INFORMS TutORials in Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Computational Optimization and Statistical Methods for Big Data Analytics: Applications in Neuroimaging

Shuai HuangW. Art Chaovalitwongse

To cite this entry: Shuai HuangW. Art Chaovalitwongse. Computational Optimization and Statistical Methods for Big Data Analytics: Applications in Neuroimaging. *In* INFORMS TutORials in Operations Research. Published online: 26 Oct 2015; 71-88.
https://doi.org/10.1287/educ.2015.0135

TUTORIALS IN
OPERATIONS RESEARCH
INFORMS 2015

 | ISBN 978-0-9843378-8-0
http://dx.doi.org/10.1287/educ.2015.0135

Computational Optimization and Statistical Methods for Big Data Analytics: Applications in Neuroimaging

Shuai Huang
Industrial and Systems Engineering, College of Engineering, University of Washington, Seattle, Washington 98195, shuaih@uw.edu

W. Art Chaovalitwongse
Industrial and Systems Engineering, College of Engineering; and Department of Radiology, University of Washington, Seattle, Washington 98195, artchao@uw.edu

Abstract This tutorial focuses on recent advances in computational optimization and statistical methodologies in big data analytics, which is a rapidly emerging interdisciplinary research area. The main focus of this paper lies on classification, regression, and feature selection, which are among the most challenging analytics tasks. We give a thorough discussion on the mathematical and statistical modeling aspects of these problems. We end the paper with an application of big data analytics in brain imaging (neuroimaging) data, which can provide a unique and often complementary characterization of the underlying neurophysiological process that may be useful in clinical diagnoses of brain diseases.

Keywords big data analytics, statistical learning, computational optimization, neuroimaging

1. Introduction

Recent advances in data and computing technologies have reshaped how we control devices, collect data, and interact with systems. Integrated healthcare, intelligent transportation systems, smart grids, and energy management are examples of applications from these advancements. However, such advancements have also brought challenges in terms of data storage and processing. In particular, the amount of data generated by sensing devices and delivered by networking devices is enormous and beyond the ability a single computer system can store and traditional data analysis methods can process. In response to these challenges, the term "big data" has been defined as an umbrella term for any collection of large and complex data sets that are difficult to store, process, analyze, and comprehend using traditional database or data processing tools. The major components in big data include data science, analytics, and visualization. As recognized by President Obama, big data is now a national grand challenge.

The theme of this paper revolves around analytics techniques to make sense of big data through knowledge and information extraction. Data analytics techniques arise in many research areas including *machine learning, data mining, knowledge discovery, artificial intelligence*, and *statistical learning*. This paper will provide tutorials of computational optimization and statistical methods in data analytics that have been widely used to "learn" information directly from data without assuming a predetermined model of the data. The methods for extracting patterns from data, such as data sorting or various hypothesis-driven methods, have been around for centuries. However, it is extremely difficult to transform massive data into valuable knowledge by the traditional means of analysis. This motivates

the development of modern analytics methods, which are designed to discover meaningful representations or structures of data using optimization and statistical methods.

In a broad sense, there are two types of analytics methods: reactive and proactive. Reactive analytics focuses on providing statistics for current and historical data with the hope of providing insights into what happened and why it happened. Methods such as data statistical modeling, trend reporting, and visualization as association and correlation analysis have been commonly used to analyze current and historical data to discover any correlations with them. Proactive analytics, on the other hand, focuses on using a known data set (called the training data set), which includes input data features (aka attributes) and response values (aka target patterns), to build a predictive (decision) model and scale it to make predictions using unseen data (called the test data set). Thus, proactive analytics operates on the assumption that the test data set has a similar structure to or characteristics similar to those of the training data set.

In machine learning and data mining, predictive analytics is known as supervised learning. Several modern optimization and statistical methods have been developed and used routinely in predictive analytics. Among the most widely used techniques for predictive analytics are support vector machines, linear regression/classification, nonlinear regression (generalized linear model, logistic), decision tree, Bayesian learning, nearest neighbor, and neural networks.

While building a predictive model, predictive analytics in big data also deals with a problem associated with superfluous information in the data; this is called feature selection. Feature selection is needed for managing the dimensionality of the data set, which grows with the number of features. More specifically, it reduces the dimensionality of data by selecting only a subset of data features to create a decision model. In addition to dimensionality reduction, feature selection is also closely related to overfitting in regression. Here, overfitting is a common risk of using machine learning models that may fit noise rather than fit the signal. Having a minimal number of features often leads to simpler models, better generalization, and easier interpretation. The concept of parsimony (Occam's razor) is often invoked to bias the search (Blumer et al. [2]): never do with more what can be done with fewer. Feature selection criteria usually involve the minimization of a specific measure of predictive error for model fitting to different subsets of data. In recent years, sparse learning (aka regularization) has gained more popularity as an integrated learning method for simultaneously selecting features and building classification models.

This tutorial will provide a brief overview of computational methods in predictive analytics, including supervised learning, feature selection, and sparse learning approaches. We will also discuss a few real-life medical applications of neuroimaging data analytics. Identifying the early stages of specific common brain diseases is challenging because their early signs and symptoms often overlap with those of other diseases. For example, the clinical features of Parkinson's disease, such as tremor and rigidity, are also seen in benign essential tremor, among others. Because of the nonspecificity of early symptoms and signs, better techniques are needed for differential diagnosis of brain diseases. Predictive analytics in neuroimaging data may be useful in assisting physicians in recognizing abnormal patterns and/or patterns of specific brain diseases and help them decide whether a patient is likely to have a brain disease. Feature selection is also extremely important in neuroimaging data analytics. Because neuroimaging data are typically high dimensional as a result of the number of voxels involved and the small sample size with a limited number of subjects, regularization techniques have been widely used for feature selection to identify brain regions that are disrupted or altered by specific brain diseases.

The rest of this tutorial is organized as follows. Section 2 will focus on predictive analytics that includes many supervised learning methods such as classification and regression. Section 3 will focus on feature selection and sparse learning. Section 4 will present several examples of how these methods can be applied in neuroimaging data analysis. Section 5 will conclude the tutorial with discussions and future research directions.

2. Predictive Analytics (Supervised Learning): Classification and Regression

Supervised learning approaches typically construct a predictive function/model from the training data set, which consists of a set of input–output (aka feature–response) pairs. In the training stage, they attempt to optimize a model for the best mapping between data features and response values. In the testing stage, the predictive model is used to make predictions of new data samples that have not been seen. The prediction outcome is a measure of the model's generalization capability. Good generalization of a supervised learning approach often requires the training data set to be sufficiently large and contain representative samples so that a valid general mapping between data features and response values can be found. There are a number of variants of supervised learning approaches, such as forecasting, predictive modeling, regression, classification, and simulation. Specific structure of the data and the objective of analytics greatly influence the selection of tasks and methods that are to be employed. In this tutorial, we will focus on two of the most widely used supervised learning approaches: classification and regression. Both approaches share the same training and testing framework but differ on the prediction of response values. If the response value is a categorical value, classification methods will be used to identify the mapping that connects the data feature and the corresponding response value (class). If the response value is continuous, regression methods will be used.

We first introduce our notation used throughout this paper. We use boldface lowercase letters to denote vectors and boldface uppercase letters to denote matrices. Variables are denoted by uppercase letters, and other scalars are denoted by lowercase letters. For example, in regression, we denote $\{X_1, \ldots, X_p\}$ as p variables, and we denote Y as the response variable. We denote $\mathbf{X} \subseteq R^{n \times p}$ as the n samples of the p variables, and we denote $\mathbf{y} \subseteq R^n$ as the corresponding responses.

2.1. Support Vector Machine

The support vector machine (SVM) is among the most widely used classification methods. Its widespread acceptance is rooted in its robustness arising from the strong fundamentals of mathematical optimization (Mangasarian [22, 23]) and statistical learning theory (Scholkopf et al. [32], Vapnik [35, 36]). A pioneering work of SVMs in optimization demonstrated how to formulate an SVM as the problem of constructing the best (hyper)plane that separates all data points of one class from those of the other class. This hyperplane is often called the decision boundary. In turn, from a statistical point of view, an SVM can be modeled to incorporate empirical risk minimization and generalization to allow for mislabeled data points (i.e., points falling on the wrong side of the hyperplane). Risk minimization of the decision boundary (aka soft margin) can be achieved by determining the hyperplane that best separates the data points so that misclassification error is minimized, and the distance from the hyperplane to the nearest data point, which is referred to as the margin, is maximized. Support vectors are the data points on the line that are closest to the hyperplane.

The so-called soft-margin SVM can be formulated as follows. Given the training data that consist of a set of points (instances) $\mathbf{x}_i \in R^p$ along with their categories $\mathbf{y}_i = \pm 1$ (*positive* and *negative* classes), where p is the number of features, the hyperplane can be expressed as $\mathbf{x}^T\mathbf{w} + b = 0$, where $\mathbf{w} \in R^p$ is a vector and b is a real scalar (bias). The objective of an SVM is to find the optimal values of $\mathbf{w}$ and b of the separating hyperplane that maximize the margin, expressed by $\|\mathbf{w}\|$, and that minimize classification error. Maximizing the margin is equivalent to minimizing $(\mathbf{w}^T\mathbf{w})/2$. Because real-life training data might not allow for a perfect separating hyperplane, we introduce a slack variable, s_i, to measure the distance required to move data point i to the right side of the hyperplane. For the data points that are already on the correct side of the hyperplane, their slack variables will be zero. We can mathematically represent the expression of the hyperplane separating data

points as $\mathbf{y}_i(\mathbf{w}^T\mathbf{x}+b) \geq 1-s_i$, where $s_i \geq 0$. By introducing a penalty parameter C, the misclassification error in the SVM objective can be expressed by $C\sum_{i=1}^{n} s_i$ for L_1-*norm* or $C\sum_{i=1}^{n} s_i^2$ for L_2-*norm*. Put all together, a quadratic programming formulation of the soft-margin SVM with L_1-*norm* penalty on misclassification is given by

$$\min_{\mathbf{w},\,s,\,b}\left(\frac{\mathbf{w}^T\mathbf{w}}{2}+C\sum_{i=1}^{n} s_i\right) \quad \text{subject to} \quad \mathbf{y}_i(\mathbf{w}^T\mathbf{x}+b)\geq 1-s_i; \quad s_i \geq 0.$$

Lagrange multipliers can be used to formulate the dual quadratic program, which is easier to solve by using a gradient method. After we obtain the optimal solution $(\mathbf{w}^*, b^*)$ of an SVM, a new data point represented by a vector $\mathbf{z}$ can be classified as class($\mathbf{z}$)=sign($\mathbf{w}^{*T}\mathbf{z}+b^*$).

The SVM was originally designed for binary classification. The multiclass SVM has been proposed in the literature and is still an open research area (Hsu and Lin [14]). Most studies reduce the multiclass classification problem into multiple binary classification problems. The earliest multiclass implementation is the *one against all* approach, which constructs k SVM classifiers, where k is the number of classes (Schölkopf et al. [32]). The basic idea is when training the ith SVM classifier, class i is considered as one class while all other classes are treated as another class (Kreßel [19]), paving the way for the application of the binary SVM model. Whereas each SVM classifier will produce an output function value for any new data point, the classification of a new data point is based on the highest output function value among all k models, $\arg\max(\mathbf{w}^{*T}\mathbf{z}+b^*)$. Another alternative method builds *one against one* classifiers by building $k(k-1)/2$ SVM classifiers, where each classifier is trained on data from two classes. Classification of new data points is based on the number of times that all $k(k-1)/2$ classifiers assign the new point to each class. SVMs can be sensitive to outliers; thus new formulations that can relax the training penalty in a controlled manner have recently been developed (Şeref et al. [33]).

2.2. Regression Model

As defined in statistical terms, linear regression models describe the relationship between a *dependent* (*target or response*) *variable*, $\mathbf{y}_i$, and independent variables (data points), $\mathbf{x}_i \in R^p$. The matrix $\mathbf{X} \in R^{n\times p}$ of observations on p data features of n data points is usually called the *design matrix*. In general, a linear regression model is in the form

$$\mathbf{y}_i = \boldsymbol{\beta}_0 + \sum_{j=1}^{p} \boldsymbol{\beta}_j \mathbf{X}_{ij} + \epsilon_i, \quad i = 1,\ldots,n,$$

where $\mathbf{y}_i$ is the response value of the ith data point, $\boldsymbol{\beta}_j$ is the jth coefficient, $\boldsymbol{\beta}_0$ is the constant term, and ϵ_i is the regression error (or noise) of the ith data point. Note that, here, $\boldsymbol{\beta} \in R^p$ is similar to the $\mathbf{w}$ used in an SVM and $\boldsymbol{\beta}_0$ is similar to b. Here, we follow the notional convenience we used for SVM formulations. The objective of regression model is to minimize the mean squared error, $\sum_{i=1}^{n} \epsilon_i^2$ (aka least squares).

Logistic regression (LR) is among the most widely studied nonlinear regression models. It has been widely used as a classifier because it is efficient, accurate, and easy to interpret. Let $\mathbf{y}_i = \pm 1$ denote the class (response) variable. A logistic regression model incorporates a linear function of the data point $\mathbf{x}_i$ into the class-conditional probability. The model is formulated as

$$J(\boldsymbol{\beta},\boldsymbol{\beta}_0) = \frac{1}{n}\sum_{i=1}^{n}\{I_{+1}(\mathbf{y}_i)\log \Pr(\mathbf{y}_i = 1 \mid \mathbf{x}_i) + I_{-1}(\mathbf{y}_i)\log \Pr(\mathbf{y}_i = -1 \mid \mathbf{x}_i)\},$$

where n is the number of data points, $I_k(\mathbf{y}_i)$ is an indicator function returning 1 when $\mathbf{y}_i = k$ and 0 otherwise, and $\Pr(\mathbf{y}_i = 1 \mid x_i) = 1/(1+e^{-(\boldsymbol{\beta}_0+\mathbf{x}^T\boldsymbol{\beta})})$ and $\Pr(\mathbf{y}_i = -1 \mid \mathbf{x}_i) = 1-$

$\Pr(\mathbf{y}_i = 1 \mid \mathbf{x}_i)$ are probability functions of both response values. The coefficients $(\boldsymbol{\beta}, \boldsymbol{\beta}_0)$ can be computed (trained) by maximizing the objective function $J(\boldsymbol{\beta}, \boldsymbol{\beta}_0)$. It should be noted that the coefficients $(\boldsymbol{\beta}, \boldsymbol{\beta}_0)$ are not scale-invariant to the data feature $\mathbf{x}_i$; therefore, $\mathbf{x}_i$ should be standardized (e.g., by computing the z-score) before solving the maximization problem. LR can be easily extended to solve a multiclass classification problem by using a penalized maximum multinomial log-likelihood, which is often referred as multinomial LR.

2.3. Classification Tree and Regression Tree

Based on features of data points, classification trees and regression trees are hierarchical decision trees that predict responses to data. If the responses are nominal, the decision tree is called a classification tree. If the responses are continuous, the tree is called a regression tree. In a decision tree, nodes represent the condition of features, and branches represent conjunctions of features that lead to those predictions. Given a set of training data, a decision tree can be induced in the form of a sequence of rules. Once a tree is formed, it can be used to easily predict responses to unseen data based on their features by following the decisions starting at the root node down to a leaf node, which contains the response. Each step in a prediction involves checking the value of one feature (variable). The key challenge for building a good decision tree is to choose features for branching. The objective is to reduce impurity or uncertainty in data as much as possible. A subset of data is pure if all data points belong to the same class or have similar response values. In general, decision tree algorithms are recursive. There are several popular algorithms that have been developed in the literature. Arguably the most well-known algorithm is C4.5, which builds decision trees from a set of training data by the measure of uncertainty associated with a response variable (Quinlan [30]). As any feature of data can be used to split the data set into smaller subsets, C4.5 examines the relative entropy (aka Shannon entropy) for each feature, and the feature with the highest normalized information gain is used for branching.

Both classification trees and regression trees provide an effective way for discovering meaningful patterns and prediction rules for a data set. However, it might be impractical or impossible to construct a decision tree using all of the data instances for a very large data set. In this perspective, optimization techniques could play a valuable role in building accurate and concise decision trees from large data sets. A method for selecting the best decision tree using the minimum description length (MDL) principle is proposed (Quinlan [30]). A decision tree is first built using the standard C4.5 algorithm. Subsequently, the optimal subtree structure is selected by the MDL, which selects a solution that minimizes the total number of bits needed to encode the tree and the description of the data given by the tree. This approach offers a way to prune the constructed tree and also avoids the problem of overfitting.

2.4. Bayesian Learning

Bayesian decision theory is one of the most widely used statistical approaches for solving problems in data analytics. The basic idea of Bayesian learning is based on the estimation of probability distributions, which can be described by the well-known naïve Bayes probability model given by $P(Y \mid X) = P(X \mid Y)P(Y)/P(X)$, where $P(Y)$ is called the prior probability of response variable (class) Y, which is the probability (or proportion) that response class Y is in the training data; $P(X \mid Y)$ is called the likelihood probability, which is the probability of observation data X given response class Y. The prior probability of observation data is denoted as $P(X)$, which is the probability of witnessing data X under all possible response classes, and $P(Y \mid X)$ is the posterior probability, which is the probability of response class Y given that the data X occur. Note that the naïve Bayes classifier is designed with the naïve assumption that the features are independent of one another within each class, but it appears to work well in practice. In the training step, the naïve Bayes classifier uses the training

data to estimate the parameters of a probability distribution. In the prediction step, for any unseen test data point, the classifier computes the posterior probability of that data point belonging to each class and uses the maximum a posteriori decision rule, which simply picks the most probable response classes (the largest posterior probability). The main challenge in Bayesian learning is estimating the probability distribution of data features in each class. Normal (Gaussian) distribution is the most commonly used distribution since it only requires the mean and the standard derivation of each feature of the training data in each class. If data features are known to be skewed or have multiple peaks or modes, a kernel distribution may be suitable because it does not require the normal distribution assumption.

2.5. Nearest Neighbor

The key concept of the nearest neighbor (NN) algorithm, often referred as the k-nearest neighbor (KNN), is based on the assumption that data instances whose features are similar should also have a similar response values. In general, the KNN classifies a test instance based on the majority category of k training instances that are most similar to the test instance. Thus, the predicted response value or class of the test instance is assigned to the most common class or a weighted average among its k nearest neighbors. Any ties can be broken arbitrarily. The performance of KNN is highly dependent on the choice of the k value. If k is too small, the prediction result might be very sensitive to noise; if k is too large, the model tends to be inaccurate since too many points from other classes might be included.

An important aspect of KNN is how to predict the response values or classes once the k nearest neighbors are identified. As mentioned above, the simplest method is to take a majority vote; however, problems may arise when the nearest neighbors are distributed widely or some classes dominate the prediction with more frequent samples. To counter this situation, the prediction step can be enhanced by assigning a weight to each neighbor according to its distance to the test instance. The weight of a neighbor is often defined as the reciprocal of its distance. The new instance is then classified as the class with the highest sum of weights. In addition, the distance measure is very essential to the KNN. There are several similarity measures that have been commonly used to compute the distance between two data instances: Euclidean distance, chi-square distance, dynamic time warping, t-index, among others. Using a measure that is appropriate for the data features is crucial in obtaining a good prediction outcome.

Compared with other supervised learning methods, KNN does not require a sophisticated modeling process, and thus it is very easy to understand and implement. However, the main challenge of KNN is that its classification accuracy can be severely degraded by the presence of irrelevant or noisy features. A good selection of features could reduce the dimensionality of data, scale down the computation time, and, most importantly, improve the classification accuracy. In the KNN literature, much effort has been put into selecting appropriate sets of features; however, most of the methods developed are based on a greedy search. Optimization-based methods for feature selection of KNN were proposed in Chaovalitwongse et al. ([4, 5]).

3. Regularization: Feature Selection and Sparse Learning

Feature selection has been a critical prerequisite for the application of many data mining and machine learning algorithms (Liu and Motoda [20]). It is usually unrealistic to use all the features as predictors in a prediction model such as support vector machine and other kernel methods (Hastie et al. [12]) because of both computational concerns and the risk of overfitting. It has been long known that, in classic regression theory, including more features (even the features that are entirely irrelevant with the response variable) in a regression model will also decrease the prediction error on the training data (Montgomery et al. [25]). However, since the ultimate goal of a regression model is to pursue a robust and accurate

performance on unseen testing data, including irrelevant features will not help with the prediction; rather, such features may add noise into the prediction model and result in an inaccurate prediction. This phenomenon is an example of overfitting. In addition to regression models, other classification methods such as KNN can benefit from optimization of the feature selection (Chaovalitwongse et al. [4, 5]). Therefore, feature selection is one effective approach that can reduce the dimensionality of the data set and lead to predictive models that are built on a small number of features.

3.1. Feature Selection

The essential idea of feature selection is to encourage parsimony in statistical modeling, which can actually be traced back to some ancient scientific principles, such as Occam's razor (Blumer et al. [2]). The parsimony of statistical models implies a preference over the models with a smaller number of free parameters among all the models that can sufficiently capture the complexity and uncertainty of the data sets. Some direct benefits of such a parsimonious statistical model include enhanced interpretability and stability of the models. There are some survey papers and books that have reviewed the existing feature selection methods developed for different data mining objectives or different data sets (e.g., Guyon and Elisseeff [11], Liu and Motoda [20]). Here, we will use the regression model as an example to illustrate how different feature selection methods work and how they are related.

The feature selection for a regression model can be described as follows:

$$\hat{\boldsymbol{\beta}} = \arg\min\{\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_0\}.$$

Here, $\|\boldsymbol{\beta}\|_0$ is the number of nonzero elements in $\boldsymbol{\beta}$, and λ is the penalty parameter that controls the shrinkage effect, i.e., a larger λ imposes more of a penalty on the magnitudes of $\hat{\boldsymbol{\beta}}$.

This presents a very challenging combinatorial optimization problem. Roughly speaking, to solve this problem, there are two major schools of thought that have been found effective in a range of applications. One is to use heuristics methods. Existing methods (e.g., Guyon and Elisseeff [11]) that fall into this category include forward selection, backward selection, and stepwise selection. There, algorithms usually have two critical components: one is an objective function (called the criterion), which the method uses to evaluate the goodness of fit of the selected features. For the regression example, the least square $\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2$ can be used as the criterion. The other component is the heuristic method that aims to identify a good set of features in a sequential manner, which adds or removes features from a candidate subset while evaluating the criterion. The forward selection methods sequentially add features to an originally empty candidate set until the addition of further features does not decrease the least square error. The backward selection methods sequentially remove features from the full list of features until the removal of further features increases the least square error. The stepwise selection is a combination of both methods, which could add features and remove features along the feature selection process.

3.2. Sparse Learning

Besides the heuristics methods, another major school of thought is the sparse learning framework that has attracted much attention recently. The basic idea of sparse learning is to impose regularization on the model complexity, which is usually characterized by the number of unknown parameters. Regularization is a technique used to control the model complexity by forcing many of the unknown parameters to be zero. A classic example of sparse learning is the ridge regression (Hoerl and Kennard [13]), which employs an L2-norm regularized least square formulation to encourage sparsity of the regression parameters, as shown below:

$$\hat{\boldsymbol{\beta}} = \arg\min\{\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2^2\}.$$

Compared with the original formulation for feature selection for regression models, the ridge regression replaces $\|\boldsymbol{\beta}\|_0$ with $\|\boldsymbol{\beta}\|_2^2$. The benefit of this replacement is to convert the challenging combinatorial optimization problem into an easy-to-solve problem, which actually has a closed-form solution. Ridge regression is commonly known as being capable of effectively shrinking many unreliable or redundant regression parameters in $\hat{\beta}$ toward zero, thus achieving more stability than ordinary least square regression models. However, since the objective function of ridge regression is a smooth function without any singularity in the parameter space for any $\boldsymbol{\beta}_i = 0$, there is actually no parameter in $\hat{\boldsymbol{\beta}}$ that will be exactly zero. This implies that ridge regression is not a truly sparse model, since all the variables are always kept in the regression model, no matter how large the penalty parameter λ used. On the other hand, in many applications, such as cancer research, there is a hypothesis that only a few genes or proteins are correlated with the disease processes, although we can obtain the gene expressions of thousands of genes simultaneously. Therefore, ridge regression is less effective in these applications, and statistical models that can achieve exact sparsity in $\hat{\boldsymbol{\beta}}$ are required.

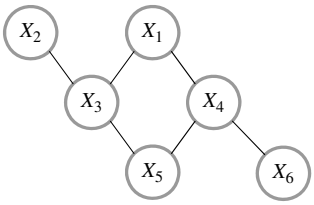
Least absolute shrinkage and selection operator (LASSO) is an approach that can effectively achieve exact sparsity in $\hat{\beta}$ (Tibshirani [34]). The formulation of LASSO is similar to that of ridge regression, except that the L_1-norm is used to replace the L_2-norm:

$$\hat{\boldsymbol{\beta}} = \arg\min\{\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1\},$$

where $\|\boldsymbol{\beta}\|_1$ is a sum of the absolute values of the elements in $\boldsymbol{\beta}$. As the L_1-norm introduces singularity into the objective function for any $\boldsymbol{\beta}_i = 0$, LASSO can set many unreliable or redundant parameters to be exactly zero, thus achieving model selection. The extra cost of LASSO is an efficient computational algorithm, since LASSO has no closed-form solution. By drawing on recent developments on optimization theories, especially on convex optimization, a number of efficient algorithms have been developed to solve LASSO. Following this line, a number of novel norms have been developed to address various kinds of complex data sets, with the goals of making use of some "structural information" about the structure of these complex data sets, such as group LASSO and fused LASSO. The idea of LASSO has also been extended to other statistical models, such as sparse principal component analysis (Zou et al. [40]), sparse linear discriminant analysis (Huang et al. [16], Qiao et al. [29]), sparse graphical models (Friedman et al. [10]; Huang et al. [15, 17]), and many other extensions (Liu et al. [21], Schmidt et al. [31]).

Note that although ridge regression and LASSO use L_2-norm (i.e., $\|\boldsymbol{\beta}\|_2^2$) and L_1-norm (i.e., $\|\boldsymbol{\beta}\|_1$), respectively, their commonality is that they use a convex norm to approximate the nonconvex norm $\|\boldsymbol{\beta}\|_0$. By doing this, the computational challenge can be alleviated and convex optimization methods can be used for solving the sparse learning methods efficiently—not to mention that the ridge regression actually admits a closed-form solution. There have been some more recent developments that produce many new norms, such as the one used in group LASSO (Yuan and Lin [38]) to select a group of features simultaneously rather than selecting features individually. The L_2/L_1-norm has also been developed (Argyriou et al. [1]) for feature selection jointly on multiple related data sets; this is also known as multitask feature selection (Argyriou et al. [1]). Note that both L_1- and L_2/L_1-norms are convex regularizations. On the other hand, there is increasing evidence that these convex norms tend to produce too severely shrunken parameter estimates. Therefore, these convex regularizations could lead to misidentification of the weak-effect features, which may collectively present strong prediction capability. Also, convex regularizations tend to select many irrelevant variables to compensate for the overly severe shrinkage in the parameters of the relevant variables. Considering these limitations of convex regularizations, nonconvex norms have been developed in the literature as well (Candès et al. [3], Huang et al. [16], Mazumder et al. [24], Zhang [39]).

FIGURE 1. A graphical visualization of (a) the IC model and (b) the corresponding IC matrix.



$$\Theta = \begin{bmatrix} 0 & \theta_{13} & \theta_{14} & 0 & 0 \\ & \theta_{23} & 0 & 0 & 0 \\ & & 0 & \theta_{35} & 0 \\ & & & \theta_{45} & \theta_{46} \\ & & & & 0 \end{bmatrix}$$

(a) (b)

3.3. Sparse Inverse Covariance Matrix Estimation

Sparse learning methods have been developed for feature selection, which is a critical tool for reducing the dimensionality of many classification and regression problems. Sparse learning has also been used for more general statistical modeling. The sparse inverse covariance matrix estimation (SICE) is a recent development for studying the interactions among a large number of random variables (Friedman et al. [10], Huang et al. [15]). Note that in feature selection formulation, there are p random variables denoted as $\{X_1, \ldots, X_p\}$, and the feature selection methods aim to identify a small subset of variables from this set that can best predict a response variable Y. Here, SICE concerns the relationships between $\{X_1, \ldots, X_p\}$ where no response variable Y is concerned. SICE assumes that $\{X_1, \ldots, X_p\}$ follow a multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. Let $\boldsymbol{\Theta} = \boldsymbol{\Sigma}^{-1}$ be the inverse covariance matrix. Rather than modeling the covariance matrix $\boldsymbol{\Sigma}$, SICE techniques are used to uncover the intrinsic interactions encoded in $\boldsymbol{\Theta}$ among the p variables (Friedman et al. [10], Huang et al. [15]). This is because the inverse covariance matrix has a clear interpretation that the off-diagonal elements correspond to partial correlations, i.e., the correlation between each pair of variables given all other variables. That means two variables are conditionally independent conditioning on all other variables if their partial correlation is zero. The inverse covariance matrix can be visualized as a network, as illustrated in Figure 1, where an arc is placed between two variables only if their inverse covariance is nonzero.

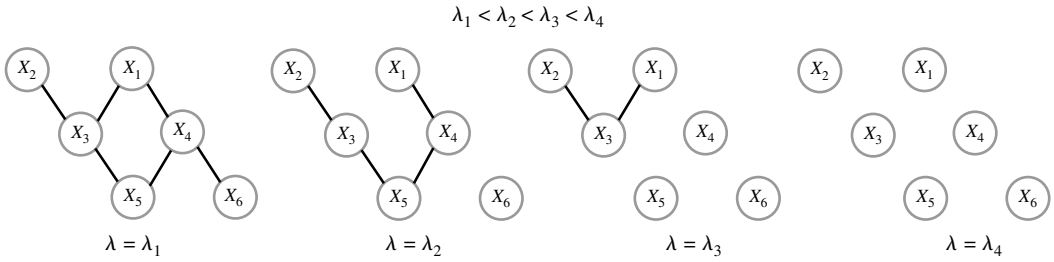
Formulation of SICE: The ability to learn the inverse covariance from data presents a challenge. This is particularly true when the number of variables is large and the sample size is relatively small compared with the number of variables, commonly called "large-p-small-n" problems. To mitigate this challenge, SICE adopts the sparsity regularization that forces many elements in the inverse covariance matrix to be zero. The working logic for SICE analysis is as follows. First, an empirical covariance matrix is computed based on the observed relationships among all variables within a data set. Next, conditioned on the empirical covariance matrix, a likelihood function of the inverse covariance matrix can be derived. Based on the assumption of sparsity (i.e., a large number of entries in the inverse covariance matrix could be set to zero without losing much information), a set of penalty parameters is applied, forcing a weak or redundant inverse covariance to be zero. The higher the penalty, the more sparse the resulting inverse covariance matrix is.

More specifically, we can formulate the SICE into an optimization problem; i.e.,

$$\hat{\boldsymbol{\Theta}} = \arg\max_{\boldsymbol{\Theta} \succ \mathbf{0}} \left\{ \log(\det(\Theta)) - \operatorname{tr}(\mathbf{S}\boldsymbol{\Theta}) - \lambda \|\operatorname{vec}(\boldsymbol{\Theta})\|_1 \right\},$$

where $\mathbf{S}$ is the sample covariance matrix; $\det(\cdot)$, $\operatorname{tr}(\cdot)$, and $\|\operatorname{vec}(\cdot)\|_1$ denote the determinant, trace, and sum of the absolute values of all elements of a matrix, respectively. The $\log(\det(\boldsymbol{\Theta})) - \operatorname{tr}(\mathbf{S}\boldsymbol{\Theta})$ part is the log-likelihood, whereas the $\|\operatorname{vec}(\boldsymbol{\Theta})\|_1$ part represents the "sparsity" of the inverse covariance matrix $\boldsymbol{\Theta}$. This formulation aims to achieve a trade-off between the likelihood fit of the inverse covariance estimate and the sparsity. The trade-off

FIGURE 2. An illustration of the monotone property of SICE.



is controlled by λ, which we call the regularization parameter; a larger λ would result in a more sparse estimate for $\mathbf{\Theta}$. The formulation follows the same line of the L_1-norm regularization, such as LASSO (Tibshirani [34]). To solve it, a block coordinate descent (BCD) algorithm can be found in Huang et al. ([15]). The basic idea of the BCD algorithm is to update each column (or row) of $\mathbf{\Theta}_i$ iteratively while fixing all other columns (or rows) until convergence.

Next, we show that with λ going from small to large, the resulting brain connectivity models have a monotone property. Before introducing the monotone property, the following definitions are needed.

Definition: In the graphical representation of the inverse covariance, if node X_i is connected to X_j by an arc, then X_i is called a *neighbor* of X_j. If X_i is connected to X_k though some chain of arcs, then X_i is called a *connectivity component* of X_k. An illustration is given in Figure 2.

Intuitively, being neighbors means that two nodes (i.e., brain regions) are directly connected, whereas being connectivity components means that two brain regions are indirectly connected; i.e., the connection is mediated through other regions. In other words, not being connectivity components (i.e., two nodes completely separated in the graph) means that the two corresponding brain regions are completely independent of each other. Connectivity components have the following monotone property.

Monotone property of the SICE: Let $\mathbf{C}_k(\lambda_1)$ and $\mathbf{C}_k(\lambda_2)$ be the sets of all the connectivity components of X_k with $\lambda = \lambda_1$ and $\lambda = \lambda_2$, respectively. If $\lambda_1 < \lambda_2$, then $\mathbf{C}_k(\lambda_1) \supseteq \mathbf{C}_k(\lambda_2)$.

Proof of the monotone property can be found in Huang et al. ([15]). This monotone property can be used to identify how strongly connected each node (brain region) X_k to its connectivity components. For example, assuming that $\mathbf{C}_k(\lambda_1) = \{X_i, X_j\}$ and $\mathbf{C}_k(\lambda_2) = \{X_i\}$, this means that X_i is more strongly connected to X_k than X_j. Thus, by changing λ from small to large, we can obtain an order for the strength of connection between pairs of brain regions.

4. Applications in Brain Imaging Analysis

The human brain is among the most complex systems known to man. For centuries, neuroscientists have been seeking to understand brain function through detailed analysis of neuronal excitability, synaptic transmission, and connectivity. Modern neuroimaging techniques are now powerful tools to measure and characterize the function and structure of the brain. Commonly used neuroimaging modalities include magnetic resonance imaging (MRI), positron emission tomography (PET), diffusion tensor imaging, and functional MRI (Mulert and Lemieux [26]). MRI is a typical structural neuroimaging technique, which allows for visualization of brain anatomy. PET is a typical functional neuroimaging technique, which measures brain activities. Neuroimaging has been found to be a powerful method with enormous implications on both scientific discovery and clinical applications, such as understanding how the brain structure supports the cognitive functions (Chou et al. [6], Kampa et al. [18]), how to identify brain regions disrupted by neurodegenerative diseases

(Weaver et al. [37]), how the disease processes such as Alzheimer's disease disrupt the functions, how to monitor the disease progression, and how to evaluate the treatment effect as a more sensitive and reliable index than conventional subjective cognitive measurements, for example. Given such great promises, there have been a number of high-quality open scientific data sets that provide big neuroimaging data to worldwide researchers, such as the Alzheimer Disease Neuroimaging Initiative, Human Connectome Project, 1000 Functional Connectomes Project, and Allen Institute for Brain Science. These ongoing and emerging projects are expected to generate a deluge of data that capture the brain activities at different levels of organization. There is thus a compelling need to develop the next generation of computational methods for data mining and knowledge discovery of neuroimaging data. These methods will allow one to make sense of this raw yet information-rich data and to understand how neurological activity encodes information. In this section, we aim to demonstrate how the analytics methods discussed in this tutorial can be used to analyze the neuroimaging data, facilitate scientific knowledge discovery, and support clinical decision making.

4.1. Alzheimer's Disease Diagnosis

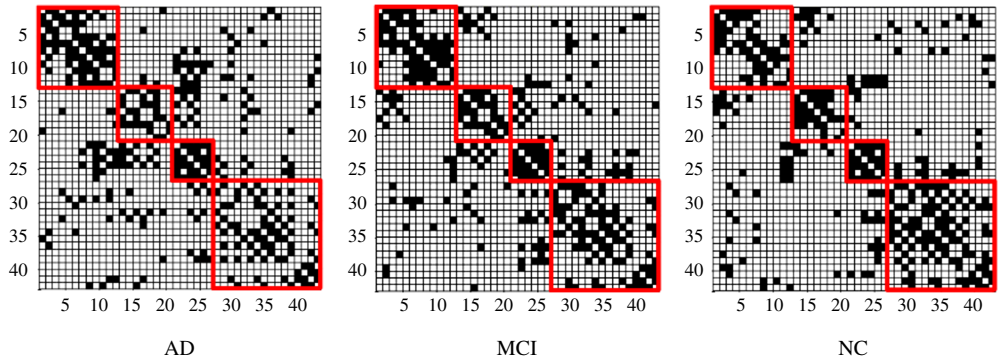
4.1.1. Sparse Learning from the Brain Connectivity Network. We applied SICE on PET images of 49 subjects with Alzheimer's disease (AD), 116 with mild cognitive impairment (MCI), and 67 with normal aging (NC); the images were downloaded from the Alzheimer's Disease Neuroimaging Initiative (ADNI) website. Forty-two brain regions (42 variables) were selected for brain connectivity modeling, because they are considered to be potentially related to AD. These regions distribute in the frontal, parietal, occipital, and temporal lobes. Using the SICE algorithm, one connectivity model can be learned for AD, one for MCI, and one for NC, for a given λ.

We compared the connectivity networks of the three groups. For example, disease knowledge can be extracted by seeing how within-lobe and between-lobe connectivity is different across AD, MCI, and NC. To achieve this, we first learned one connectivity model for AD, one for MCI, and one for NC. We adjusted the λ in the learning of each model such that the three models, corresponding to AD, MCI, and NC, respectively, will have the same total number of arcs. This is to "normalize" the models, so that the comparison will be more focused on how the arcs distribute differently across different models. By selecting different values for the total number of arcs, we can obtain models representing the brain connectivity at different levels of strength. Specifically, given a small value for the total number of arcs, only strong arcs will show up in the resulting connectivity model, so the model is a model of strong brain connectivity; when increasing the total number of arcs, mild arcs will also show up in the resulting connectivity model, so the model is a model of mild and strong brain connectivity.

For example, Figure 3 shows the connectivity models for AD, MCI, and NC with the total number of arcs equal to 180. In this paper, we use a "matrix" representation for the SICE of a connectivity model. In the matrix, each row represents one node and each column also represents one node. The matrix contains black and white cells: a black cell at the ith row, jth column of the matrix represents the existence of an arc between nodes X_i and X_j in the SICE-based connectivity model, whereas a white cell represents the absence of an arc. According to this definition, the total number of black cells in the matrix is equal to twice the total number of arcs in the SICE-based connectivity model. Moreover, on each matrix, four red cubes are used to highlight the brain regions in each of the four lobes; that is, from top left to bottom right, the red cubes highlight the frontal, parietal, occipital, and temporal lobes. The black cells inside each red cube reflect within-lobe connectivity, whereas the black cells outside the cubes reflect between-lobe connectivity.

Interesting patterns can be observed from the learned networks. For example, the temporal lobe of the AD brain has significantly less connectivity than that of the NC. In other

FIGURE 3. SICE-based brain connectivity models (total number of arcs equal to 180).

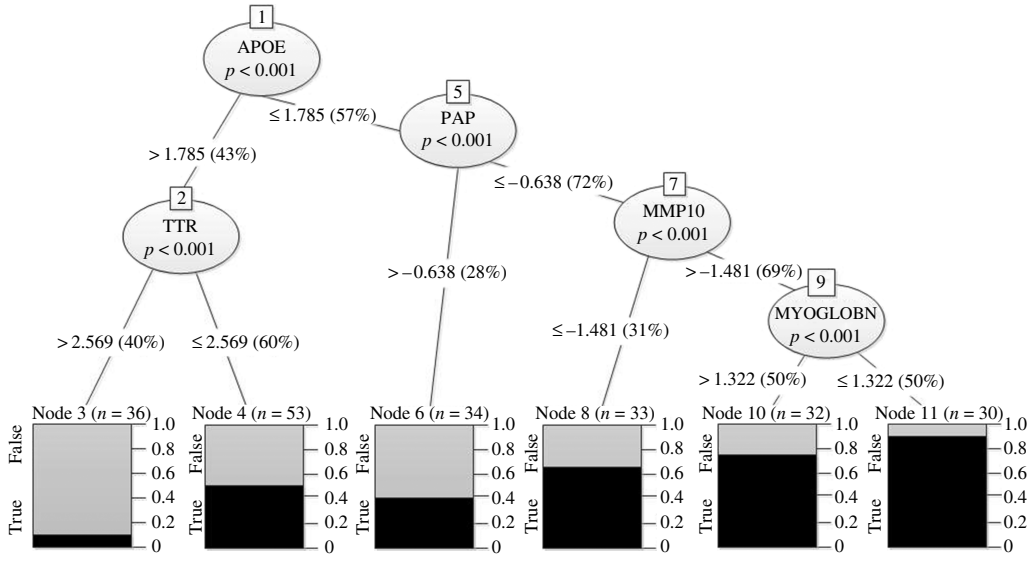


words, even the connectivity between some strongly connected brain regions in the temporal lobe may be disrupted by AD. The decrease in connectivity in the temporal lobe of the AD brain—especially between the hippocampus and other regions—has been extensively reported in the literature. Furthermore, the temporal lobe of the MCI brain does not show a significant decrease in connectivity compared with that of the NC. This may be because MCI does not disrupt the temporal lobe as badly as does AD. The frontal lobe of AD has significantly more connectivity than that of NC, which is true across different strength levels of the connectivity. This has been interpreted as compensatory reallocation or recruitment of cognitive resources. Because the regions in the frontal lobe are typically affected later in the course of AD (our data are early AD), the increased connectivity in the frontal lobe may help preserve some cognitive functions in AD patients. Furthermore, the frontal lobe of MCI does not show a significant increase in connectivity compared with that of NC. This indicates that the compensatory effect in MCI brains may not be as strong as that in AD brains. More discussions of the clinical relevance of the results of applying SICE can be found in Huang et al. ([15]).

4.1.2. Decision Tree Model. Anti-amyloid preventative treatments hold great promise of preventing AD. Amyloid is a human protein and abnormal amyloid-β deposition has been widely regarded as the initial event in a cascade of pathological processes leading to synaptic dysfunction and neuronal death, followed by the development of cognitive impairment and eventually dementia. The success of the clinical trials of these preventative treatments requires appropriately selected participants who are positive for Aβ pathology. However, the identification of these suitable individuals with elevated brain amyloid burden poses a great challenge in terms of feasibility and cost. For instance, the molecular imaging tracers that bind to amyloid, such as Pittsburgh Compound B (PiB), are economically challenging for routine use given the current cost and restrictions on reimbursement. Similarly, the clinical use of other useful biomarkers such as beta-amyloid 1–42 (Aβ_{1-42}) and phosphorylated tau in cerebral spinal fluid (CSF) is also limited, since lumbar puncture carries risks and is met with resistance in elderly subjects. Furthermore, it is unlikely to be used in primary healthcare centers to routinely screen large numbers of participants. Given the cost and limited availability of these brain amyloid measurement techniques, they are not reasonable first-line approaches for screening participants at risk of having elevated brain amyloid burden.

On the other hand, it has been reported that some cost-effective measurements, such as blood-based biomarkers, are probably predictive of brain amyloid deposition level. Therefore, our aim is to investigate the feasibility of extracting cost-effective predictive rules of brain amyloid-β levels for enriching the clinical trials of anti-amyloid treatments when blood-based markers are used as predictors. Rather than focusing on predictive regression models as in most of the relevant existing studies, a decision tree model can lead to simple decision

FIGURE 4. The decision tree model of blood-based markers (M2).



Note. Note that the model is estimated only using the training data, but the classification results of all 218 subjects (including both the training data set and testing data set) are presented.

rules that can be naturally translated into the clinical settings for detecting amyloid-positive cases. Furthermore, these rules will permit some individuals to be classified on the basis of only one measurement, or at most a few, whereas scores derived from regression-based prediction models, such as logistic regression or SVM, require that all covariates be available.

To build the decision tree model, we used 146 blood-based markers from the proteomic data downloaded from the ADNI website. For measurements of amyloid burden, we used both the PiB-PET imaging and the CSF Aβ_{1-42} level. The subjects were then dichotomized into either PiB positive (PiB retention summary measure > 1.5) or PiB negative (PiB retention summary measure < 1.5), based on a threshold used in the literature. The CSF samples were acquired from these subjects by the ADNI Biomarker Core laboratory at the University of Pennsylvania Medical Center. The subjects were then dichotomized into either CSF Aβ_{1-42} positive (CSF Aβ_{1-42} level of ≤ 192 pg/mL) or CSF Aβ_{1-42} negative (CSF Aβ_{1-42} level of > 192 pg/mL), based on a threshold used in the literature. Finally, a subject is classified as amyloid positive if this subject is positive either by PiB-PET or CSF Aβ_{1-42}. Data used for the analyses presented here were accessed on May 11, 2013 and comprise data from 50 NC and 168 MCI subjects for which blood proteomics data and Aβ status were available.

Of the 146 blood-based markers, the decision tree algorithm automatically identified 5 blood-based markers that are predictive to the amyloid pathology. These five markers are apolipoprotein E (APOE), prostatic acid phosphatase (PAP), transthyretin (TTR), matrix metalloproteinase-10 (MMP10), and myoglobin (MYOGLOBN). It also identified two homogeneous subgroups, Node 3 (the majority is amyloid negative) and a merge of Nodes 10 and 11 (the majority is amyloid positive). These two subgroups are characterized by two rules, M2_Rule1: APOE > 1.785 AND TTR > 2.569 and M2_Rule2: APOE ≤ 1.785 AND PAP ≤ -0.638 AND MMP10 ≥ -1.481, respectively. The tree structure is shown in Figure 4.

This model identified effective prediction models for detecting subjects with elevated amyloid burden. Simple rules are found to be predictive of brain amyloid level. These rules use cost-effective measurements and also permit some individuals to be classified on the basis of only one measurement, or at most a few. For example, as shown in Figure 4, 43% of the 216 subjects have an APOE plasma value >1.785, and only 33% of this group are

amyloid positive. By contrast, 57% of the 216 subjects have an APOE plasma value ≤ 1.785, and 67% of this group are amyloid positive. This implies that by implementing the decision rule, APOE ≤ 1.785, the amyloid-positive population will be enriched twofold. Therefore, as long as these rules can be clinically validated, we believe that these simple decision rules can be naturally translated into clinical settings, such as enrichment screening for Alzheimer's prevention trials of anti-amyloid treatments.

4.2. Selecting Brain Biomarkers in Attention Deficit Hyperactivity Disorder

Current diagnosis of attention deficit hyperactivity disorder (ADHD) predominantly relies on a series of subjective tests/evaluations and parental surveys that are in turn coupled with the personal experience of attending physicians. An accurate biomarker for objective diagnosis of ADHD remains a challenging task for researchers and clinicians. Recent advances in neuroimaging have enabled neuroscientists to search for both structural (e.g., cortical thickness, brain volume) and functional (functional connectivity) abnormalities in the brain that can potentially be used as new biomarkers of ADHD. However, structural and functional characteristics of neuroimaging data, especially MRI, usually generate a large number of features (e.g., cortical regions, regions of interest). With a limited sample size, traditional machine learning techniques can be problematic for discovering the true diagnostic features of ADHD as a result of overfitting, computational burden, and the interpretability of the model. This paper presents an example of how optimization and statistics methods for regularization and classification can be applied to this ADHD differential diagnosis problem (i.e., classifying ADHD versus control). Based on MRI scans, the features of our interest are normalized brain cortical thicknesses of the gray matter in standard cortical regions of interest (ROIs), which were generated by an automated labeling system based on gyral regions (Desikan et al. [8]). Cortical thickness features are thought to provide specific information about neuronal loss or degradation indicated by thinning of the cortex, which might serve as a robust biomarker of ADHD. We developed a new computational framework based on an integration of the stepwise feature selection using an information-theoretic criterion and regularization using the LASSO method. Our framework can optimize a sparse prediction model while ensuring that the most informative features are included in the model.

The framework operates in two main steps. The first step is to quantify and rank individual features based on their mutual information (MI) scores. Generally speaking, MI is a measure of the inherent dependence expressed in the joint distribution of X and Y relative to the joint distribution of X and Y under the assumption of independence. It can be expressed by $I(X,Y) = \sum_{y\in Y}\sum_{x\in X} p(x,y)\log(p(x,y)/(p(x)p(y)))$. In our case, MI measures how much information a feature X contains about the class (or response value) Y without making any assumptions about the nature of their underlying relationships. After all features are ranked in a descending order of their MI values, a subset of high-ranked redundant features is removed. Redundant features are identified based on pairwise correlation between the features, and they are removed to prevent multicollinearity. In the second step, we use a generalized LASSO model by setting the nonredundant top MI features to be penalty-free. A generalized LASSO model is in the form of $\min_{\beta\in R^p} \|\mathbf{y} - X\beta\|_2^2 + \lambda\|D\beta\|_1$, where D is a $p \times p$ matrix representing penalties of features and joint penalties between feature pairs. The generalized LASSO model can be reduced as an adaptive LASSO and solved as the standard LASSO using any convex optimization algorithm. For the penalty-free features, we can simply set the diagonal elements associated with those features to be 0 in the D matrix. We note that our LASSO model explores the full feature space of all other features that are not penalty-free; that is, we allow other features that do not have high MI values to be selected if their contributions to the prediction model are deemed to improve the model accuracy. One main parameter of our framework is the number of nonredundant top

Table 1. Classification results of five different feature selection methods.

Feature selection method	Training accuracy (%)	No. of selected features	Testing accuracy (%)
Our framework	87	4	81
MRMR	78	5	76
CMIM	76	5	74
Floating search	76	7	66
PCA	74	14	70

MI features to be set as penalty-free. In our empirical investigation, we iteratively set the number of penalty-free features to be 1, 2, and 3.

In our study, we tested our classification framework on a data set of 47 right-handed subjects matched on gender, socioeconomic status, and ethnicity. In the data set, there were 32 ADHD subjects and 15 healthy control subjects. All ADHD subjects in this study had less than 15 standard score point differences between general conceptual ability and all achievement measures, and they were matched on severity of symptoms based on Conners' Ratings Scale. In addition, all ADHD subjects met the Diagnostic and Statistical Manual of Mental Disorders criterion for ADHD and did not meet all other psychiatric or psychological disorders including learning disorders, anxiety disorders, mood disorders, or oppositional defiant disorders. Healthy control subjects did not meet any criteria for a psychiatric or learning diagnosis nor had a history of medication treatment. All MRI images were acquired at the University of Texas Health Science Center at San Antonio using three-dimensional gradient recalled acquisitions in the study state with a repetition time (TR)of 33 ms, echo time (TE) of 12 ms, and a flip angle of 60° to obtain a $256 \times 192 \times 192$ volume of data with a spatial resolution of 1 mm $\times$ 1 mm $\times$ 1 mm. Subsequently, all MRI images were processed and normalized using the FreeSurfer image analysis suite (Dale et al. [7]). All 45 cortical ROIs were extracted and their cortical thicknesses measured using the FreeSurfer suite. Cortical thickness can be measured in a two-step process. The first step is to find surfaces between white matter (WM) and gray matter (GM) and GM and CSF by segmenting brain volume data. The second step is to calculate the distance between the WM–GM surface and the GM–CSF surface.

In this study, we employed a leave-one-out cross-validation procedure to train and test our prediction framework. As mentioned above, our framework iteratively searched for different numbers of penalty-free features. It was found that the optimal number of penalty-free features was four. The classification performance of our framework achieved a testing accuracy of 81%, as shown in Table 1. We also implemented and compared the classification performance of other state-of-the-art feature selection algorithms including minimal-redundancy-maximal-relevance (MRMR) (Peng et al. [27]), conditional mutual information maximization (CMIM) (Fleuret [9]), as well as the popular Pudil's floating search method (Pudil et al. [28]) and the principle component analysis (PCA)-based approach with 95% data variance. As shown in Table 1, compared with the state-of-the-art feature selection approaches, our prediction framework yields the highest testing prediction accuracy while using the minimum number of features in the final prediction model. The selected regions of interest in our model were consistent with recent neurological studies of ADHD and thus confirmed the interpretability of the selected features by the proposed approach.

5. Conclusion

In this tutorial we give an overview of computational optimization and statistical methods for big data analytics that includes a range of techniques developed in machine learning, data mining, knowledge discovery, artificial intelligence, and statistical learning. As data-driven methods, these techniques provide a powerful data analysis capability to "learn"

information directly from data without assuming a predetermined model of the data. We review some classification and regression models such as support vector machine, regression models, decision tree, Bayesian learning, and nearest neighbor. We also introduce feature selection and recently developed sparse learning methods. As a demonstration of real-life applications of these analytics techniques, we present results of studies in neuroimaging data analysis and show how knowledge discovery and predictive models can be achieved using data analytics. In addition to medicine, big data is becoming a ubiquitous challenge in many disciplines and sectors, calling for intellectual innovations in analytics techniques to cope with the challenge and convert all kinds of big data into values.

Acknowledgments

The authors thank Danica Cao Xiao and Margaret Semrud-Clikeman, Jesse Bledsoe, and Shouyi Wang for their help in collecting and analyzing the neuroimaging data used in the ADHD study. This work is supported by the National Science Foundation under Grants CMMI-1505260 and CMMI-1333841.

References

[1] A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine Learning* 73(3):243–272, 2008.

[2] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Occam's razor. *Information Processing Letters* 24(6):377–380, 1987.

[3] E. Candès, M. Wakin, and S. Boyd. Enhancing sparsity by reweighted L1 minimization. *Journal of Fourier Analysis and Applications* 14(5):877–905, 2008.

[4] W. Chaovalitwongse, Y.-J. Fan, and R. C. Sachdeo. Support feature machine for classification of abnormal brain activity. *Proceedings of the* 13*th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (*SIGKDD*), ACM, New York, 113–122, 2007.

[5] W. Chaovalitwongse, Y.-J. Fan, and R. C. Sachdeo. Novel optimization models for abnormal brain activity classification. *Operations Research* 56(6):1450–1460, 2008.

[6] C. A. Chou, K. Kampa, S. H. Mehta, R. F. Tungaraza, W. Chaovalitwongse, and T. J. Grabowski. Voxel selection framework in multi-voxel pattern analysis of fMRI data for prediction of neural response to visual stimuli. *IEEE Transactions on Medical Imaging* 33(4):925–934, 2014.

[7] A. M. Dale, B. Fischl, and M. I. Sereno. Cortical surface-based analysis: I. Segmentation and surface reconstruction. *NeuroImage* 9(2):179–194, 1999.

[8] R. S. Desikan, F. Ségonne, B. Fischl, B. T. Quinn, B. C. Dickerson, D. Blacker, R. L. Buckner, 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] F. Fleuret. Fast binary feature selection with conditional mutual information. *Journal of Machine Learning Research* 5:1531–1555, 2004.

[10] J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics* 9(3):432–441, 2008.

[11] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research* 3:1157–1182, 2003.

[12] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*, 2nd ed. Springer, New York, , 2009.

[13] A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* 12(1):55–67, 1970.

[14] C.-W. Hsu and C.-J. Lin. A comparison of methods multi-class support vector machines. *IEEE Transactions on Neural Networks* 13(2):415–425, 2002.

[15] S. Huang, J. Li, L. Sun, J. Ye, A. Fleisher, T. Wu, K. Chen, and E. Reiman. Learning brain connectivity of Alzheimer's disease by sparse inverse covariance estimation. *NeuroImage* 50(3):935–949, 2010.

[16] S. Huang, J. Li, J. Ye, L. Chen, T. Wu, A. Fleisher, and E. Reiman. Identifying Alzheimer's disease-related brain regions from multi-modality neuroimaging data using sparse composite linear discrimination analysis. J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, eds. *Advances in Neural Information Processing Systems 24: Proceedings of Neural Information Processing Systems Conference*, Curran Associates, Red Hook, NY, 1431–1439, 2011.

[17] S. Huang, J. Li, J. Ye, A. Fleisher, K. Chen, T. Wu, and E. Reiman. A sparse structure learning algorithm for Bayesian network identification from high-dimensional data. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35(6):1328–1342, 2013.

[18] K. Kampa, S. H. Mehta, C. A. Chou, W. Chaovalitwongse, and T. J. Grabowski. Sparse optimization in feature selection: Application in neuroimaging. *Journal of Global Optimization* 59(2–3):439–457, 2014.

[19] U. H.-G. Kreßel. Pairwise classification and support vector machines. B. Schölkopf, C. J. C. Burges, and A. J. Smola, eds. *Advances in Kernel Methods: Support Vector Learning*. MIT Press, Cambridge, MA, 255–268, 1999.

[20] H. Liu and H. Motoda. *Feature Selection for Knowledge Discovery and Data Mining*. Kluwer Academic Publishers, Boston, 1998.

[21] J. Liu, S. Ji, and J. Ye. SLEP: Sparse learning with efficient projections. Arizona State University, Tempe. http://www.yelab.net/software/SLEP/, 2009.

[22] O. L. Mangasarian. Linear and nonlinear separation of pattern by linear programming. *Operations Research* 13(3):445–453, 1965.

[23] O. L. Mangasarian. Multisurface method for pattern separation. *IEEE Transactions on Information Theory* 14(6):801–807, 1968.

[24] R. Mazumder, J. H. Friedman, and T. Hastie. SparseNet: Coordinate descent with nonconvex penalties. *Journal of the American Statistical Association* 106(495):1125–1138, 2011.

[25] D. C. Montgomery, E. A. Peck, and G. G. Vining. *Introduction to Linear Regression Analysis*, 5th ed. John Wiley & Sons, New York, 2012.

[26] C. Mulert and L. Lemieux, eds. *EEG-fMRI: Physiological Basis, Technique and Applications*. Springer, Berlin, 2010.

[27] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27(8):1226–1238, 2005.

[28] P. Pudil, J. Novovičová, and J. Kittler. Floating search methods in feature selection. *Pattern Recognition Letters* 15(11):1119–1125, 1994.

[29] Z. Qiao, L. Zhou, and J. Huang. Sparse linear discriminant analysis with applications to high dimensional low sample size data. *IAENG Applied Mathematics* 39(1):48–60, 2006.

[30] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Francisco, 1993.

[31] M. Schmidt, G. Fung, and R. Rosales. Fast optimization methods for L1 regularization: A comparative study and two new approaches. J. N. Kok, J. Koronacki, R. Lopez de Mantaras, S. Matwin, D. Mladenič, and A. Skowron, eds. *Machine Learning: ECML 2007*, Lecture Notes in Computer Science, Vol. 4701. Springer, Berlin, 286–297, 2007.

[32] B. Schölkopf, C. Burges, and V. Vapnik. Extracting support data for a given task. U. M. Fayyad and R. Uthurusamy, eds. *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, AAAI Press, Menlo Park, CA, 252–257, 1995.

[33] O. Şeref, W. A. Chaovalitwongse, and J. P. Brooks. Relaxing support vectors for classification. *Annals of Operations Research* 216(1):229–255, 2014.

[34] R. Tibshirani. Regression shrinkage and selection via the LASSO. *Journal of Royal Statistical Society: Series B* 58(1):267–288, 1996.

[35] V. Vapnik. *The Nature of Statistical Learning*. Springer, New York, 1995.

[36] V. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, New York, 1998.

[37] K. Weaver, W. Chaovalitwongse, E. J. Novotny, A. D. Poliakov, T. J. Grabowski, and J. Ojemann. Local functional connectivity as a pre-surgical tool for seizure focus identification in non-lesion, focal epilepsy. *Frontiers in Neurology* 4(43):1–14, 2013.

[38] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of Royal Statistical Society: Series B* 68(1):49–67, 2006.

[39] T. Zhang. Multi-stage convex relaxation for learning with sparse regularization. D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, eds. *Advances in Neural Information Processing Systems 21: Proceedings of Neural Information Processing Systems Conference*, Curran Associates, Red Hook, NY, 1929–1936, 2008.

[40] H. Zou, T. Hastie, and R. Tibshirani. Sparse PCA. *Journal of of Computational and Graphical Statistics* 15(2):265–286, 2006.