keyword spotting. From 1996 to 1997, he was with the City University of Hong Kong, Kowloon, Hong Kong, where he was engaged in developing robust speech recognition under noisy environment. From 1997 to 1999, he was with the Kent Ridge Digital Laboratories, Singapore, Singapore, where he was involved in multilingual, large vocabulary, continuous speech recognition. He was a Research Manager and the R&D Director for five years in industries, focusing on the development of spoken dialogue technologies. In 2003, he established the Brain–Computer Interface Group at the Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR), Singapore, where he is currently a Senior Scientist and a Program Manager. His current research interests include brain–computer interface, neural signal processing, machine learning, pattern classification, and statistical signal processing, with applications to assistive device, rehabilitation, and health monitoring.

Jianfeng Feng received the Ph.D. degree from the Probability and Statistics Department, Peking University, P. R. China, in 1993.

Currently, he is a Professor in the Centre for Scientific Computing, Warwick University, Coventry, U.K., and the Director of the Centre for Computational Systems Biology, Fudan University, China. He has authored or coauthored more than 130 journal papers in journals, such as *Current Biology*, *Journal of Neuroscience*, *Physics Review Letters*, *PLoS Computational Biology,* etc.

Zhenghui Gu (M'00) received the Ph.D. degree from Nanyang Technological University, Singapore, Singapore, in 2003.

From 2002 to 2008, she was with the Institute for Infocomm Research, Singapore. In 2009, she joined the College of Automation Science and Engineering, South China University of Technology, Guangzhou, China, as an Associate Professor. Her current research interests include the fields of signal processing and pattern recognition.