J Glob Optim (2014) 59:439–457
DOI 10.1007/s10898-013-0134-2

Sparse optimization in feature selection: application in neuroimaging

K. Kampa · S. Mehta · C. A. Chou · W. A. Chaovalitwongse · T. J. Grabowski

Received: 8 June 2013 / Accepted: 18 December 2013 / Published online: 8 January 2014

Abstract Feature selection plays an important role in the successful application of machine learning techniques to large real-world datasets. Avoiding model overfitting, especially when the number of features far exceeds the number of observations, requires selecting informative features and/or eliminating irrelevant ones. Searching for an optimal subset of features can be computationally expensive. Functional magnetic resonance imaging (fMRI) produces datasets with such characteristics creating challenges for applying machine learning techniques to classify cognitive states based on fMRI data. In this study, we present an embedded feature selection framework that integrates sparse optimization for regularization (or sparse

K. Kampa · W. A. Chaovalitwongse (✉)
Department of Industrial and Systems Engineering, University of Washington, Seattle, WA, USA
e-mail: artchao@uw.edu

K. Kampa · W. A. Chaovalitwongse · T. J. Grabowski
Integrated Brain Imaging Center, University of Washington Medical Center, Seattle, WA, USA
e-mail: kittipat@gmail.com

S. Mehta
Department of Radiology, University of Washington, Seattle, WA, USA

S. Mehta
Department of Psychology, University of Washington, Seattle, WA, USA
e-mail: mehtas2@uw.edu

C. A. Chou
Department of Systems Science and Industrial Engineering, Binghamton University, State University of New York, Vestal, NY, USA
e-mail: cachou@binghamton.edu

W. A. Chaovalitwongse · T. J. Grabowski
Department of Radiology, University of Washington, Seattle, WA, USA

T. J. Grabowski
Department of Neurology, University of Washington, Seattle, WA, USA
e-mail: tgrabow@uw.edu

regularization) and classification. This optimization approach attempts to maximize training accuracy while simultaneously enforcing sparsity by penalizing the objective function for the coefficients of the features. This process allows many coefficients to become zero, which effectively eliminates their corresponding features from the classification model. To demonstrate the utility of the approach, we apply our framework to three different real-world fMRI datasets. The results show that regularized classifiers yield better classification accuracy, especially when the number of initial features is large. The results further show that sparse regularization is key to achieving scientifically-relevant generalizability and functional localization of classifier features. The approach is thus highly suited for analysis of fMRI data.

Keywords Sparse optimization · Feature selection · Machine learning · fMRI · Cognitive neuroscience · Regularization · Pattern classification

1 Introduction

The availability of large datasets in real-world applications poses significant challenges in optimization and machine learning. These massive datasets are often referred to as *Big Data* as they consist of very large numbers of data samples as well as features. Feature selection plays a pivotal role in the analysis of such data as it enables the extraction of salient information to base decisions. As Big Data are very high-dimensional, this step reduces the likelihood of model overfitting and computational complexity of decision models. Feature selection is a process of selecting a subset of the original features according to certain criteria [59]. Not only does feature selection reduce the dimensionality of the data, but it also increases the signal-to-noise ratio by removing irrelevant, redundant or noisy features, which in turns improves the performance of decision models in terms of prediction accuracy, result interpretability and computational run-time.

Feature selection is an optimization problem by nature. Its objective is to find the optimal subset of features that can achieve the best performance on some criterion (e.g., prediction accuracy). If the number of original features is p, the number of possible subsets is $2^p - 1$. Even if the number of features to be selected is known, k, there are still $\binom{p}{k}$ subsets of features. Generally speaking, feature selection can be formulated as a mathematical program with p binary variables, each indicating if a feature is selected. The criteria used to select the features may be modeled as an objective function as well as included as knapsack-type selection constraints. Thus, one can generally say that feature selection problem is *NP-hard* and cannot be solved in a polynomial time [1]. The problem becomes even harder when the number of features far exceeds the number of observations (data instances). Given that n is the number of observations, such a problem is often called the "$n \ll p$" problem.

In this paper, we focus on an application of feature selection in neuroimaging. Feature selection is extremely important in neuroimaging because the features correspond to anatomical region(s), allowing inference about which brain structures are involved in cognitive processes. In addition, there are systematic sources of overfitting that need to be mitigated to allow for scientifically meaningful generalizability of classification models. Thus, selected features have real-world meaning and offer interpretability when reconstructing classification models. Multi-voxel pattern analysis (MVPA) of functional magnetic resonance imaging (fMRI) data will be the main case study in this paper. MVPA is used to study cognitive processes measured by fMRI by ascertaining where and how information is encoded in the

brain. A main focus of MVPA is to classify or "decode" different cognitive states based on patterns of neural activity measured in a feature subset of image voxels. By the nature of the functional organization of the brain, only some fMRI voxels will be relevant for decoding. The remaining voxels will be uninformative for the particular cognitive task, with their signal variance for practical purposes reflecting noise. Using all the voxels in a classification model would lead to *overfitting* and result in poor generalization. Thus, feature selection is key to building an accurate and robust classification model. Because of the duration and economic constraints of fMRI acquisition, most fMRI studies include relatively few observations (e.g., $n < 100$). Meanwhile, the number of voxels (also referred to as features) is comparatively very large (e.g., $p > 10{,}000$). Thus, MVPA is a classic "$n \ll p$" feature selection problem.

The fMRI signal is inherently multivariate, reflecting spatially distributed neural processing captured in the activity pattern across multiple voxels. Successful interrogation of cognitive representations requires joint assessment of this activity. While there have been many feature selection algorithms proposed in the literature, certain approaches are better suited to fMRI. Here we focus on an embedded feature selection framework, which includes all features in an integrated feature selection and classification model. In such a framework, sparsity is enforced in the classification model that is trained to maximize the classification accuracy and minimize the number of selected features. This sparse optimization for regularization (or sparse regularization) is very important for MVPA because feature selection allows for functional localization of cognitive processes, with sparser feature selection providing more concise localization. In this paper we focus on logistic regression (LR) with sparse regularization as a supervised feature selection and classification framework. Our contribution is to introduce, employ and evaluate the embedded feature selection framework to the application of MVPA. The framework provides an alternative approach to select features while simultaneously performing classification. The logistic regression is used because its linear model offers better interpretability in cognitive neuroscience. Three types of regularization are employed: *ridge*, *lasso* and *elastic net* penalties.

The remainder of the paper is organized as follows. In Sect. 2, we provide the background of feature selection and more details of MVPA. In Sect. 3, we present the optimization formulation of logistic regression with various types of penalty. In Sect. 4, the details of our computational framework including solution approaches, cross-validation and parameter selection procedure are given. We present the datasets and the experimental results in Sect. 5. We conclude the study in Sect. 6.

2 Background

2.1 Feature selection

The curse of dimensionality poses challenges to learning algorithms when dealing with high-dimensional data, in which the number of features is large and only a few are informative. In such a situation, learning algorithms likely overfit classification models and the learned models are less generalizable. Feature selection is a method to identify relevant features in order to improve classification accuracy and facilitate more stable and interpretable results [15,30,45]. Feature selection algorithms can be categorized as *supervised*, *semi-supervised* or *unsupervised*. Supervised feature selection algorithms [44,46,52,53] use the statistical dependency between the feature and the class variable to determine the degree of feature relevance. In an absence of class labels, unsupervised feature selection algorithms evaluate the degree of feature relevance from data variance and separability [9,22]. In a situation

where labeled data can be obtained but very expensive, semi-supervised feature selection algorithms [55,58] can use a small portion of labeled data as an additional information to improve the performance of unsupervised feature selection algorithms.

A large number of feature selection algorithms have been developed but most can be grouped into one of the three models: *filter*, *wrapper* or *embedded* [59]. The filter model depends on the characteristics of the data alone without involving the learning (e.g., classification and regression) algorithms. Many feature selection algorithms in the filter model rely on using certain metrics to rank or eliminate features. For instance, correlation [6,54,56], t-test [49,60] and mutual information (MI) [2,39,47,50,51] have been used to rank features or eliminate irrelevant features. The wrapper model requires a learning algorithm to assess the classification performance (e.g., prediction accuracy or cardinality) as evaluation criteria to select features [3,25,43]. The embedded model integrates feature selection with the classification model in the training process. Training performance and selected features are achieved simultaneously. Examples of embedded models include decision tree C4.5 [42], L_1-norm SVM [32], and logistic regression with L_1-norm regularization [10,12,24,48].

Logistic regression (LR) has been widely used as a classifier because of not only its performance, but also the interpretability and simplicity to implement. However, LR alone without regularization often results in a high variance estimation of its coefficients, especially when there are many correlated features (variables). Such an issue can be mitigated using ridge (L_2-norm) regularization to shrink the size of coefficients [23]. Nevertheless, almost all (if not all) of the coefficients still remain non-zero. Thus, this method does not possess the characteristic of feature selection. Moreover, the resulting coefficients tend to spread equally within a set of correlated features, resulting in underestimated coefficients which can often be over-enforced when performing feature selection by thresholding the coefficients. The problem can be alleviated by imposing lasso (L_1-norm) regularization [10,14,48] which introduces sparse solution compared to ridge penalty. However, this penalty tends to pick only a few features (if not only one) from a set of correlated features, yielding a very sparse solution which is often not robust in practice. L_q-norm was proposed to relieve the issue by generalizing the norm and selecting L_q-norm such that q is between 1 and 2 to combine the effect of both ridge and lasso as appeared in [11]. However, L_q-norm penalty in such range of q does not provide a sparse solution because the norm is still differentiable at zero when $q > 1$. Elastic net penalty [61] was introduced as a linear combination between ridge and lasso penalty, resulting in a compromise characteristic of both. The lasso part in elastic net penalty encourages sparse solution whereas the ridge part encourages spreading coefficients among a set of correlated features, resulting in theoretically more robust classification compared to lasso and explicit feature selection not available through ridge. More detailed explanations for each model are described in Sect. 3. Furthermore, LR is a very promising model computationally as its inference on a large dataset can also be accomplished using stochastic gradient descent [29,57], which can be parallelized in MapReduce framework, but is beyond the scope of this paper. Enthusiastic readers please refer to [4,7,26].

2.2 Multi-voxel pattern analysis (MVPA)

Conventional fMRI data analysis has relied on univariate statistical approaches to elucidate the neural basis of cognition. In such approaches, the response is assessed at each voxel in the brain independently. However, a growing body of evidence suggests that mental representations are more effectively studied by considering the joint activity of multiple voxels [19,21,37]. Thus, MVPA, adapted from machine learning and pattern recognition, has

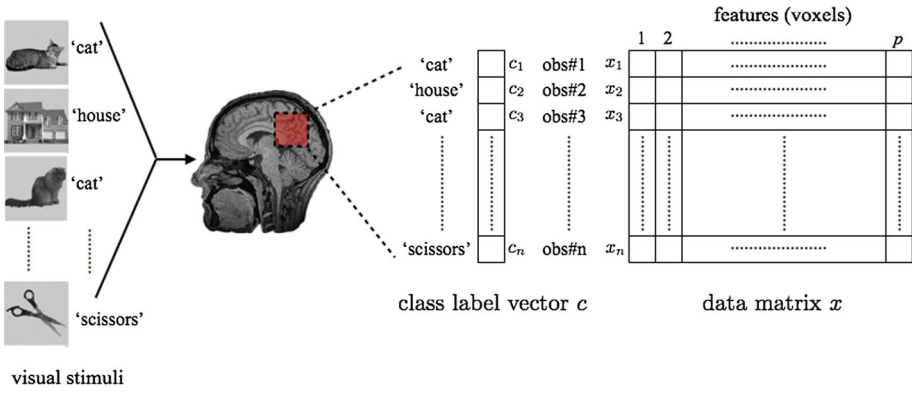

Fig. 1 An illustration of the canonical data matrix of the fMRI data used in the pattern classification system. Each experimental condition is induced by a visual stimulus (image) presented to a human subject in a short period of time, and is eventually transformed into each row i of the $n \times p$ data matrix x, whose class label is denoted by c_i

emerged as a new analysis framework for fMRI. MVPA is often used to perform cognitive state decoding, whereby cognitive representations are classified into discrete categories of stimulus conditions.

MVPA involves several computational steps: feature extraction, feature selection, and pattern classification. For fMRI data, features are conventionally operationalized as voxels. Feature extraction is a procedure to characterize the temporally-evolving response to a stimulus at a voxel, often with a summary value such as a regression coefficient. Feature selection is a procedure to identify and select the subset of voxels to use with the classifier. The voxel selection process is considerably important, especially in cognitive neuroscience where selected voxels implicate brain regions involved in cognitive processes. Pattern classification is a procedure to train a classification algorithm to create a prediction/classification model that best separates the stimulus categories represented in the multidimensional space defined by the selected features (voxels).

Figure 1 illustrates the feature extraction step of fMRI signals from the ventral temporal (VT) cortex as the region of interest (ROI). To characterize the *blood-oxygen-level dependence* (BOLD) response to a given stimulus condition (an indirect measure of the neural response), a general linear model (GLM) is applied, and coefficient parameters "beta" are estimated by fitting a GLM with different predictors for each stimulus block or entity. Unless otherwise noted, in the studies presented here, the predictors were modeled with a boxcar convolved with a canonical hemodynamic response function (HRF) [41]. The HRF has been used to characterize the temporally-evolving BOLD signal change in response to a briefly presented stimulus. In summary, each stimulus can be represented by a 3-dimensional volume matrix, with each entry in the matrix representing a real-valued beta coefficient of a voxel.

In practice, when performing feature selection and classification, it is more convenient to reorganize the volume matrix into a canonical 2-dimensional input data matrix (see Fig. 1). The data matrix is denoted by $\mathbf{x}$ of the dimension $n \times p$, where n is the number of data instances/observations (the total number of presented stimuli); and p is the number of features (voxels) in the ROI. The entry x_{ij} of the data matrix represents the real-valued coefficient parameter beta of the ith data instance at the jth voxel. We denote class label $c_i \in \{1, \ldots, K\}$ (i.e., stimulus category), where K is the total number of stimulus categories. For each data instance i, c_i is known precisely according to the experiment design.

In our previous study [3], a new feature selection based on MI criterion, called maximum informativeness (MaxI), was developed. MI is widely used as a criterion to rank the feature relevance [2,47,50,51], starting from calculating the MI between each feature and the class label vector. MaxI prioritizes the voxels to be selected based on the informativeness of individual features to class labels, assessed by the value of MI (called importance index). The notion of MaxI is to determine the best level of importance index of voxels, rather than the best number for voxels to be selected. To optimize the best level of importance index, a calibration procedure is iteratively carried out with a classification algorithm on the leave-one-run-out cross validation. In that study, SVM, LR, and Gaussian Naïve Bayes (GNB) model were used as classification algorithms.

One of the main drawbacks of MaxI is that it evaluates each individual feature on a univariate basis. That is, it does not consider the non-decomposable information of jointly working features involved in cognitive representations. In the literature, an efficient way to capture the jointly working features is to use *forward/backward selection* algorithm [16], where each feature will be included in the selected set when its combination with the selected features gives the best performance. This process is continued until all the features are included into the selected set or until the storage cost is reached. However, the approach requires $O(p^2)$ which is intractable with a large p value. Recently, an efficient approach based on *submodularity optimization* has been proposed [27,28,31]. Although this approach provides theoretical foundation on the performance, it strictly requires that the objective function of the classification model to be submodular.

3 Logistic regression with regularizations

Logistic regression is widely used as a classifier for classification problems together with feature selection because of its simplicity on performance and implementation. In this section, we present the formulation and the characteristics of linear (binomial and multinomial) logistic regression with penalties of ridge, lasso, L_q-norm, and elastic net, respectively.

3.1 Logistic regression

Let c denote the class variable and $\mathscr{C} = \{1, 2\}$ denote the label set with two categories. A logistic regression model incorporates a linear function of the predictors x into the class-conditional probability. The model is formulated as follows:

$$J(\beta_0, \beta) = \frac{1}{n}\sum_{i=1}^{n}\{I_1(c_i)\log Pr(c_i = 1|x_i) + I_2(c_i)\log Pr(c_i = 2|x_i)\}, \tag{1}$$

where n is the number of data instances, $I_k(c_i)$ is an indicator function returning 1 when $c_i = k$ and 0 otherwise, and $Pr(c_i = 1|x) = \frac{1}{1+e^{-(\beta_0+x^\top\beta)}}$ and $Pr(c_i = 2|x) = 1 - Pr(c_i = 1|x) = \frac{e^{-(\beta_0+x^\top\beta)}}{1+e^{-(\beta_0+x^\top\beta)}}$ are probability functions of both class outcomes. The coefficients (β_0, β) can be computed (trained) by maximizing the objective function $J(\beta_0, \beta)$ with respect to β_0 and β:

$$\max_{(\beta_0,\beta)\in\mathscr{R}^{p+1}} J(\beta_0, \beta). \tag{2}$$

It is noted that the learned coefficients (β_0, β) are not scale-invariant to the input x, so it is often necessary to standardize the input x (e.g., z-score) before solving the maximization problem in Eq. (2).

3.2 Ridge penalty

When there are many correlated features (variables) in the linear model, the coefficients (β_0, β) of these correlated features may cancel each other out, and unbiased estimates may be associated with high variance. Such issue can be alleviated by imposing a size constraint on the coefficients using the L_2-norm squared of β, called *ridge* penalty $\hat{\beta}_{ridge}$. A new maximization model with the size constraint is given by

$$(LR + ridge) \quad \max_{(\beta_0,\beta)\in\mathscr{R}^{p+1}} \quad J(\beta_0, \beta) \tag{3}$$

$$\text{s.t.} \quad \sum_{j=1}^{p} \beta_j^2 \leq t, \tag{4}$$

where t is the bound of the sum of coefficients squared. Note that the β_0 is excluded from the sum.

To facilitate such optimization problem, we apply a Lagrange multiplier method to incorporate the constraint in Eq. (3) into the objective function in Eq. (4). The Lagrangian is then given by

$$\max_{(\beta_0,\beta)\in\mathscr{R}^{p+1}} \left\{ J(\beta_0, \beta) - \lambda \sum_{j=1}^{p} \beta_j^2 \right\}, \tag{5}$$

where $\lambda \geq 0$ is the Lagrange multiplier that represents a complexity parameter and in turn controls the amount of shrinking: the larger value of λ, the greater the amount of shrinkage and hence the smaller the size of coefficients. There is a one-to-one correspondence between the parameters λ in Eq. (5) and t in Eq. (3) [11].

It is worth noting that the maximization model with the ridge penalty does not have the characteristic of feature selection because all the coefficients still remain non-zero even though the ridge penalty shrinks the size of coefficients toward zero. The coefficients of correlated features tend to spread among them and underestimate the true importance of the correlated features. A thresholding strategy can be used to eliminate features with coefficient values near zero. However, this strategy would degrade the performance of the classification model, as the weights of these features underestimate their joint contribution to the model. Theoretically, the lack of explicit thresholding should result in the classification model with the ridge penalty having the same number as features as the model without any regularization. However, in practice, coefficients with numerical values very close to zero (e.g., $\beta < 10^{-14}$) are rounded to zero for numerical robustness, leading to an occasional reduction in the number of features.

3.3 Lasso penalty

Lasso penalty works similar to ridge penalty except that L_1-norm is used in the constraint of the coefficients. The optimization model of logistic regression with a lasso penalty is given by

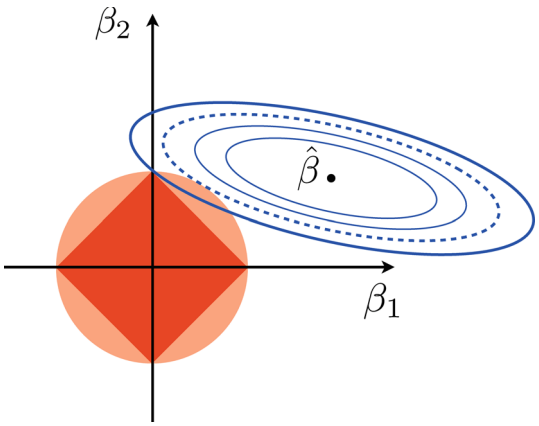

Fig. 2 The geometry of lasso and ridge penalty in the space (β_1, β_2). The constraint region for L_2-norm and L_1-norm is represented by the circular disk (*orange*) and the diamond disk (*red*) respectively. The residual sum of squares has elliptical contour (*blue*) and has its center at the least square solution $\hat{\beta}$. The *dotted line* and the outer-most thick *solid line* each is the *contour* where the ridge solution and the lasso solution occur respectively. The corners of the *diamond* suggest sparse solution, which can happen with greater probability in the lasso regularization than in the ridge regularization. (Color figure online)

$$(LR + Lasso) \quad \max_{(\beta_0,\beta)\in\mathscr{R}^{p+1}} \quad J(\beta_0, \beta) \tag{6}$$

$$\text{s.t.} \quad \sum_{j=1}^{p} |\beta_j| \leq t. \tag{7}$$

An equivalent Lagrangian form is given by

$$\max_{(\beta_0,\beta)\in\mathscr{R}^{p+1}} \left\{ J(\beta_0, \beta) - \lambda \sum_{j=1}^{p} |\beta_j| \right\}. \tag{8}$$

The lasso (L_1-norm) penalty introduces a sparse solution, compared to the ridge (L_2-norm) penalty. Figure 2 displays a geometric example with two parameters β_1 and β_2. The residual sum of squares has elliptical contour and has its center at the *least squares* solution. The point where the elliptical contour first touches the constraint region is the solution to the optimization problem. The constraint region of lasso has corners, and each corner forces that at least one of the features must be zero. It therefore results in a sparse solution. Furthermore, in a higher-dimensional space ($p > 2$), there are more corners, and thus there is a higher chance that the first-touch point ends up at one of the corners. However, it is not true for ridge because the first-touch point can hit anywhere with equal probability. The sparse solution for ridge penalty is rare when p is large.

It is important to note that the lasso penalty tends to pick only a few features (if not only one) from a set of correlated features, yielding a very sparse solution. In practice, the solution might be less robust across validation folds. Therefore, a more generalized form is suggested, called L_q-norm with a penalty $\lambda \sum_{j=1}^{p} |\beta_j|^q$. The value of q in the range of $q \in (1, 2)$ suggests the compromise between the ridge and lasso penalties. However, $|\beta_j|^q$ is differentiable at 0 in such range of q, thus does not share the ability of lasso for assigning some β_j's to zero. In other words, the L_q-norm does not provide a sparse solution when $q \in (1, 2)$.

3.4 Elastic net penalty

Lasso penalty may be too stringent in the selection among a set of strong but correlated features, whereas the ridge regularization tends to shrink the coefficients of correlated features toward each other. The elastic net penalty is introduced to compromise between both penalties. A combined optimization model is given by

$$(LR + Elastic\ net) \quad \max_{(\beta_0,\beta)\in\mathscr{R}^{p+1}} \quad J(\beta_0, \beta) \tag{9}$$

$$\text{s.t.} \quad \sum_{j=1}^{p} \left(\alpha \left|\beta_j\right| + (1-\alpha)\beta_j^2\right) \le t, \tag{10}$$

where $\alpha \in [0, 1]$ is a tradeoff parameter between the lasso and ridge penalties. Its equivalent Lagrangian form is given by

$$\max_{(\beta_0,\beta)\in\mathscr{R}^{p+1}} \left\{ J(\beta_0, \beta) - \lambda \sum_{j=1}^{p} \left(\alpha \left|\beta_j\right| + (1-\alpha)\beta_j^2\right) \right\}. \tag{11}$$

The penalty term is a linear combination of lasso penalty and ridge penalty. The first term (lasso) encourages a sparse solution of β, while the second term (ridge) encourages strongly correlated features to be averaged. Therefore, the elastic net provides both sparsity and selection of correlated features although the α needs to be predetermined.

3.5 Multinomial logistic regression with elastic net penalty

For multi-class classification problems, a maximization model using a penalized maximum multinomial log-likelihood and incorporating the elastic net penalty is given by

$$(MLR + Elastic\ net) \quad \max_{(\beta_{l0},\beta_l)_1^K\in\mathscr{R}^{K(p+1)}} \left\{ J\left((\beta_{l0}, \beta_l)_1^K\right) - \lambda \sum_{l=1}^{K} P_\alpha(\beta_l) \right\}, \tag{12}$$

where

$$J\left((\beta_{l0}, \beta_l)_1^K\right) = \frac{1}{n}\sum_{i=1}^{n} \log Pr(c_i|x_i)$$

and

$$P_\alpha(\beta_l) = \sum_{j=1}^{p} \left(\alpha \left|\beta_{lj}\right| + (1-\alpha)\beta_{lj}^2\right),$$

where c is the class variable taking the value from the label set $\mathscr{C} = \{1, \ldots, K\}$ and $Pr(c = l|x) = \frac{e^{-(\beta_{l0}+x^\top\beta_l)}}{\sum_{l'=1}^{K} e^{-(\beta_{l'0}+x^\top\beta_{l'})}}$ is the probability function of multiple outcomes, employed from [61].

4 Computational framework

In this section, we present a computational framework of feature selection in cognitive neuroscience datasets where the number of data instances is much less than the number of features (i.e., $n \ll p$) due to the collection-time limitation and human-factor practicality.

Optimization of feature selection in prediction with a family of logistic regression classifier is discussed specifically. The coefficients of features represent the contribution of features and the features with non-zeros coefficients becomes features selected in prediction models.

4.1 Optimization of the penalized logistic regression

Because the closed-form analytical solution is not available, we resort to numerical optimization approach for logistic regression. The penalized (binomial and multinomial) logistic regression can be solved differently based on penalty type.

For conventional logistic regression (LR) without regularization and ridge penalized LR (LR + ridge), optimal solutions are obtained using a *trust-region* algorithm. The algorithm can be available in the unconstrained optimization package in MATLAB. In particular, the algorithm takes the derivative of the Lagrangian in Eq. (5) with respect to each β_j for $j \in \{0, 1, \ldots, K\}$, and the local minimum of the Lagrangian is obtained for each λ. Because of the convexity of the objective function, a globally optimal solution can be determined among all local solutions with respect to λ. When $\lambda = 0$, we obtain the solution for LR, while $\lambda > 0$ suggests the solution for LR + ridge.

Lasso penalized logistic regression (LR + lasso) can be regarded as a special case of elastic net penalized logistic regression (LR + elastic net). We employ the algorithms using *cyclical coordinate descent* (CCD) to solve the LR + elastic recently proposed by [12] because it has computational advantages over least angle regression (LAR) algorithm proposed in [10]. They compute a regularization path of λ along. For each value of λ, the CCD creates an outer loop cycling over class l and evaluates a partial quadratic approximation of the multinomial log-likelihood $J((\beta_{l0}, \beta_l)_1^K)$ about the current parameters (β_{l0}, β_l). Consequently, a quadratic approximation is incorporated with the elastic net penalty and becomes penalized weighted least squares problem which can be solved using coordinate descent. In our study, the optimization of LR + elastic net and LR + lasso are implemented using the optimization package *glmnet* provided by [13] while there are considerable numerical techniques used to stabilize CCD.

Recall the tradeoff parameter α in Eq. (11), the solution of LR + lasso can be obtained when $\alpha = 1$. On the other hand, the solution of LR + ridge can be obtained when $\alpha = 0$. However, the computation is not stable in this paradigm, and it has to derive the solution of LR + ridge separately.

4.2 Cross validation and free parameters selection

In this study, we apply a leave-one-run-out cross validation paradigm for optimizing/learning the model parameters, selecting the free parameters, and reporting the prediction accuracy. A dataset $\mathbf{x}$ is divided into F mutually exclusive sections or runs (a shorthand for experiment runs); then a run is marked as a testing dataset and the remaining are divided into a training dataset, and a validation dataset denoted by $\mathbf{x}_{test}$, $\mathbf{x}_{train}$ and $\mathbf{x}_{valid}$, respectively. In a cross validation process, each run (or a subset of data) takes a turn as a testing run. For each run, the training dataset is used to train the model parameters, the free parameters are selected based on the validation dataset, and finally the prediction accuracy is evaluated based on the testing dataset. Usually, a final (prediction) accuracy is reported as the average accuracy across all runs.

It was mentioned in [61] that free parameter α is problem-dependent and fixed while a model refinement can be done by adjusting λ. However, in the current work, we treat both α

Table 1 Summary of datasets used in this paper, including regions of interest (ROI), the number of subjects (no. of subjects) and the number of voxels (no. of voxels) in each ROI

Dataset	(ROI; no. of subjects; no. of voxels)	Description
Haxby	(vtc; 6; 307–675), (wb; 6; 36,292–39,280)	There are 96 observations in total: 12 runs × 8 classes/run × 1 observation/class. The eight classes are "face", "house","cat", "bottle", "scissor", "shoe", "chair" and "scrambled pictures"
Lexical-animtool	(vtc; 7; 1,254–1,423), (wb; 4; 9,786–13,239)	There are 104 observations in total: 4 runs × 2 classes/run × 13 observations/class. The 2 classes are "animals" = {'leopard', 'ant', 'duck', 'fish', 'turtle', etc.} and "tools" = {'paperclip', 'spatula', 'pliers', 'scissors', etc.}
CMU-4class	(unknown; 9; 19,750–21,764)	There are 120 observations in total: 6 runs × 4 classes/run × 5 observations/class. The 4 classes are "animal", "insect","tool" and "vegetable"
CMU-animtool	(unknown; 9; 19,750–21,764)	There are 120 observations in total: 6 runs × 2 classes/run × 10 observations/class. The 2 classes are "animals" = {'animal', 'insect'} and "tools" = {'tool', 'furniture'}

and λ as free parameters to be optimized so that the model is fully data-driven. The model parameters are essentially the function of the free parameters $(\beta_0(\alpha, \lambda),\ \beta(\alpha, \lambda))$ and can be greatly determined by the choice of the free parameters given to the model. Note that the selection criteria of the free parameters are subjective and problem-dependent.

Given a free parameter pair (α, λ), the training dataset $\mathbf{x}_{train}$ is used to learn the model parameters (β_0, β) of a LR classifier. Validation accuracy is then reported from the LR classifier with the learned model parameters on the validation dataset $\mathbf{x}_{valid}$. The free parameter pair (α^*, λ^*) is optimized according to the validation accuracy as follows:

$$(\alpha^*, \lambda^*) = \arg \max_{(\alpha,\lambda)\in\Omega} accuracy(x_{valid}, y_{valid}; \beta_0(\alpha, \lambda),\ \beta(\alpha, \lambda)),$$

where Ω is a set of free parameter candidates defined by the user; $\mathbf{x}_{valid}$ and $\mathbf{y}_{valid}$ denote a data matrix and its corresponding class label vector in the validation dataset, respectively. Consequently, the optimal model parameters can be obtained from $\beta_0^* = \beta_0(\alpha^*, \lambda^*)$ and $\beta^* = \beta(\alpha^*, \lambda^*)$ accordingly. We report the testing accuracy by applying the optimal model parameters (β_0^*, β^*) to the testing dataset $\mathbf{x}_{test}$.

5 Experimental results

In this paper, we evaluate the performances of feature selection via regularization methods on three datasets: (1) Haxby, (2) Lexical and (3) CMU. Summary information for each dataset can be found in Table 1, with more details provided in Sect. 5.2.

5.1 Implementation and evaluation

For each dataset, we applied the linear-kernel logistic regression (LR) with four different types of penalties as described in Sect. 4:

1. *LR + elastic net*: using a linear combination of both lasso and ridge regularization to find the compromise of sparsity and predictivity. That is, $0 < \alpha < 1$ and $\lambda > 0$.

2. *LR+lasso*: using L_1-norm regularization to effectively induce a sparse solution by assigning a large portion of β_j's to be zero. That is, $\alpha = 1$ and $\lambda > 0$.
3. *LR+ridge*: using L_2-norm regularization to shrink the coefficients by imposing a penalty based on their size. The solution is not sparse however, since the coefficients are still non-zero. That is, $\alpha = 0$ and $\lambda > 0$.
4. *LR+none*: this is a control case, meaning that there is no regularization added into the objective function at all. That is, $\lambda = 0$ regardless of α.

Although the free parameter pair (α, λ) are to be selected automatically with respect to the dataset, it is fair to assure the robustness of the solution by imposing the search range of the free parameters $\alpha \in \mathscr{A} = \{0, 0.1, 0.2, \ldots, 1\}$ and $\lambda \in \mathscr{L} = \{0, 0.001, 0.002, \ldots, 1\}$. We shall emphasize that the range of α and λ should entirely represent all possible regularization methods we wish to benchmark. That is, for ridge penalty ($\alpha = 0, \lambda \in \mathscr{L}$); lasso penalty ($\alpha = 1, \lambda \in \mathscr{L}$); elastic net penalty ($0 < \alpha < 1, \lambda \in \mathscr{L}$); and $\lambda = 0$ (regardless of any α) for not penalizing at all.

An additional criterion is used in our experiment as a tie-breaker when the validation accuracy of two free parameter pairs are approximately equal (within some tolerance.) In such a case we prefer the solution that is more sparse (i.e., more interpretable), namely, the solution with fewer non-zero coefficients. Specifically, the solution with bigger α or bigger λ is more preferable.

For lasso and elastic net we use the MATLAB package glmnet from [12,13], and for ridge and none we implemented our own MATLAB code, more details are discussed earlier in Sect. 4.1. We adopt the cross validation paradigm as illustrated in Sect. 4. For all datasets, we organized the training, validation, and testing folds according to experiment run number mentioned in Sect. 4.2. The approach avoids positively biasing results due to the within-run signal structure but also makes the classification problem more challenging due to the presence of difference in signal structure among training, testing and validation folds. Details with respect to each dataset are described in Sect. 5.2 and in Table 1. Performance of each classification model is evaluated by the prediction accuracy in the testing set or *testing accuracy* for short-handed notation. For each subject, the individual testing accuracy is calculated by averaging the testing accuracies across all runs. In each run, the individual testing accuracy is obtained at the optimal free parameter (α^*, λ^*) and the optimal model parameters (β_0^*, β^*) according to Sect. 4.2. Finally, we average the testing prediction accuracy across all subjects in the experiment and report the average testing accuracy. The details of the dataset and the experiment settings are discussed in the following section.

5.2 Data and experiment setting on each dataset

5.2.1 Haxby dataset

The seminal work by [19] demonstrated the utility of pattern classification approaches in fMRI for investigating object category representation in ventral temporal cortex (VTC). The data have since been made publicly available and are widely used to benchmark performance of pattern classification techniques [17–20,38]. In the study, subjects viewed gray-scale images from eight different object categories (face, house, cat, bottle, scissor, shoe, chair, and 'scrambled pictures') as part of a one-back detection task. Exemplars from each category were presented in blocks of 24 s followed by 12 s of rest. Each object category was shown once per fMRI run, with 12 runs of fMRI data acquired per subject. The fMRI data were acquired on a 3T GE scanner and consisted of image volumes of $64 \times 64 \times 40$ voxels acquired every 2.5 s.

Table 2 Summary of the results from Haxby dataset

ROI	Model type	Train accur (%)	Valid accur (%)	Test accur (%)	Average no. of selected voxels
vtc	LR + lasso	90.55	78.13	67.54	57.00
	LR + elastic net	94.73	87.5	73.44	243.25
	LR + ridge	98.79	91.67	80.56	443.00
	LR + none	98.79	60.42	68.75	443.00
wb	LR + lasso	89.04	70.83	58.51	48.50
	LR + elastic net	94.21	72.92	62.50	400.75
	LR + ridge	98.79	43.75	39.93	38,888.00
	LR + none	98.79	21.88	24.31	38,888.00

Standard processing was performed on the fMRI data, including motion correction and linear de-trending. Data were then standardized (z-scored) by subtracting the mean and dividing by the standard deviation of the time series signal at each voxel. To characterize the fMRI response associated with each object category, beta coefficient parameters were estimated by fitting a general linear model (GLM). A different predictor was used to model each object block in each run, producing 96 different parameter estimates (12 parameters for each of the eight object categories) for each subject. We refer interested readers to [40,41] for more details.

The original dataset contains 12 runs per subject, which is divided into 1 run for testing, 2 runs for validating and 9 runs for training denoted by (1:2:9). For this dataset, we focus on selecting features from two different initial regions of interest (ROI):

1. Ventral temporal masks provided by the Haxby group (*vtc*): The masks were defined using combined anatomic functional criteria [19]. The resultant ROI masks were relatively small, ranging from 307 to 675 voxels across subjects.
2. All voxels in the whole brain (*wb*): Across subjects, the number of voxels varied from 36,292 to 39,280.

The dataset information is summarized in Table 1. The experimental results can be found in Table 2.

When using the *vtc* mask, LR + ridge gives the best classification performance, followed by LR + elastic net, LR + none, and LR + lasso. These results illustrate that the ridge penalty performs very well if the feature subset is initially well-constrained. Although the LR + elastic net is about 7 % poorer than LR + ridge, the selected feature subset is roughly half the size of the initial mask. LR without regularization (LR + none) is the baseline model we want to compare with as it demonstrates the characteristic of LR when the size of LR coefficients β's are not regularized. LR + lasso gives the fewest, hence, most sparse, voxels of all the approaches. LR + lasso gives the poorest results here, perhaps because the solution it gives is too sparse, especially given it is performed on a dataset with a small feature dimensionality.

It is also interesting to see that the accuracy gap between training accuracy and validation/testing accuracy is not small despite the regularization is imposed in the classifier. That is because the dataset is partitioned into training, validation and testing set based on the experiment run number. fMRI data from different runs have substantial run-related structured "noise" due to instrument variation and/or differences in the subject factors (e.g., amount of head movement) [5,33]. Therefore, the model learned by the classifier also likely includes

the run-specific information present in the training set, but absent from the validation/testing set, which would contribute to the accuracy gap. Furthermore, the classification model will not embody the run-related information of the testing/validation run. These run-specific effects contribute to the gap between training accuracy and validation/testing accuracy and reflect the reduction in the ability of the classifier to generalize to the class conditions (i.e. the scientifically meaningful information). While partitioning the data based on run reduces classification accuracy in the validation/testing runs, it ensures that accuracy is not positively biased by run effects. The ideal way to partition the data would be to ensure that each dataset contains at least a few examples from each run so that the run-specific information would be captured by the model. We note that the accuracy gap is smaller in the approaches with regularization than ones without regularization, suggesting that regularization mitigates the undesirable effects caused by run-specific information.

In the case that p is very large compared to n ($n \ll p$) like in the *wb* mask, it is more obvious that the sparsity regularization approaches such as elastic net and lasso outperform those without enforcement (i.e., ridge and none). This is because the irrelevant features are subdued better in approaches with sparsity regularization which can be seen as an automatic feature selection step in the classifier.

5.2.2 Lexical dataset

The lexical fMRI data, denoted by *Lexical*, were acquired from seven subjects performing an object naming task. The subjects were scanned on a Siemens 3T TIM Trio Scanner during which they produced names out loud in response to 104 color pictures of 'animals' or man-made manipulable objects (i.e., 'tools') across four runs. The pictures were presented in a rapid event-related design, with each pictured entity randomly repeated four times (using different examples) within a run. Different entities were presented in each run. Imaging data were analyzed using FMRIB's Improved Linear Model [54] using standard preprocessing approaches. Each stimulus entity was modeled separately to obtain individual coefficient estimates of the fMRI response per entity [36].

The dataset is used in a binary classification experiment of "animals" versus "tools", denoted by *Lexical-animtool*. The class "animals" is obtained from combining all the observations whose entities belong to the animal category such as 'leopard', 'ant', 'duck', 'fish', 'turtle', etc. The class "tools" is the combination of 'paperclip', 'spatula', 'pliers', 'scissors', etc. The 4 runs of the data are divided into testing, validation and training in the format of (1:1:2), and there are 13 observations per class per run, therefore 104 observations in total. For testing, only category entities not used during training are evaluated. Consequently, testing accuracy reflects the ability of the classification model to capture generalized category-level and not the entity-level information.

The gap between the training accuracy and validation/testing accuracy is not small due to the nature of this experiment where we expect the classifier to capture the generalized category-level and not the entity-level information. However, there is some entity-level information captured by the classifier. In other words, the accuracy gap is contributed partially by the entity-level information captured by the classifier in each run. It is also worth noting that the accuracy gap is even larger when regularization is not imposed, underscoring the importance of regularization to produce scientifically meaningful results.

Instead of analyzing the whole brain data, we focus our attention on two ROI masks:

1. Features (operationally, voxels) were initially selected based on structural anatomical mask (i.e., posterior occipitotemporal cortex defined using Freesurfer's Desikan parcel-

Table 3 Summary of the results from Lexical dataset

ROI	Model type	Train accur (%)	Valid accur (%)	Test accur (%)	Average no. of selected voxels
vtc	LR + lasso	99.52	85.19	83.45	68.00
	LR + elastic net	100.00	89.45	84.60	258.00
	LR + ridge	100.00	85.32	85.28	2,845.00
	LR	100.00	58.59	59.31	2, 845.00
wb	LR + lasso	100.00	78.84	78.25	82.00
	LR + elastic net	100.00	82.84	81.77	482.50
	LR + ridge	100.00	70.37	71.36	12,474.50
	LR	100.00	52.73	48.33	12,545.50

lation scheme [8]) in the ventral temporal cortex (*vtc*). This ROI mask is available for all seven human subjects.
2. The whole brain's gray-matter mask (*wb*) which aims to reveal the feature that are relevant. This ROI mask was evaluated for only four human subjects.

The dataset information is summarized in Table 1. The experimental results can be found in Table 3. In the *vtc* mask, which is a small preselected ROI mask, LR + ridge is the best, followed by LR + elastic net, LR + lasso and LR + none. LR + elastic net and LR + ridge perform competitively, but LR + elastic net requires fewer features than ridge. In fact, the testing accuracy of the classification model from the lasso regularization is not much lower than that for elastic net and ridge, though the model is much sparser than either of them. LR + none performs the worst, well below all regularized approaches in this experiment.

When considering the case where p is large such as in the *wb* mask, the prediction accuracy of both sparsity regularization approaches, LR + elastic net and LR + lasso, clearly outperforms that of LR + ridge and LR + none. Again, the sparse regularization is more advantageous when p is larger.

5.2.3 CMU dataset

The dataset was collected and used in [34] and is made available to public in the authors' supplemental website [35]. Since the dataset was originally collected by the researchers from Carnegie Mellon University, we shall refer to the dataset as *CMU*.

fMRI data were available from nine participants who viewed 60 different word-picture pairs, each pair is presented six times, with the randomly permuted sequence of stimuli on each presentation. Participants were asked to think about the properties of the item as they were viewing. Data were acquired on a Siemens Allegra 3.0T scanner, with an acquisition matrix was 64×64 with $3.125\,\text{mm} \times 3.125 \times 5\,\text{mm}$ voxels. Data were corrected for motion and slice acquisition timing.

The dataset contains 12 image categories, with each category consisting of five entities each with six observations. The dataset is used in two classification experiments:

1. Binary classification of "animals" versus "tools", denoted by *CMU-animtool*. The class "animals" is obtained from combining the observations from two original categories, 'animal' and 'insect' in the CMU dataset. The class "tools" is the combination of 'tool' and 'furniture'. Thus, there are 120 observations in total.

Table 4 Summary of the results from CMU dataset

ROI	Model type	Train accur (%)	Valid accur (%)	Test accur (%)	Average no. of selected voxels
Animtool	LR + lasso	98.61	73.06	70.28	199.00
	LR + elastic net	100.00	76.11	72.13	629.00
	LR + ridge	100.00	70.37	71.30	20,601.00
	LR + none	100.00	51.76	51.57	20,601.00
4class	LR + lasso	87.11	41.30	40.74	198.00
	LR + elastic net	99.03	47.50	42.78	3,058.00
	LR + ridge	100.00	46.94	42.41	20,601.00
	LR + none	100.00	41.57	39.81	20,601.00

2. Multiclass classification of "animal", "insect","tool" and "vegetable", denoted by *CMU-4class*. The four classes are directly retrieved from the respective categories in the original dataset without modification. Thus, there are 120 observations in total.

Since there are six runs in total, we arrange testing, validation and training set in the format of (1:1:4) in both experiments, yielding 10 and 5 observations/class/run in CMU-animtool and CMU-4class respectively. Since the dataset was preprocessed and the ROI was pre-selected, we adopt the original voxels set provided by [34] without modification. The feature size (number of voxels) of the nine subjects varies from 19,750 to 21,764. The dataset information is summarized in Table 1. The experimental results can be found in Table 4. In both binary classification and multiclass classification experiments, LR + elastic net gives the best testing accuracy, followed by LR + ridge, LR + lasso and LR + none. All regularization approaches report the testing accuracies above chance; however, we note that the accuracies drop significantly from binary to multiclass classification. This may be because the cognitive processes of those four categories are quite similar.

6 Conclusion

In this paper, we presented a sparse optimization framework for regularizing pattern recognition models. The framework was applied to emerging cognitive neuroscience problems based on analyses of neuroimaging data. Logistic regression classifiers with a penalty (regularization) yielded better prediction accuracy performance than ones without regularization. This was especially noticeable when the number of features p was large. The benefits of regularization were observed even when the features were initially well-constrained using anatomic functional criteria. Under these initial conditions, the ridge penalty was sufficient for high classification accuracy and outperformed sparsity-enforcing regularization methods. We note that the LR + ridge is not technically a feature selection method, as the ridge penalty does not eliminate features but rather shrinks their coefficients towards zero.

When the feature size p was bigger (i.e. brain voxels were not restricted using anatomical and/or functional criteria), the advantages of sparsity-enforcement methods became apparent. In such cases, both the LR + lasso and LR + elastic net penalty resulted in classification models with higher prediction accuracy than models obtained using the LR + ridge penalty. These former two regularization methods eliminate irrelevant and noisy features by setting their coefficients to zero. Thus, they embed a feature selection step as part of the training of

the classification model and substantially reduce the number of model features. Of the two methods, the lasso penalty produced the sparsest solution. However, classification models obtained with the lasso penalty had lower prediction accuracy than those obtained with the elastic net penalty. This finding suggests that the lasso regularization method produced feature subsets that were too sparse, and hence as robust in their ability to generalize to the testing data. When the features were well defined initially, LR + elastic net performed competitively with LR + ridge. As elastic net attempts to find the optimal compromise between lasso and ridge regularization, it retains the good prediction accuracy of the ridge penalty, while still providing quite sparse solutions like lasso. Therefore, when taking into account both prediction accuracy and the conciseness in the number of selected features, elastic net appears to be more a desirable regularization approach for fMRI applications.

In the methods described here, optimization of the classifier was achieved by incorporating a penalty term into the objective function. This optimization framework is extensible and allows for incorporation of additional domain specific constraints. In neuroimaging, functional and/or anatomical criteria, such as spatial contiguity and anatomical or functional connectivity, could also be included as constraints embedded in the training process of the classification model. Implementing such approaches could improve scientific interpretability of the results and is an exciting, but non-trivial, future research direction for optimization.

References

1. Amaldi, E., Kann, V.: On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. Theor. Comput. Sci. **209**(1), 237–260 (1998)
2. Chou, C.-A., Kampa, K., Mehta, S.H., Tungaraza, R.F., Chaovalitwongse, W.A., Grabowski, T.J.: Information-theoretic based feature selection for multi-voxel pattern analysis of fMRI data. In: Brain Informatics, pp. 196–208. Springer (2012)
3. Chou, C.-A., Kampa, K., Mehta, S.H., Tungaraza, R.F., Chaovalitwongse, W.A., Grabowski, T.J.: Voxel selection framework in multi-voxel pattern analysis of fMRI signals for prediction of neural response to visual stimuli. IEEE Trans. Med. Imag., under review (2013)
4. Chu, C., Kyun, K.S., Kunle, O.: Map-reduce for machine learning on multicore. Adv. Neural Inf. Process. Syst. **19**, 281 (2007)
5. Coutanche, M.N., Thompson-Schill, S.L.: The advantage of brief fmri acquisition runs for multi-voxel pattern detection across runs. Neuroimage **61**(4), 1113–1119 (2012)
6. Cui, Y., Jin, J., Zhang, S., Luo, S., Tian, Q.: Correlation-based feature selection and regression. In: Qiu, G., Lam, K., Kiya, H., Xue, X.-Y., Kuo, C.-C., Lew, M. (eds.) Advances in Multimedia Information Processing—PCM 2010, vol. **6297** of Lecture Notes in Computer Science, pp. 25–35. Springer, Berlin, Heidelberg (2010) ISBN 978-3-642-15701-1
7. Dean, J., Ghemawat, S.: Mapreduce: simplified data processing on large clusters. Commun. ACM **51**(1), 107–113 (2008)
8. Desikan, R.S., Ségonne, F., Fischl, B., Blacker, D., et al.: An automated labeling system for subdividing the human cerebral cortex on mri scans into gyral based regions of interest. Neuroimage **31**(3), 968–980 (2006)
9. Dy, J.G., Brodley, C.E.: Feature selection for unsupervised learning. J. Mach. Learn. Res. **5**, 845–889 (2004)
10. Efron, B., Hastie, T., Johnstone, I., Tibshirani, R.: Least angle regression. Ann. Stat. **32**(2), 407–499 (2004)
11. Friedman, J., Hastie, T., Tibshirani, R.: The Elements of Statistical Learning: Data Mining, Inference and Prediction. Springer, New York, NY (2009)
12. Friedman, J., Hastie, T., Tibshirani, R.: Regularization paths for generalized linear models via coordinate descent. J. Stat. Soft. **33**(1), 1 (2010a)
13. Friedman, J., Hastie, T., Tibshirani, R.: Lasso (l1) and elastic-net regularized generalized linear models (2010b). http://www-stat.stanford.edu/tibs/glmnet-matlab/
14. Fuchs, J.-J.: On the application of the global matched filter to DOA estimation with uniform circular arrays. IEEE Trans. Signal Process. **49**(4), 702–709 (2001)

15. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. J. Mach. Learn. Res. **3**, 1157–1182 (2003)
16. Guyon, I., Weston, J., Barnhil, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. Mach. Learn. **46**, 389–422 (2002)
17. Hanke, M., Halchenko, Y.O., Sederberg, P.B., Haxby, J.V.: Pymvpa: A python toolbox for multivariate pattern analysis of fMRI data. Neuroinformatics **7**(1), 37–53 (2009)
18. Hanson, S.J., Matsuka, T., Haxby, J.V.: Combinatorial codes in ventral temporal lobe for object recognition: Haxby (2001) revisited: is there a face area? Neuroimage **23**(1), 156–166 (2001)
19. Haxby, J.V., Gobbini, M.I., Ishai, A., Pietrini, P.: Distributed and overlapping representations of faces and objects in ventral temporal cortex. Science **293**(5539), 2425–2430 (2001)
20. Haxby, J.V., Gobbini, M.I., Furey, M.L., Ishai, A., Schouten, J.L., Pietrini, P.: Faces and objects in ventral temporal cortex (fMRI). http://data.pymvpa.org/datasets/haxby2001/ (2010)
21. Haynes, J.-D., Rees, G.: Decoding mental states from brain activity in humans. Neuroscience **7**, 523–534 (2006)
22. He, X., Cai, D., Niyogi, P.: Laplacian score for feature selection. Adv. Neural Inf. Process. Syst. **18**, 507 (2006)
23. Hoerl, A.E., Kennard, R.W.: Ridge regression: biased estimation for nonorthogonal problems. Technometrics **12**(1), 55–67 (1970)
24. Koh, K., Kim, S.-J., Boyd, S.: An interior-point method for large-scale l1-regularized logistic regression. J. Mach. Learn. Res. **8**(8), 1519–1555 (2007)
25. Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artif. Intell. **97**(1), 273–324 (1997)
26. Komarek, P.: Logistic regression for data mining and high-dimensional classification. Robotics Institute, p. 222 (2004)
27. Krause, A., Guestrin, C.: Near-optimal nonmyopic value of information in graphical models. arXiv, preprint arXiv:1207.1394 (2012)
28. Krause, A., Guestrin, C., Gupta, A., Kleinberg, J.: Near-optimal sensor placements: maximizing information while minimizing communication cost. In: Proceedings of the 5th International Conference on Information Processing in Sensor Networks, pp. 2–10. ACM (2006)
29. Le Cun, L.B.Y., Bottou, L.: Large scale online learning. Adv. Neural Inf. Process. Syst. **16**, 217 (2004)
30. Liu, H., Yu, L.: Toward integrating feature selection algorithms for classification and clustering. IEEE Trans. Knowl. Data Eng. **17**(4), 491–502 (2005)
31. Lovász, L.: Submodular functions and convexity. In: Mathematical Programming: The State of the Art, pp. 235–257. Springer (1983)
32. Mangasarian, O.L.: Minimum-support solutions of polyhedral concave programs*. Optimization **45**(1–4), 149–162 (1999)
33. Misaki, M., Kim, Y., Bandettini, P.A., Kriegeskorte, N.: Comparison of multivariate classifiers and response normalizations for pattern-information fMRI. NeuroImage **53**(1), 103–118 (2010)
34. Mitchell, T.M., Shinkareva, S.V., Carlson, A., Chang, K.-M., Malave, V.L., Mason, R.A., Just, M.A.: Predicting human brain activity associated with the meanings of nouns. Science **320**, 1191–1195 (2008)
35. Mitchell, T.M., Shinkareva, S.V., Carlson, A., Chang, K.-M., Malave, V.L., Mason, R.A., Just, M.A.: Supplemental web site in support of the paper: predicting human brain activity associated with the meanings of nouns, September (2009). http://www.cs.cmu.edu/afs/cs/project/theo-73/www/science2008/data.html/
36. Mumford, J.A., Turner, B.O., Ashby, F.G., Poldrack, R.A.: Deconvolving bold activation in event-related designs for multivoxel pattern classification analyses. NeuroImage **59**(3), 2636–2643 (2012)
37. Norman, K.A., Polyn, S.M., Detre, G.J., Haxby, J.V.: Beyond mind-reading: multi-voxel pattern analysis of fMRI data. RENDS Cogn. Sci. **10**(9), 424–430 (2006)
38. O'toole, A.J., Jiang, F., Abdi, H.: Partially distributed representations of objects and faces in ventral temporal cortex. J. Cogn. Neurosci. **17**(4), 580–590 (2005)
39. Peng, H., Long, F., Ding, C.: Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans. Pattern Anal. Mach. Intell. **27**(8), 1226–1238 (2005). ISSN 0162–8828. doi:10.1109/TPAMI.2005.159
40. Pereira, F., Mitchell, T., Botvinick, M.: Machine learning classifiers and fMRI: a tutorial overview. NeuroImage **45**, 199–209 (2009)
41. Poldrack, R.A., Mumford, J.A., Nichols, T.E.: Handbook of Functional MRI Data Analysis. Cambridge University Press, Cambridge (2011)
42. Quinlan, J.R.: C4. 5: Programs for Machine Learning, vol. 1. Morgan Kaufmann, Los Altos (1993)
43. Reunanen, J.: Overfitting in making comparisons between variable selection methods. J. Mach. Learn. Res. **3**, 1371–1382 (2003)
44. Robnik-Šikonja, M., Kononenko, I.: Theoretical and empirical analysis of ReliefF and RReliefF. Mach. Learn. **53**(1–2), 23–69 (2003)

45. Saeys, Y., Inza, I., Larrañaga, P.: A review of feature selection techniques in bioinformatics. Bioinformatics **23**(19), 2507–2517 (2007)
46. Song, L., Smola, A., Gretton, A., Borgwardt, K. M., Bedo, J.: Supervised feature selection via dependence estimation. In: Proceedings of the 24th International Conference on Machine Learning, pp. 823–830. ACM (2007)
47. Thomas, J.A., Cover, T.M.: Elements of Information Theory. Wiley, New York (2006)
48. Tibshirani, R.: Regression shrinkage and selection via the lasso. J. R. Stat. Soc. Ser. B (Methodological), 267–288 (1996)
49. Tusher, V.G., Tibshirani, R., Chu, G.: Significance analysis of microarrays applied to the ionizing radiation response. Proc. Natl. Acad. Sci. **98**(9), 5116–5121 (2001)
50. Verleysen, M., Rossi, F., François, D.: Advances in feature selection with mutual information. In: Biehl, M., Hammer, B., Verleysen, M., Villmann, T. (eds.) Similarity-Based Clustering, pp. 52–69. Springer, Berlin, Heidelberg (2009) ISBN 978-3-642-01804-6
51. Vinh, La The, Thang, N.D., Lee, Y.-K.: An improved maximum relevance and minimum redundancy feature selection algorithm based on normalized mutual information. In: International Symposium on Applications and the Internet, IEEE/IPSJ vol. 0, pp. 395–398 (2010)
52. Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., Vapnik, V.: Feature selection for SVMs. In: Advances in Neural Information Processing Systems, vol. 13, pp. 668–674. MIT Press (2001)
53. Weston, J., Elisseeff, A., Schölkopf, B., Tipping, M.: Use of the zero norm with linear models and kernel methods. J. Mach. Learn. Res. **3**, 1439–1461 (2003)
54. Woolrich, M.W., Ripley, B.D., Brady, M., Smith, S.M.: Temporal autocorrelation in univariate linear modeling of fMRI data. Neuroimage **14**(6), 1370–1386 (2001)
55. Xu, Z., King, I., Jin, R.: Discriminative semi-supervised feature selection via manifold regularization. IEEE Trans. Neural Netw. **21**(7), 1033–1047 (2010)
56. Yu, L., Liu, H.: Feature selection for high-dimensional data: a fast correlation-based filter solution. In: Proceedings of the 20th International Conference on Machine Learning, pp. 856–863 (2003)
57. Zhang, T.: Solving large scale linear prediction problems using stochastic gradient descent algorithms. In: Proceedings of the 21st International Conference on Machine Learning, p. 116. ACM (2004)
58. Zhao, Z., Liu, H.: Semi-supervised feature selection via spectral analysis. In: Proceedings of the 7th SIAM International Conference on Data Mining, Minneapolis, MN, pp. 1151–1158 (2007)
59. Zhao, Z., Morstatter, F., Sharma, S., Alelyani, S., Aneeth, A., Huan, L.: Advancing feature selection research, ASU Feature Selection Repository (2010)
60. Zhou, N., Wang, L.: A modified t-test feature selection method and its application on the hapmap genotype data. Genomics, Proteomics Bioinf. **5**(3), 242–249 (2007)
61. Zou, H., Hastie, T.: Regularization and variable selection via the elastic net. J. R. Stat. Soc. Ser. B (Statistical Methodology) **67**(2), 301–320 (2005)