INVITED PAPER

# Classification of Big Data With Application to Imaging Genetics

*This paper investigates the classification of big data with a particular focus on classification of 3-D MRI images of the human brain in a genetic context.*

By Magnus Orn Ulfarsson, *Member IEEE*, Frosti Palsson, *Student Member IEEE*, Jakob Sigurdsson, *Member IEEE*, and Johannes R. Sveinsson, *Senior Member IEEE*

**ABSTRACT** | Big data applications, such as medical imaging and genetics, typically generate datasets that consist of few observations $n$ on many more variables $p$, a scenario that we denote as $p \gg n$. Traditional data processing methods are often insufficient for extracting information out of big data. This calls for the development of new algorithms that can deal with the size, complexity, and the special structure of such datasets. In this paper, we consider the problem of classifying $p \gg n$ data and propose a classification method based on linear discriminant analysis (LDA). Traditional LDA depends on the covariance estimate of the data, but when $p \gg n$, the sample covariance estimate is singular. The proposed method estimates the covariance by using a sparse version of noisy principal component analysis (nPCA). The use of sparsity in this setting aims at automatically selecting variables that are relevant for classification. In experiments, the new method is compared to state-of-the art methods for big data problems using both simulated datasets and imaging genetics datasets.

**KEYWORDS** | Factor analysis; image classification; imaging genetics; linear discriminant analysis; noisy principal component analysis; support vector machines; $l_0$ vector penalty

## I. INTRODUCTION

Recent technological achievements and globalization have increased data acquisition capability in almost all corners of human activities, ranging from scientific and engineering endeavors such as genomics, medical imaging, remote sensing, economics, and finance, and all the way to people's personal lives with the emergence of social media through the World Wide Web and mobile networks. The enormous growth of data creates daunting challenges, not only in finding out how to store and access the data, but more importantly, how to process and make sense of it. Also, since data collection is expensive, we are somehow obliged to make good use of the data at hand, so it is obvious that for further progress, the development of efficient algorithms for processing big data is very important.

Big data is usually considered in terms of the number of observations $n$ and the number of variables $p$ measured on each observation. In many branches of science such as genetics and medical imaging, the number of variables is very large and is often much larger than the number of observations. This scenario is often denoted as $p \gg n$. This makes information extraction challenging due to high variance and risk of overfitting. In the last decade, much effort has been put into developing and analyzing algorithms for $p \gg n$ problems in statistical signal processing and related fields.

An important problem in dealing with big data is how to design and build algorithms that recognize and classify patterns. This problem is in the realm of statistical learning [1]–[3]. There are three types of statistical learning problems, i.e., supervised learning, unsupervised learning, and reinforcement learning. Supervised learning, also known as classification, is based

Manuscript received July 24, 2015; revised October 6, 2015; accepted November 14, 2015. Date of publication March 31, 2016; date of current version October 18, 2016. This work was supported in part by the Research Fund of the University of Iceland, by the Icelandic Research Fund under Grant 130635-051, and by the European Community's Seventh Framework Programme under FP7/2007–2013, Grant Agreement 602450, IMAGEMEND.

**M. O. Ulfarsson** is with the Faculty of Electrical and Computer Engineering, University of Iceland, 107 Reykjavik, Iceland, and also with deCODE genetics/Amgen, 107 Reykjavik, Iceland (e-mail: mou@hi.is).

**F. Palsson**, **J. Sigurdsson**, and **J. R. Sveinsson** are with the Faculty of Electrical and Computer Engineering, University of Iceland, 107 Reykjavik, Iceland.

Digital Object Identifier: 10.1109/JPROC.2015.2501814

on using labeled training data for constructing a function that is able to classify new observations. On the other hand, unsupervised learning, also known as clustering, is based on using distance measures to find hidden structures in unlabeled data. Finally, reinforcement learning [4] operates in a real-time setting where the classifier uses a feedback mechanism to improve results. In this paper, we are concerned with supervised learning, i.e., classification. The development and use of classification algorithms is important in many applications areas, including remote sensing [5], handwritten digit recognition [6], face recognition [7], DNA expression microarrays [8], and functional magnetic resonance imaging (fMRI) [9].

A rich source of high-dimensional data is structural magnetic resonance imaging (MRI). MRI is a noninvasive method that can be used to visualize the living human brain. It delivers detailed brain maps in high spatial resolution that have proven to be very useful, both in a clinical and research setting, for detecting structural features in the brain. A typical imaging experiment acquires MRI data from two groups of people: a control group and a group consisting of subjects having some specific condition, such as a disease. In [10], it was shown that a support vector machine (SVM) classifier can successfully distinguish subjects with mild clinically probable Alzheimer's disease from age/sex matched controls in 89% of cases; reference [11] used an SVM classifier to investigate how well certain subtypes of schizophrenia patients can be discriminated from healthy controls; reference [12] developed a classification method based on discriminant analysis for differentiating between schizophrenia patients and controls; reference [13] classified 71.4% of schizophrenia patients correctly from healthy controls in a large sample using an SVM; reference [14] proposed to use an SVM classifier to separate bipolar patients from schizophrenia patients.

Since the completion of the draft sequence of the human genome [15], there has been an explosion in the availability of genomic data. This has been driven by the development of more efficient sequencing and genotyping technologies. The availability of genetic data has made it possible to investigate the association of genetic variants with a disease, e.g., Alzheimer's disease [16], or a quantitative phenotype, e.g., height [17]. There are several kinds of DNA sequence variations, and these include single-nucleotide polymorphisms (SNPs), which are variations at the single nucleotide level that occur commonly in the population, and copy-number variations (CNVs), which are deletions or duplications of genomic regions, ranging from a several base pairs to large fractions of chromosomes.

The concurrent availability of genomic and neuroimaging data has launched the field of imaging genetics [18]. The aim of imaging genetics is to view neuroimaging data as a multivariate phenotype and investigate how genetic variations affect the brain. A voxelwise genome-wide association (vGWAS) was used in [19] to search for genetic variants that influence brain structure; references [20] and [21] developed a sparse reduced-rank method for vGWAS. The effect of a certain CNV, which confers a risk for schizophrenia, on the brain was demonstrated in [22]. Currently, there are large-scale imaging genetics projects in operation, e.g., [23] and [24].

The size of the datasets involved in imaging genetics problems is typically very large and of the $p \gg n$ type. Here, a novel method for classifying $p \gg n$ data is developed. The method is based on linear discriminant analysis (LDA) and uses a covariance model based on a sparse version of noisy principal component analysis (nPCA). Sparsity enables the method to automatically retain variables that are important for classification and discard the rest. At the same time, this sparsity increases the interpretability and also decreases computational time. We call the new method LDA-svnPCA$_0$.

The remainder of the paper is organized as follows. The classification problem in general is reviewed in Section II. Section III presents LDA in the $p \gg n$ settings and discusses the use of the nPCA covariance estimate for LDA. In Section IV, the proposed LDA-svnPCA$_0$ method is detailed. Section V describes the structural MRI data used in this paper and the associated feature extraction and selection methods used in the experiments. Then, in Section VI, experimental results are presented. Finally, in Section VII, the conclusion is drawn.

### A. Notation

Vectors and matrices are denoted using bold-faced symbols. Row vector number $i$ of a matrix $\boldsymbol{X}$ is denoted by $\mathbf{x}_i^T$, column vector $j$ is denoted by $\mathbf{x}_{(j)}$; the $i$th element of a vector $\mathbf{x}$ is denoted as $x_i$; $\text{sign}(\mathbf{x})$ is a vector of the same size of $\mathbf{x}$ with its $i$th element equal to the sign of $x_i$. The hinge loss function is defined as $|x|_+ = x$ if $x \geq 0$ and 0 otherwise; $\mathbf{x} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ means that the random vector $\mathbf{x}$ is Gaussian distributed with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$; $I(x \geq h)$ is the indicator function that is equal to one if the inequality is true and zero otherwise; $E[x]$ denotes the expectation of a random variable $x$; $\boldsymbol{I}_p$ is the $p \times p$ identity matrix.

## II. CLASSIFICATION

The general setup of a classification problem is that there is an object that belongs to one of $K$ classes $C_1, \ldots, C_K$, and the aim is to determine to which class it belongs. This object could for example be an individual that has one of $K$ different genetic variants. Associated with the object is a measurement of a feature vector $\mathbf{x}$ and behind each feature vector is a true unknown discrete valued function $c(\mathbf{x})$ where $c(\mathbf{x}) = k$ if and only if $\mathbf{x} \in C_k$. A classifier is an estimator $\hat{c}(\mathbf{x})$ of this discrete valued function.

The first step in measuring the performance of a classifier is to define a loss function that quantifies the cost associated with misclassification. A commonly used loss function is the 0–1 loss given by

$$L(k,l) = \begin{cases} 1, & l \neq k \\ 0, & l = k. \end{cases}$$

Associated with this loss function is a risk function that is the expected loss of a classifier $\hat{c}$ when the true class is $k$, i.e.,

$$\begin{aligned} R(k,\hat{c}) &= E[L(c,\hat{c})|c=k] \\ &= \sum_{l=1}^{K} L(k,l) Pr(\hat{c}=l|c=k) \\ &= Pr(\hat{c} \neq k|c=k). \end{aligned}$$

The risk is simply equal to the probability of a misclassification. The performance of a classifier is measured by the total risk, or the so-called Bayes risk given by

$$\begin{aligned} R_\pi(\hat{c}) &= E[R(k,\hat{c})] \\ &= \sum_{k=1}^{K} Pr(\hat{c} \neq k|c=k)\pi_k \end{aligned}$$

where $\pi_k$ is the prior probability of the $k$th class. It can be shown [2] that a classifier that minimizes the Bayes risk is given by the Bayes classifier

$$\hat{c}(\mathbf{x}) = \arg\max_l p(l|\mathbf{x},\boldsymbol{\theta}) \tag{1}$$

that assigns a new sample to the most probable class according to the posterior class probability density function (pdf)

$$p(l|\mathbf{x},\boldsymbol{\theta}) = \frac{\pi_l p_l(\mathbf{x},\boldsymbol{\theta})}{\sum_{k=1}^{K} \pi_k p_k(\mathbf{x},\boldsymbol{\theta})}$$

where $p_l(\mathbf{x};\boldsymbol{\theta})$ is the pdf of class $l$, and $\boldsymbol{\theta}$ is a parameter vector. In the last few decades, many different classifiers have been developed and shown to be useful for a wide variety of applications. These classifiers include the $k$-nearest neighbors, neural networks, random forests [25], penalized logistic regression [26], support vector machines [27], [28], etc.

### A. Support Vector Machines

Among the most successful classification methods are the SVMs. There are two kinds of SVMs, nonlinear and linear, and for $p \gg n$, the linear SVM has been found to work as well as the nonlinear versions [1]. The two-class linear SVM is given by

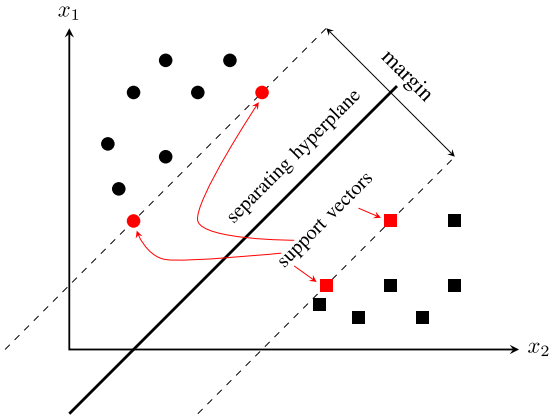


**Fig. 1.** ***Linear SVM illustrated.***

$$\hat{c}(\mathbf{x}) = \text{sign}(f(\mathbf{x}))$$

where $f(\mathbf{x}) = \beta_0 + \boldsymbol{\beta}^T\mathbf{x}$. The main idea behind linear SVM is to find a separating hyperplane, i.e., determine $\beta_0$ and $\boldsymbol{\beta}$, that maximize the margin that separates the closest observations from either class. Fig. 1 shows a diagram illustrating the principle of the linear SVM.

Given training data $(\mathbf{x}_i, c_i), i = 1,\ldots,n$, where $c_i \in \{-1,1\}$ is a label, the SVM problem can be formulated as a penalized regression problem [29]

$$(\hat{\beta}_0,\hat{\boldsymbol{\beta}}) = \arg\min_{\beta_0,\boldsymbol{\beta}} \sum_{i=1}^{n} [1 - c_i f(\mathbf{x}_i)]_+ + \frac{1}{2C}\|\boldsymbol{\beta}\|^2. \tag{2}$$

The first term is the so-called hinge loss, which is a convex function that upper-bounds the 0–1 loss function. The second term is a regularization term that encourages smoothness. Provided a reasonable choice of the tuning parameter $C$ and if the training data is separable, this problem delivers a decision boundary $\{f(\mathbf{x}) = 0\}$ that maximizes the margin between the two classes. It turns out that for $p \gg n$ problems, the choice of $C = \infty$ (no regularization) separates the data [1]. However, some amount of regularization is often preferable and this means that the tuning parameter $C$ has to be selected. The papers [30] and [31] discuss automatic ways for choosing this parameter.

The formulation (2) allows for various extensions of the method, e.g., [32]–[34] investigate smooth alternatives to

the hinge loss function, while sparse extensions of SVM are proposed in [26] and [35]–[40].

## III. LINEAR DISCRIMINANT ANALYSIS

LDA is probably one of the most commonly used classification techniques. Similarly to the linear SVM, it aims to find a linear decision boundary that separates the classes of interest. LDA assumes that the class probabilities are multivariate normal distributions with a common covariance matrix $\boldsymbol{\Sigma}$ of size $p \times p$ and centroids $\boldsymbol{\mu}_k, k = 1, \ldots, K$, i.e., the pdf of the $k$th class is given by $p_k(\mathbf{x}, \boldsymbol{\theta}) \sim N(\boldsymbol{\mu}_k, \boldsymbol{\Sigma})$.

In this Gaussian framework, it is easier to work with the so-called discriminant function $\delta_k(\mathbf{x}) = \log(p_k(\mathbf{x}, \boldsymbol{\theta})\pi_k)$ that is given by (ignoring irrelevant terms)

$$\delta_k(\mathbf{x}) = \mathbf{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k - \frac{1}{2}\boldsymbol{\mu}_k^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k + \log \pi_k \tag{3}$$

and then assign the sample $\mathbf{x}$ to a class according to the decision rule [equivalent to (1)]

$$\hat{c}(\mathbf{x}) = \arg\max_k \delta_k(\mathbf{x}). \tag{4}$$

In practice, the parameters $\boldsymbol{\Sigma}$ and $\boldsymbol{\mu}_k, k = 1, \ldots, K$ need to be estimated from the data. Given training data $(\mathbf{x}_i, c_i), i = 1, \ldots, n$ where $n > p$ one traditionally uses the following estimates:

$$\hat{\boldsymbol{\mu}}_k = \frac{1}{n_k}\sum_{i \in C_k} \mathbf{x}_i, \qquad k = 1, \ldots, K \tag{5}$$

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{n-K}\sum_{k=1}^{K}\sum_{i \in C_k}(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)^T \tag{6}$$

$$\pi_k = \frac{n_k}{n}, \qquad k = 1, \ldots, K \tag{7}$$

where $C_k = \{i : c_i = k\}$, and $n_k = |C_k|$. We note that the above is not the only way for motivating LDA. It can also be motivated from optimal scoring, and Fisher's discriminant analysis [41].

### A. Big Data Case: $p \gg n$

In the big data case, where one has many more variables than samples ($p \gg n$), the covariance estimate (6) is singular, and therefore the LDA classifier cannot be constructed. In this case, researchers have suggested to constrain the estimate to be positive definite. In [42]–[44], the covariance was constrained to be diagonal, i.e., $\hat{\boldsymbol{\Sigma}}_d = \text{diag}(\hat{\sigma}_1^2, \ldots, \hat{\sigma}_p^2)$, where $\hat{\sigma}_j^2$ is the $j$th diagonal element of (6). An alternative approach for avoiding the singularity problem was presented in [45] where the regularized covariance estimate

$$\tilde{\boldsymbol{\Sigma}} = \alpha\hat{\boldsymbol{\Sigma}} + (1-\alpha)\boldsymbol{I}_p \tag{8}$$

is used for some $\alpha > 0$.

Another problem with LDA in high dimensions is that the classifier is a linear combination of all the $p$ variables, which hinders interpretability. A solution to this problem is to select a small subset of the variables while conserving the discriminative power of the classifier. The paper [46] proposed the nearest shrunken centroid (NSC) method that is based on traditional LDA using a diagonal covariance matrix. The idea is to shrink the class centroids toward the overall mean. In detail, the $k$th NSC discriminant function is given by

$$\delta_k(\mathbf{x}) = \sum_{j=1}^{p}\frac{x_j\tilde{\mu}_{kj} - \frac{1}{2}\tilde{\mu}_{kj}^2}{\hat{\sigma}_j^2} + \log \pi_k$$

where the shrunken centroids $\tilde{\mu}_{kj}$ are given by

$$\begin{aligned}\tilde{\mu}_{kj} &= \hat{\mu}_j + (\hat{\sigma}_j + \gamma)\tilde{d}_{kj}\\ \tilde{d}_{kj} &= \max\big(|d_{kj}| - h, 0\big)\text{sign}(d_{kj})\\ d_{kj} &= \frac{\hat{\mu}_{kj} - \hat{\mu}_j}{m_k(\hat{\sigma}_j + \gamma)}\end{aligned}$$

where $\hat{\mu}_{kj}$ is the $k$th centroid, $\hat{\mu}_j$ is the overall mean, $m_k = \sqrt{(1/n_k) - (1/n)}$, and $\gamma$ is a small positive constant. By increasing $h$, the method zeroes out some of the $\tilde{d}_{kj}$ variables, which in turn means that the corresponding variables $x_j$ do not contribute to the class prediction. The tuning parameter $h$ has to be selected, and [46] suggests to use $L$-fold cross-validation.

In [45], the shrunken centroid idea was combined with using the regularized covariance estimate (8) in the LDA framework. A covariance model based on the noisy principal component analysis (nPCA) (see Section III-B) was used in [47], and a sparse optimization framework was used to select variables. The LDA was formulated as an $l_1$ penalized optimal scoring problem in [48] to achieve automatic variable selection; in [49], LDA was formulated as an $l_1$ penalized Fisher's linear discriminant problem with the same objective. The paper [50] provides a discussion and compares the formulation of sparse optimal scoring and sparse Fisher's discriminant analysis.

In the following, the singularity issue of the covariance matrix is solved by using an estimate that is positive definite and relates to nPCA.

### B. Noisy Principal Component Analysis

The nPCA model is a multivariate model of the following form:

$$\mathbf{x} = \boldsymbol{\mu} + \boldsymbol{G}\boldsymbol{u} + \boldsymbol{\epsilon} \tag{9}$$

where $\boldsymbol{\mu}$ is the mean vector, $\boldsymbol{G}$ is a $p \times r$ matrix, $p \gg r$, $\boldsymbol{u} \sim N(\boldsymbol{0}, \boldsymbol{I}_r)$ contains $r$ noisy principal components, $\boldsymbol{\epsilon} \sim N(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_p)$ is noise, and $\boldsymbol{u}$ and $\boldsymbol{\epsilon}$ are independent. It can be shown that $\mathbf{x} \sim N(\boldsymbol{\mu}, \boldsymbol{\Omega})$, where $\boldsymbol{\Omega} = \boldsymbol{G}\boldsymbol{G}^T + \sigma^2 \boldsymbol{I}_p$ is the nPCA model of the covariance.

The nPCA model has been found to be useful in signal processing applications such as array signal processing [51] and medical imaging applications [52]. The model was originally developed by [53] but later popularized under the name probabilistic PCA [54], however, we prefer to call it nPCA.

Given a sample $\mathbf{x}_i, i = 1, \ldots, n$, the unknown model parameters $\boldsymbol{\mu}, \boldsymbol{G}, \sigma^2$ are estimated using the maximum likelihood principle, yielding [53]

$$\hat{\boldsymbol{\mu}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i$$

$$\hat{\boldsymbol{G}} = \boldsymbol{P}_r(\boldsymbol{L}_r - \hat{\sigma}^2 \boldsymbol{I}_r)^{\frac{1}{2}}\boldsymbol{R} \tag{10}$$

$$\hat{\sigma}^2 = \frac{1}{M-r}\sum_{j=r+1}^{M} l_j \tag{11}$$

where $\boldsymbol{L}_r = \mathrm{diag}(l_1, \ldots, l_r)$ contains the $r$ largest eigenvalues of the data covariance matrix $\boldsymbol{S}_x = (1/n)\sum_{i=1}^{n}(\mathbf{x}_i - \hat{\boldsymbol{\mu}})(\mathbf{x}_i - \hat{\boldsymbol{\mu}})^T$, $\boldsymbol{R}$ is an arbitrary $r \times r$ orthogonal rotation matrix, and $\boldsymbol{P}_r$ contains the $r$ first eigenvectors of $\boldsymbol{S}_x$ in its columns. In practice, $\boldsymbol{R}$ can be set equal to the identity matrix. Alternatively, the nPCA model parameters can be computed using an expectation-maximization (EM) algorithm [54].

The nPCA model can be extended in various ways. In [55], a method called sparse variable nPCA (svnPCA) was proposed that is based on performing nPCA while automatically discarding irrelevant variables from the analysis.

A closely related model to the nPCA model is the factor analysis (FA) model [56]. The difference is that the noise term in (9) is $\boldsymbol{\epsilon} \sim N(\boldsymbol{0}, \boldsymbol{\Psi})$ where $\boldsymbol{\Psi} = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_p^2)$. In this case, the observed data is also Gaussian distributed but with a different covariance matrix $\boldsymbol{\Omega} = \boldsymbol{G}\boldsymbol{G}^T + \boldsymbol{\Psi}$. Unlike nPCA, there are no closed-form solutions for $\boldsymbol{G}$ and $\boldsymbol{\Psi}$, so an iterative algorithm is needed to estimate them.

### C. Linear Discriminant Analysis Using Noisy Principal Component Analysis

Our motivation for introducing nPCA is to use it to estimate the covariance $\boldsymbol{\Omega}$ of the observed data. The benefit of using this method is the large difference in the number of covariance parameters that needs to be estimated. In the case of nPCA, the number of covariance parameters is $pr + 1 - r(r-1)/2$. This can be contrasted with the number of parameters in the full covariance model, which is $p(p+1)/2$, which is much larger if $r$ is a relatively small number. A discussion of the use of nPCA, and the closely related FA, as a covariance model can be found in [54] and [57].

In the following, we propose to use the nPCA covariance model $\boldsymbol{\Omega}$ in the LDA model. This involves exchanging $\boldsymbol{\Sigma}^{-1}$ for $\boldsymbol{\Omega}^{-1}$ in (3). In our applications, the variable dimension $p$ is very high, and therefore the construction of $\boldsymbol{\Omega}^{-1}$ is unfeasible. Instead, we use the matrix inversion lemma

$$\boldsymbol{\Omega}^{-1} = \frac{1}{\sigma^2}\boldsymbol{I}_p - \frac{1}{\sigma^2}\boldsymbol{G}\boldsymbol{W}^{-1}\boldsymbol{G}^T$$
$$\boldsymbol{W} = \sigma^2 \boldsymbol{I}_r + \boldsymbol{G}^T\boldsymbol{G}$$

and note that to construct the discriminant function, it is only necessary to construct $\boldsymbol{\Omega}^{-1}\boldsymbol{\mu}_k$, which is much cheaper than having to explicitly form the covariance.

## IV. LINEAR DISCRIMINANT ANALYSIS USING SPARSE NOISY PRINCIPAL COMPONENT ANALYSIS

As stated above, the lack of interpretability is a problem in the $p \gg n$ scenario. In this paper, we propose a method to automatically drop out variables that do not contribute to the class prediction. To begin reformulating the LDA problem for better interpretation, we write the model for class $k$ in terms of the distance of the $k$th centroid from the mean vector, i.e., $\boldsymbol{d}_k = \hat{\boldsymbol{\mu}}_k - \hat{\boldsymbol{\mu}}$. Then the discriminative function for class $k$ can be written as

$$\delta_k(\mathbf{x}) = \tilde{\mathbf{x}}^T\boldsymbol{\Omega}^{-1}\boldsymbol{d}_k - \frac{1}{2}\boldsymbol{\mu}_k^T\boldsymbol{\Omega}^{-1}\boldsymbol{d}_k + \log \pi_k$$

where $\tilde{\mathbf{x}} = \mathbf{x} - \hat{\boldsymbol{\mu}}$. Now we want to identify variables $\tilde{x}_j$ that can be discarded from the analysis while maintaining the class discrimination. First note that if the $j$th element of $\boldsymbol{\Omega}^{-1}\boldsymbol{d}_k$ is zero, then the $j$th variable can be discarded from the classifier. Now note that this element can be written as

$$[\boldsymbol{\Omega}^{-1}\boldsymbol{d}_k]_j = \frac{d_{kj}}{\sigma^2} - \frac{1}{\sigma^2}\boldsymbol{g}_j^T\boldsymbol{W}^{-1}\boldsymbol{G}^T\boldsymbol{d}_k$$

where $\boldsymbol{g}_j^T$ is the $j$th column of $\boldsymbol{G}$. From this formula, we see that the variable $j$ does not play a part in the $k$th discriminative function if $d_{kj} = 0$ and $\boldsymbol{g}_j = \boldsymbol{0}$. So in general, we can discard the $j$th variable if all the elements in the vector

$$\boldsymbol{a}_j = \left[d_{1j}, d_{2j}, \ldots, d_{Kj}, \boldsymbol{g}_j^T\right]^T$$

are zero.

The method we propose, which we call LDA-svnPCA$_0$, is based on maximizing the discriminant functions, while automatically discarding variables that do not contribute to the class prediction, by encouraging the $\boldsymbol{a}_j$ vectors to be zero. Given training data $(\mathbf{x}_i, c_i)$, $i = 1, \ldots, n$, we propose to estimate the parameters $\boldsymbol{\theta} = \{\boldsymbol{G}, \sigma^2, \boldsymbol{d}_1, \ldots, \boldsymbol{d}_K\}$ of the classifier by solving the following optimization problem:

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}}(\mathcal{X}) \tag{12}$$

where $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ is the observed data, and $J_{\boldsymbol{\theta}}(\mathcal{X})$ is a penalized discriminant function given by

$$J_{\boldsymbol{\theta}}(\mathcal{X}) = \frac{1}{n}\sum_{k=1}^{K}\sum_{\mathbf{x}_i \in C_k} \delta_k(\mathbf{x}_i) - \frac{h}{2}\sum_{j=1}^{p} |||\boldsymbol{a}_j|||_0. \tag{13}$$

This cost function aims to maximize the discriminant function, $\delta_k(\mathbf{x}_i)$, for each class while using the vector $l_0$ penalty, $|||\boldsymbol{a}_j|||_0$, to enforce some of the $\boldsymbol{a}_j = [d_{1j}, \ldots, d_{Kj}, \boldsymbol{g}_j^T]^T$ to be zero. The vector $l_0$ penalty works in the following way: $|||\boldsymbol{a}_j|||_0 = I(\boldsymbol{a}_j \neq \boldsymbol{0}) = 1$ if $\|\boldsymbol{a}_j\| \neq 0$, and otherwise it equals 0. In other words, when $\boldsymbol{a}_j \neq \boldsymbol{0}$, then $h/2$ is subtracted from the criterion, and when $\boldsymbol{a}_j = \boldsymbol{0}$, nothing gets subtracted. The vector $l_0$ penalty has previously been used in other context, e.g., for sparse principal component analysis [55]; sparse independent component analysis [58]; multivariate regression [59]; hyperspectral denoising [60]; magnetoencephalography (MEG) [61], [62]. It either totally removes or keeps the vector $\boldsymbol{a}_j$, and it should not be confused with the scalar $l_0$ penalty $\|\boldsymbol{a}_j\|_0 = \sum_i I(\boldsymbol{a}_{ij} \neq 0)$, which only removes or keeps individual elements of $\boldsymbol{a}_j$.

The optimization problem (12) does not have a closed-form solution, so we need to resort to an iterative algorithm. Fortunately, there is an efficient EM algorithm [63] that can solve this problem.

### A. EM Algorithm

The EM algorithm is an iterative algorithm that is often efficient at optimizing likelihood functions. Typically, one is interested in estimating a parameter vector $\boldsymbol{\theta}$ from a likelihood $p_{\boldsymbol{\theta}}(\mathbf{x})$. The EM algorithm is primarily useful when the model behind $p_{\boldsymbol{\theta}}(\mathbf{x})$ depends on a missing data vector $\boldsymbol{u}$, that observed would make the estimation of $\boldsymbol{\theta}$ easy. The usefulness of the algorithm is demonstrated with numerous examples in [64].

The algorithm is centered around the EM functional that minorizes $p_{\boldsymbol{\theta}}(\mathbf{x})$. The EM functional is constructed in the expectation (E) step of the algorithm. Then, in the maximization (M) step, the EM functional is maximized w.r.t. $\boldsymbol{\theta}$. This process is repeated until convergence. Issues relating to the convergence of the algorithm are discussed in [65] and [66].

### B. EM Algorithm for the Proposed Method

As stated in Section IV-A, the EM algorithm consists of performing the E- and the M-step of the algorithm in an iterative manner. In the following, we develop those steps for the LDA-svnPCA$_0$ algorithm. Note that this is an iterative algorithm, and we denote the current iterate of a parameter with a subscript 0 and the next iterate with a subscript 1.

The E-step consists of constructing the EM functional. First, it is necessary to construct the complete penalized discriminant function, which is the discriminant function for the case where the (missing) data $\mathcal{U} = \{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n\}$ is observed. The complete penalized discriminant function is (ignoring irrelevant terms)

$$J_{\boldsymbol{\theta}}(\mathcal{X}, \mathcal{U}) = -\frac{p}{2}\log\sigma^2 - \frac{1}{n}\sum_{k=1}^{K}\sum_{i \in C_k}\frac{\|\tilde{\mathbf{x}}_i - \boldsymbol{d}_k - \boldsymbol{G}\boldsymbol{u}_i\|^2}{2\sigma^2} - \frac{h}{2}\sum_{j=1}^{p}|||\boldsymbol{a}_j|||_{\boldsymbol{\theta}}.$$

The EM functional $\mathrm{EM}(\boldsymbol{\theta}_0, \boldsymbol{\theta}) = E_{\boldsymbol{\theta}_0}[J_{\boldsymbol{\theta}}(\mathcal{X}, \mathcal{U})|\mathcal{X}]$ is the expected value of the complete penalized discriminate function w.r.t.

$$p\left(\boldsymbol{u}_i | \tilde{\mathbf{x}}_i, \boldsymbol{G}_0, \sigma_0^2\right) = N\left(\boldsymbol{W}_0^{-1}\boldsymbol{G}_0^T(\tilde{\mathbf{x}}_i - \boldsymbol{d}_k), \sigma_0^2\boldsymbol{W}_0^{-1}\right)$$

and leads to

$$\mathrm{EM}(\boldsymbol{\theta}_0, \boldsymbol{\theta}) = -\frac{p}{2}\log\sigma^2 - \frac{1}{n}\sum_{k=1}^{K}\sum_{i \in C_k}\frac{\|\tilde{\mathbf{x}}_i - \boldsymbol{d}_k - \boldsymbol{G}\boldsymbol{u}_{i0}\|^2}{2\sigma^2} - \frac{\sigma_0^2\mathrm{tr}\left(\boldsymbol{G}\boldsymbol{W}_0^{-1}\boldsymbol{G}^T\right)}{2\sigma^2} - \frac{h}{2}\sum_{j=1}^{p}|||\boldsymbol{a}_j|||_0 \tag{14}$$

where $\boldsymbol{W}_0 = \boldsymbol{G}_0^T\boldsymbol{G}_0 + \sigma_0^2\boldsymbol{I}_r$, and $\boldsymbol{u}_{i0} = \boldsymbol{W}_0^{-1}\boldsymbol{G}_0^T(\tilde{\mathbf{x}}_i - \boldsymbol{d}_{k0})$.

In the M-step of the EM algorithm, we optimize the EM functional to obtain the update for the model parameter. The update formulas depend on the following quantities:

$$\begin{aligned}\boldsymbol{U} &= \left[\boldsymbol{u}_{i0}^T\right] \\ \boldsymbol{A}_0 &= \sigma_0^2 \boldsymbol{W}_0^{-1} + \frac{1}{n}\boldsymbol{U}^T\boldsymbol{U} \\ \boldsymbol{B}_0 &= \frac{1}{n}\tilde{\boldsymbol{X}}^T\boldsymbol{U}.\end{aligned}$$

The model parameter updates also depend on the following thresholding parameters:

$$\tau_j^2 = \boldsymbol{b}_{j0}^T \boldsymbol{A}_0^{-1}\boldsymbol{b}_{j0} + \sum_{k=1}^{K}\frac{n_k}{n}(\hat{\mu}_{kj} - \hat{\mu}_j)^2, \quad j = 1, \ldots, p$$

that are used to determine which variables are dropped from the model. The model parameter updates are given by (see the Appendix for a derivation)

$$\begin{aligned} d_{kj1} &= (\hat{\mu}_{kj} - \hat{\mu}_j) I\left(\tau_j^2 \geq h\sigma_0^2\right), \quad j = 1, \ldots, p \\ \boldsymbol{g}_{j1} &= \boldsymbol{A}_0^{-1}\boldsymbol{b}_{j0} I\left(\tau_j^2 \geq h\sigma_0^2\right), \quad j = 1, \ldots, p \\ \sigma_1^2 &= \frac{1}{p}\sum_{j\in\mathcal{I}}\left(S_{xjj} - \boldsymbol{b}_{j0}^T\boldsymbol{A}_0^{-1}\boldsymbol{b}_{j0}\right) + \frac{1}{p}\sum_{j\in\mathcal{I}^c} S_{xjj}\end{aligned}$$

where $S_{xjj}$ is the $j$th diagonal element of

$$\boldsymbol{S}_x = \frac{1}{n}\sum_{k=1}^{K}\sum_{i\in C_k}(\tilde{\mathbf{x}}_i - \boldsymbol{d}_{k1})(\tilde{\mathbf{x}}_i - \boldsymbol{d}_{k1})^T$$

and $\mathcal{I} = \{j : \tau_j^2 \geq h\sigma_0^2\}$, and $\mathcal{I}^c = \{j : \tau_j^2 < h\sigma_0^2\}$. For convenience, the algorithm is listed in Algorithm 1.

*Remark 1:* Note that due to the dependency of the $d_{kj}$ and the $\boldsymbol{g}_j$ updates with $\sigma_0^2$, the update equations need to be iterated to maximize the M-step. However, in our experiments, we found one iteration to be sufficient.

*Remark 2:* The EM algorithm depends on two tuning parameters $r$ and $h$. In this paper, cross-validation is used to select them. Refer to Section VI-B for details.

*Remark 3:* Due to the nonsmoothness of the vector $l_0$ penalty, the convergence theory in [65] and [66] does not apply. However, the convergence result in [55] does apply for the proposed EM algorithm.

**Algorithm 1:** The LDA-svnPCA$_0$ algorithm

**Input**: Data matrix $\tilde{\boldsymbol{X}}$, $C_1$,...,$C_K$, $r$, and $h$
**Initialization**: $\boldsymbol{G}_0$, $\hat{\boldsymbol{\mu}}$, $\boldsymbol{d}_{10}$,...,$\boldsymbol{d}_{K0}$, and $\sigma_0^2$
**while** *(Not converged)* **do**
  $\boldsymbol{W}_0 = \boldsymbol{G}_0^T\boldsymbol{G}_0 + \sigma_0^2\boldsymbol{I}_r$
  **for** $k = 1, \ldots, K$ **do**
    **for** $i \in C_k$ **do**
      $\boldsymbol{u}_{i0} = \boldsymbol{W}_0^{-1}\boldsymbol{G}_0^T(\tilde{\mathbf{x}}_i - \boldsymbol{d}_{k0})$
  $\boldsymbol{U} = [\boldsymbol{u}_{i0}^T]$
  $\boldsymbol{A}_0 = \sigma_0^2\boldsymbol{W}_0^{-1} + \frac{1}{n}\boldsymbol{U}^T\boldsymbol{U}$
  $\boldsymbol{B}_0 = \frac{1}{n}\tilde{\boldsymbol{X}}^T\boldsymbol{U}$
  **for** $j = 1, \ldots, p$ **do**
    $\tau_j^2 = \boldsymbol{b}_{j0}^T\boldsymbol{A}_0^{-1}\boldsymbol{b}_{j0}^T + \sum_{k=1}^{K}(\hat{\mu}_{kj} - \hat{\mu}_j)^2$
    $d_{kj1} = (\mu_{kj} - \hat{\mu}_j)I(\tau_j^2 \geq h\sigma_{j0}^2)$
    $\boldsymbol{g}_{j1} = \boldsymbol{A}_0^{-1}\boldsymbol{b}_{j0}I(\tau_j^2 \geq h\sigma_{j0}^2)$
    $S_{xjj} = \frac{1}{n}\sum_{k=1}^{K}\sum_{i\in C_k}\|\tilde{\mathbf{x}}_i - \boldsymbol{d}_{k1}\|^2$
  $\boldsymbol{G}_1 = [\boldsymbol{g}_{j1}^T]$
  $\boldsymbol{d}_{k1} = [d_{kj1}]$
  $\mathcal{I} = \{j : \tau_j^2 \geq h\sigma_0^2\}$
  $\mathcal{I}^c = \{j : \tau_j^2 < h\sigma_0^2\}$
  $\sigma_1^2 = \frac{1}{p}\sum_{j\in\mathcal{I}}\left(S_{xjj} - \boldsymbol{b}_{j0}^T\boldsymbol{A}_0^{-1}\boldsymbol{b}_{j0}\right) + \frac{1}{p}\sum_{j\in\mathcal{I}^c}S_{xjj}$
**Output**: $\hat{\boldsymbol{G}}$, $\hat{\boldsymbol{d}}_1, \ldots, \hat{\boldsymbol{d}}_K$, and $\hat{\sigma}^2$.

*Remark 4:* The computational complexity for an iteration of Algorithm 1 is only $O(prn)$. This does not account for possible savings due to sparsity, i.e., if there are only $p_h < p$ nonzero columns of $\boldsymbol{G}_0$, then the complexity becomes $O(p_h rn)$. There are no extra memory constraints apart from storing the data and the iterates. In fact, if storing the data matrix is a problem due to its size, it is easy to construct a sequential version of this algorithm, i.e., process one sample (row of $\tilde{\boldsymbol{X}}$) at a time.

### C. Special Case: Two Classes and $r = 0$

When $r = 0$, then $\boldsymbol{G} = \boldsymbol{0}$ and the LDA-svnPCA$_0$ method reduces to a diagonal covariance LDA with automatic selection of variables. Assuming we have $n_1$ samples from class 1 and $n_2$ samples from class 2, then the variables included in the classifier satisfy the condition $\tau_j^2 \geq h\sigma_0^2$ where

$$\tau_j^2 = \frac{n_1}{n}(\hat{\mu}_{1j} - \hat{\mu}_j)^2 + \frac{n_2}{n}(\hat{\mu}_{2j} - \hat{\mu}_j)^2$$

and now, since $n = n_1 + n_2$ and $\hat{\mu}_j = (n_1/n)\hat{\mu}_{1j} + (n_2/n)\hat{\mu}_{2j}$, we can write

$$\tau_j^2 = \frac{(\hat{\mu}_{1j} - \hat{\mu}_{2j})^2}{\left(\frac{1}{n_1} + \frac{1}{n_2}\right)}.$$

Therefore, it can be seen that the variable $j$ is selected based on whether it satisfies the following threshold:

$$|\mathcal{T}_j| = \frac{|\hat{\mu}_{1j} - \hat{\mu}_{2j}|}{\hat{\sigma}\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} \geq h.$$

This is the classical two-sample t-test [67]. Hence, in this special case, the LDA-svnPCA$_0$ algorithm consists of first selecting variables that pass a threshold defined by the two-sample t-test, and then performing diagonal covariance LDA on the retained variables.

## V. STRUCTURAL MRI DATASETS

The MRI data used in this paper were collected by deCODE Genetics[1] in a study on how rare CNVs conferring high risk of schizophrenia and/or other neurodevelopmental disorders affect the physiological function for an otherwise healthy brain. The MRI data were generated for healthy controls carrying neuropsychiatric CNVs, controls carrying CNVs not known to be associated with psychiatric disorders (Other CNVs), and controls without large CNVs (NoCNV) (see [22] for more detailed description of the recruitment and phenotyping). The MRI was collected with a 1.5 T whole body Philips Achieva scanner. The scans were performed with a sagittal 3-D fast $T_1$-weighted gradient echo sequence (TR = 8.6 ms, TE = 4.0 ms, flip angle = 8°, slice thickness = 1.2 mm, field of view = 240 × 240 mm$^2$).

### A. Segmentation

Each of the $n$ $p$-dimensional $T_1$ weighted (vectorized) MRIs were tissue segmented into white-matter and gray-matter images using the VBM8 software (http://dbm.neuro.uni-jena.de), which is integrated into the SPM8 software (Wellcome Department of Cognitive Neurology, Institute of Neurology, London, UK, http://www.fil.ion.ucl.ac.uk/spm) implemented in MATLAB R2014 (Mathworks Inc., Sherborn, MA, USA). Note that the segmentation does not change the dimensionality of the resulting images, e.g., a white-matter segmented image is still $p$-dimensional. After the segmentation, a spatial normalization step was performed that uses the DARTEL algorithm [68]to register the tissue segments into a common coordinate system. Good spatial normalization will tighten class clusters and reduce dimensionality. Finally, the maps from the normalization step were modulated, i.e., intensity corrected for local volume changes during spatial normalization. No spatial smoothing was applied.

### B. Feature Selection

Feature selection is a simple preliminary screening of variables that is often a helpful first step when dealing with $p \gg n$ data. Many effective and well known feature selection techniques have been proposed in recent years, such as the Dantzig selector [69], the Lasso [70], adaptive Lasso [71], SCAD [72], minimax concave penalties [73], and the locally weighted least squares regression methods [74]. However, these techniques are often not suitable for very high-dimensional data due to the unfeasible computational cost when the number of dimensions becomes very high, and finding the optimal model is not guaranteed.

The method we have chosen to employ here is called Sure Independence Screening (SIS) [75]. A brief overview of SIS can be given using the 2-class classification problem. Let $\boldsymbol{X}$ be an $n \times p$ data matrix, where each column $\mathbf{x}_{(i)}$ has been standardized with zero mean and unit variance, and let $\boldsymbol{c}$ be the associated $n \times 1$ label vector. A component-wise regression is performed, yielding the $p$-vector

$$\boldsymbol{\omega} = \boldsymbol{X}^T\boldsymbol{c}.$$

This is obviously a very computationally cheap operation. The vector $\boldsymbol{\omega}$ is basically the correlation coefficients of each feature with the label vector $\boldsymbol{c}$. The next step is to sort the magnitudes of $\boldsymbol{\omega}$ in a decreasing order and then select the first $m$ values of the sorted $\boldsymbol{\omega}$ as our features. Now suppose we have $n_1$ samples from class 1 and $n_2$ samples from class 2. The component-wise regression estimate $\boldsymbol{\omega}$ can then be written as

$$\boldsymbol{\omega} = \sum_{i \in C_1} \mathbf{x}_{(i)} - \sum_{i \in C_2} \mathbf{x}_{(i)}$$

where the $j$th component of $\boldsymbol{\omega}$ is given by

$$\omega_j = \frac{n_1\hat{\mu}_{1j} - n_2\hat{\mu}_{2j}}{\hat{\sigma}_j}$$

where $\hat{\mu}_{kj}$ is $j$th element of the centroid for class $k \in \{1, 2\}$, and $\hat{\sigma}_j$ is the standard deviation of the $j$th feature. When the classes are of equal size, i.e., $n_1 = n_2$, then each value of $\boldsymbol{\omega}$ is simply a scaled version of the two-sample t-test for the corresponding feature.

## VI. EXPERIMENTAL RESULTS

In this section, the proposed method is compared to state-of-the-art classification methods for big data problems. We perform several experiments using both simulated and real data. There are two simulated datasets. In the first dataset, the covariance matrix has an independent structure, i.e., $\boldsymbol{\Sigma} = \boldsymbol{I}_p$, and the only difference

[1]deCODE Genetics/Amgen, Reykjavik, Iceland.

between the classes is in their centroids. The second simulated dataset is generated according to the nPCA model (9), and thus has a more complex covariance structure than the first simulated dataset. Again, the difference between the classes is in their centroids.

The first real dataset is the Golub dataset [8], which is a gene expression dataset that consists of two leukemia classes and 7129 genes. In this experiment, the objective is to correctly classify samples into two groups that represent different types of leukemia. We include this dataset in our experiments to show that our method is able to handle microarray data.

Finally, we perform two experiments using the structural MRI data, which was described in Section V. In the first MRI experiment, we classify samples based on their structural MRI data into two groups. One group consists of control samples (NoCNV), while the second group has neuropsychiatric CNVs affecting cognition. The second MRI experiment involves classifying control samples into two groups that are defined by high and low polygenic risk scores (PRSs) for schizophrenia. The group with high PRS has a higher risk of developing schizophrenia, while the low-PRS group has a lower risk.

### A. Comparison Methods

The methods that are used for comparison are the shrunken centroid regularized discriminant analysis (SCRDA) method [45], the nearest shrunken centroid (NCS) method [76], [77], the penalized LDA method [49], and the linear SVM. All the comparison methods were implemented in the R statistical computing environment [78].

The linear SVM has a single tuning parameter $C$ that controls the smoothness of the SVM solution. The classification accuracies were insensitive to the choice of $C$, so the parameter was set equal to 1 in all experiments. The SVM was implemented using the R package e1071 [79], which is an interface for R to the popular LIBSVM software [80].

The SCRDA method avoids the singularity problem of the covariance estimate $\hat{\mathbf{\Sigma}}$ arising when $p \gg n$ by using a regularized covariance estimate (8). Another important feature of the SCRDA is that the class centroids are shrunken towards the overall centroid, and thus zeroing or eliminating features that do not contribute to the classification. The SCRDA method depends on two tuning parameters that control the sparsity and the covariance regularization, respectively. Both tuning parameters are chosen using cross-validation (CV). The implementation used for this method can be found in the R package RDA [81].

The NSC method is essentially a simplified version of SCRDA where the covariance is modeled as a positive definite diagonal matrix. It has a single tuning parameter that directly controls the sparsity of the classifier. As before, the optimal value of the tuning parameter was chosen based on CV. The implementation of NSC is given in the R package PAMR [82].

The penalized LDA method in [49] is a method that addresses the shortcomings of the classical LDA for $p \gg n$ problems, i.e., the within-class covariance matrix becomes singular and the interpretability of the classifier is low when the features are many. The method uses a diagonal estimate of the within-class covariance and applies an $\ell_1$ penalty to the discriminant vectors, which is controlled by the tuning parameter $\lambda$. We used 30 values for the sparsity tuning parameter $\lambda$, and the optimal value was chosen based on CV. This method is implemented in the R package penalizedLDA [83]. The methods used in the experiments are summarized in Table 1.

All the comparison methods, except the linear SVM, are based on LDA. When the number of features is very large, as is typical for $p \gg n$ problems, the interpretability of the algorithms is reduced since the classifier output depends on all the $p$ features. All the comparison LDA-based methods have the ability to drop variables that do not contribute to the classification via some sparsity penalty, i.e., $\ell_1$-penalty, shrunken centroids with covariance regularization, etc. Our method models the covariance using the nPCA model and via the $\ell_0$-penalty penalty on the variables, has the ability to drop irrelevant variables or features. This, coupled with the SIS feature selection helps to further increase the interpretability of the method and make the representation of the results more compact since only a small fraction of the original $p$ variables is retained in the trained model.

### B. Parameter Selection and General Experimental Procedure

Our method depends on two tuning parameters, the sparsity parameter, $h$, and the number of nPCs, $r$, that need to be selected. Here, we describe the procedure

Table 1 Methods Used in the Experiments

| | |
|---|---|
| Comparison methods | |
| SVM | SVM with a linear kernel |
| SCRDA | Shrunken centroids regularized LDA |
| NSC | Nearest shrunken centroids (NSC) LDA |
| PLDA | Penalized sparse LDA using $\ell_1$ penalty |
| Proposed method | |
| LDA-svnPCA0 | Penalized sparse LDA-svnPCA$_0$ based on svnPCA |

Table 2 Performance Metrics Used in All Experiments

| Name | meaning |
|---|---|
| CV err | The minimum CV test error |
| Nonzeros | The min. # of features obtained for optimal $TE_{opt}$ |
| TE | min test error (TE) obtained for optimal param. values |
| $TE_{opt}$ | min TE obtained over whole tuning parameter space |

used to select them. There are many tuning parameter selection methods in the literature such as AIC [84], BIC [85], and the extended BIC [86]. However, the tuning parameter method most often used in classification problems is CV [87].

CV is a technique to assess the prediction error of a model. There are many variants of CV methods, and the one used in this paper is called $L$-fold CV. The basic idea behind $L$-fold CV is to split the training data into $L$ roughly equally sized partitions or folds. One fold is kept for validation, and the classifier is trained using the remaining $L-1$ folds, i.e., the classifier is trained on $L-1$ folds of the data and tested on the $l$th fold, where $l = 1, 2, \ldots, L$. Finally, the sum of the prediction error for all the folds is used to obtain the CV estimate of the prediction error. By repeating this for each value of the tuning parameters, one can assign a CV prediction error estimate to them and thus use the CV estimate to choose the tuning parameters.

For the simulated data and Golub data, 10-fold CV is used, while for the MRI data, 5-fold CV is used. The Golub dataset is provided with predefined training and testing sets, while for the MRI data, we use two thirds of the available data for CV to select the tuning parameters and one third of the data for validation.

It is important to note that many parameter values can yield the same CV test error, and thus the question remains how to select the tuning parameter(s) that give the optimal test results. In this paper, the four metrics given in Table 2 are used to assess the performance of the various methods. The first metric is the lowest CV test error; the second, denoted by Nonzeros, is the minimum number of features obtained for $TE_{opt}$ (see below). The third metric, denoted by TE, is the minimum test error obtained for all the tuning parameter values that give rise to the same minimum CV test error. Finally, the fourth metric, which is denoted by $TE_{opt}$, is the minimum test error obtained by training the classifier using the entire training set and iterating over the whole tuning parameter space. Note that $TE_{opt}$ can be regarded as an indicator of the optimal performance of the classifier, given the data and predefined tuning parameter values. The TE metric depends entirely on the method used to choose the optimal tuning parameters, and there are many methods to do that. Hence, $TE_{opt}$ is more informative regarding how well a given classifier actually performs.

SIS feature selection is used to reduce the amount of features prior to classification using the proposed method, typically by a factor of hundred and, in some cases, by a factor of thousand. SIS is not used for the comparison methods, primarily since they are self-contained and complete methods. Also, feature selection does not work well with linear SVM. It can actually degrade the performance of the classifier [88]. Thus, we consider the application of SIS as an important part of the proposed method.

### C. Simulated Data

The first simulated dataset has two classes of multivariate Gaussian distribution with the same covariance, i.e., $\boldsymbol{\Sigma} = \boldsymbol{I}_p$, while the second simulated dataset is based on the nPCA model in (9) with the number of nPCs, $r$, set to 5. The only difference between the two classes lies in the centroids for the distributions.

*1) Simulation One:* There are two classes of $N(\boldsymbol{\mu}_1, \boldsymbol{I}_p)$ and $N(\boldsymbol{\mu}_2, \boldsymbol{I}_p)$ distributed independent variables of dimension $p = 10\,000$ where $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ are all zeros, except that the first 100 components of $\boldsymbol{\mu}_2$ are equal to 0.5. Here, we are essentially generating data according the nPCA model (9) with $r = 0$. There are 100 training samples and 500 testing samples generated for each class.

The experiment is repeated 50 times, and the values of the performance metrics CV err, TE, and $TE_{opt}$, are given as the mean values of all the trials along with the standard deviation. We chose not to use SIS feature reduction in the simulations.

The results are shown in Table 3, where all the methods give relatively good test accuracies, except SVM, which performs considerably worse than the other methods. The SCRDA and NSC methods perform very similarly, while the LDA-svnPCA$_0$ method, with $r = 0$, has the lowest mean $TE_{opt}$ value. The PLDA method

Table 3 Simulation One With $\boldsymbol{\Sigma} = \boldsymbol{I}_p$. The Standard Deviation is Given in Parentheses

| Method | CV err | TE | $TE_{opt}$ |
|---|---|---|---|
| SVM | N/A | 217.9/1000 (12.1) | 217.9/1000 (12.13) |
| SCRDA | 6.6/200 (2.8) | 33.8/1000 (8.0) | 31.0/1000 (7.0) |
| NSC | 6.9/200 (2.5) | 33.7/1000 (8.0) | 30.4/1000 (6.7) |
| PLDA | 12.1/200 (3.8) | 49.0/1000 (9.8) | 43.5/1000 (11.2) |
| LDA-svnPCA$_0$ | 6.1/200 (2.7) | 34.5/1000 (10.5) | 29.6/1000 (7.2) |

Table 4 Simulation Two With Data Simulated Using nPCA Model (9). The Standard Deviation is Given in Parentheses

| Method | CV err | TE | $TE_{opt}$ |
|---|---|---|---|
| SVM | N/A | 434/1000 (12.9) | 434/1000 (12.9) |
| SCRDA | 47.7/200 (6.0) | 222.1/1000 (46.6) | 212.2/1000 (46.2) |
| NSC | 72.3/200 (5.90) | 390.16/1000 (28.8) | 370.68/1000 (22.5) |
| PLDA | 80.2/1000 (8.3) | 427.1/1000 (18.3) | 414.0/1000 (14.2) |
| LDA-svnPCA$_0$ | 6.7/200 (2.92) | 22.4/1000 (8.36) | 19.8/1000 (4.8) |

performs worse than the other methods, with the exception of the linear SVM. The N/A for the SVM method in the table is because for the linear SVM we did not perform CV to determine the weight tuning parameter $C$ since its value turned out to be irrelevant.

*2) Simulation Two:* This simulation is similar to the previous one with the exception that now the data is correlated according to the nPCA covariance matrix $\boldsymbol{\Omega} = \boldsymbol{G}\boldsymbol{G}^T + \sigma^2\boldsymbol{I}_p$. We set $r = 5$ and use $p = 10\,000$ features as before. Thus, sample $i$ from class $k$ is generated by using the nPCA model (9), $\mathbf{x}_i = \boldsymbol{\mu}_k + \boldsymbol{G}\boldsymbol{u}_i + \boldsymbol{\epsilon}_i$, where the mean vectors $\boldsymbol{\mu}_k$ are the same as in Simulation 1. Only the first 100 rows of $\boldsymbol{G}$ are nonzero. Note that $\boldsymbol{G}$ is a $p \times r$ random matrix where we have chosen $r = 5$. We use 50 trials while keeping $\boldsymbol{G}$ fixed. The number of training samples is 200, and the number of test samples is 1000. The results for this experiment are summarized in Table 4.

The comparison methods perform poorly here, especially the SVM, PLDA, and NSC methods, but the proposed method performs very well, having a minimum test error $TE_{opt}$ of an order of magnitude smaller than for the other methods.

### D. Real Microarray Data. The Golub Dataset

This cancer microarray dataset [89] consists of 7129 gene expressions for 47 subjects having acute lymphoblastic leukemia (ALL) and 25 subjects who have acute myeloid leukemia (AML). The samples are divided into 38 training samples and 34 test samples, giving a total of 72 samples with 7192 features (genes). The objective here is to correctly classify the samples to either the ALL group or the AML group.

The results are given in Table 5. All the methods have a low misclassification rate for this dataset. The SVM method performs worst with five misclassified samples, and the PLDA method is slightly better with four misclassifications, while the SCRDA and LDA-svnPCA$_0$ methods give excellent results with $TE_{opt} = 0$ and $TE = 1$.

Table 5 Misclassification Results for the Golub Dataset. The Parentheses After LDA-svnPCA$_0$ Indicate What $r$ was Used for White and Gray Matter, Respectively

| Method | CV err | Nonzeros | TE | $TE_{opt}$ |
|---|---|---|---|---|
| SVM | N/A | N/A | 5/34 | 5/34 |
| SCRDA | 0/38 | 66 | 1/34 | 0/34 |
| NSC | 1/38 | 3334 | 1/34 | 1/34 |
| PLDA | 2/38 | 1311 | 4/34 | 4/34 |
| LDA-svnPCA$_0$ (2) | 1/38 | 404 | 1/34 | 0/34 |

### E. Neuropsychiatric CNVs Affecting Cognition

In this experiment, gray- and white-matter segmented MRI data is used to classify subjects belonging to three groups, i.e., to classify a control group from two groups of subjects having neuropsychiatric CNVs that affect cognition. The control group (NoCNV) contains subjects with no large CNVs (NoCNV). The first group of subjects that have CNVs that affects cognition is subjects with 16p13.1 duplication. The second group of subjects with CNVs is subjects with 22q11.2 duplication.

The 16p13.1 duplication group contains subjects that have a DNA segment duplicated at chromosome 16 at location (locus) p13.11 and the 22q11.2 duplication CNV has a DNA segment duplicated at locus q11.2. In [22], it is shown that controls carrying those CNVs perform considerably worse than controls having no CNV.

The groups used for the 16p13.1 and 22q11.2 duplication experiments are summarized in Tables 6 and 7, respectively. Both duplication groups have 18 subjects, and they are age-matched with the NoCNV group. We used two thirds of the data for training and one third for testing, so there are 24 subjects in the training set and 12 in the test set. We randomly assigned six males and six females out of the 18 subjects to each class in the training set. We used 5-fold CV to choose the optimal training parameters, and we chose the folds such that the gender ratio was equal or very similar in each fold, i.e., stratified CV. The experiment was performed for both types of matter, white and gray.

*1) Results for 16p13.1 Dup:* The results for the NoCNV vs. 16p13.1 dup classification experiment are shown in Table 8. The performance of the methods varies

Table 6 Summary of Groups NoCNV and 16p13.1 Dup. The Groups Have Been Matched for Age

| group | mean age | sd(age) | males | females |
|---|---|---|---|---|
| NoCNV | 44.39 | 12.94 | 7 | 11 |
| 16p13.1 dup | 44.39 | 12.94 | 9 | 9 |

Table 7 Summary of Groups NoCNV and 22q11.2 Dup. The Groups Have Been Matched for Age

| group | mean age | sd(age) | males | females |
|---|---|---|---|---|
| NoCNV | 42.67 | 13.11 | 7 | 11 |
| 22q11.2 dup | 42.67 | 13.11 | 10 | 8 |

considerably for both types of matter. For white matter, the LDA-svnPCA$_0$ and SCRDA methods perform best with zero misclassifications for the TE$_{opt}$ metric. However, LDA-svnPCA$_0$ performs better than SCRDA in terms of the TE metric. The SVM method performs worst in terms of TE$_{opt}$, and the PLDA method performs second worst.

Considering the gray-matter results, we see that NSC, SCRDA, and the proposed method give the same results of 83.3% accuracy for TE$_{opt}$, while the SCRDA method has 50% accuracy in TE. SVM performs worst with 75% TE and TE$_{opt}$ accuracy, and PLDA is second worst with 66.7% accuracy for both TE and TE$_{opt}$. The LDA-svnPCA$_0$ method turns out to have the best TE score. It seems that the SCRDA and NSC methods have trouble finding the optimal tuning parameters using CV.

Using SIS to reduce features approximately 100-fold down to 5000 is shown to be beneficial. For white matter, the LDA-svnPCA$_0$ method has a perfect TE$_{opt}$ of zero. For gray matter, the LDA-svnPCA$_0$ method also benefits from the reduction of features, achieving perfect results, i.e., 100% accuracy for both the TE metrics. Fig. 2 shows the regions (voxels) of the brain that turned out to be the most discriminating between the NoCNV and 16p13.1 dup groups, according to the best LDA-svnPCA$_0$ result for white matter in Table 8.

*2) Results for 22q11.2 Dup:* For the NoCNV vs. 22q11.2 dup classification experiment, the results are summarized in Table 9. For white matter, the SVM and PLDA methods have TE and TE$_{opt}$ values of 4 out of 12, which is 66.7% classification accuracy. The LDA-svnPCA$_0$ and both the SCRDA and NSC methods have a TE$_{opt}$ value of 1, i.e., 91.7% accuracy. Again, the LDA-svnPCA$_0$ method has the lowest TE score, by a considerable margin.

For the gray matter, the LDA-svnPCA$_0$ and NSC methods are the best performers considering both TE and TE$_{opt}$. SIS feature selection failed to improve the LDA-svnPCA$_0$ method, except that TE was improved for white matter. Once again, we see PLDA and SVM performing similarly. Fig. 3 shows the regions (voxels) of the brain that turned out to be the most discriminating between the NoCNV and 22q11.2 dup groups, according to the best LDA-svnPCA$_0$ result for gray matter in Table 9.

In summary, for both NoCNV vs. CNV duplication experiments, the proposed method clearly outperforms the other state-of-the-art comparison methods. The results obtained by the proposed method indicate that there are indeed significant structural differences in the brain between the NoCNV group and the CNV duplication groups.

### F. Polygenic Risk Scores for Schizophrenia

The aim of this final experiment is to determine if high or low PRS values for schizophrenia result in structural morphology of white or gray matter, which can be used for classification purposes. PRS is defined as [90]

$$\hat{S} = \sum_{i=1}^{m} \hat{\beta}_i g_i.$$

It is a linear combination of $m$ schizophrenia-associated alleles $g_i \in \{0, 1, 2\}$ weighted by effect size $\hat{\beta}_i$. The number $m$ is selected based on some threshold on the $P$-value of the effect size, typically $P < 0.1$.

For this study, we have structural MRI data for 54 control subjects with no large CNV, for which a PRS value based on the subject's genome has been calculated. There are two groups: low PRS, which consists of 27 subjects from the lower 5% tail of the PRS population distribution, and high PRS, which consists of 27 subjects from the upper 5% tail of the PRS population distribution. The odds ratio for schizophrenia in the low PRS group is 0.28, while the odds ratio for schizophrenia for the high PRS group is 4.63. We use the same procedure as in the

Table 8 NoCNV vs 16p13.1 Dup Misclassification Results. The Parentheses After LDA-svnPCA$_0$ Indicate What $r$ was Used for White and Gray Matter, Respectively

| | White matter | | | | Gray matter | | | |
|---|---|---|---|---|---|---|---|---|
| Method | CV err | Nonzeros | TE | TE$_{opt}$ | CV err | Nonzeros | TE | TE$_{opt}$ |
| SVM | N/A | N/A | 4/12 | 4/12 | N/A | N/A | 3/12 | 3/12 |
| SCRDA | 10/24 | 435 | 6/12 | 0/12 | 8/24 | 11439 | 6/12 | 2/12 |
| NSC | 8/24 | 3 | 6/12 | 1/12 | 6/24 | 26 | 2/12 | 2/12 |
| PLDA | 14/24 | 0 | 6/12 | 2/12 | 11/24 | 95024 | 4/12 | 4/12 |
| LDA-svnPCA$_0$ (2,2) | 7/24 | 3584 | 1/12 | 0/12 | 7/24 | 230 | 2/12 | 2/12 |
| SIS (5000) | | | | | | | | |
| LDA-svnPCA$_0$ (3,1) | 6/24 | 74 | 1/12 | 0/12 | 8/24 | 83 | 0/12 | 0/12 |

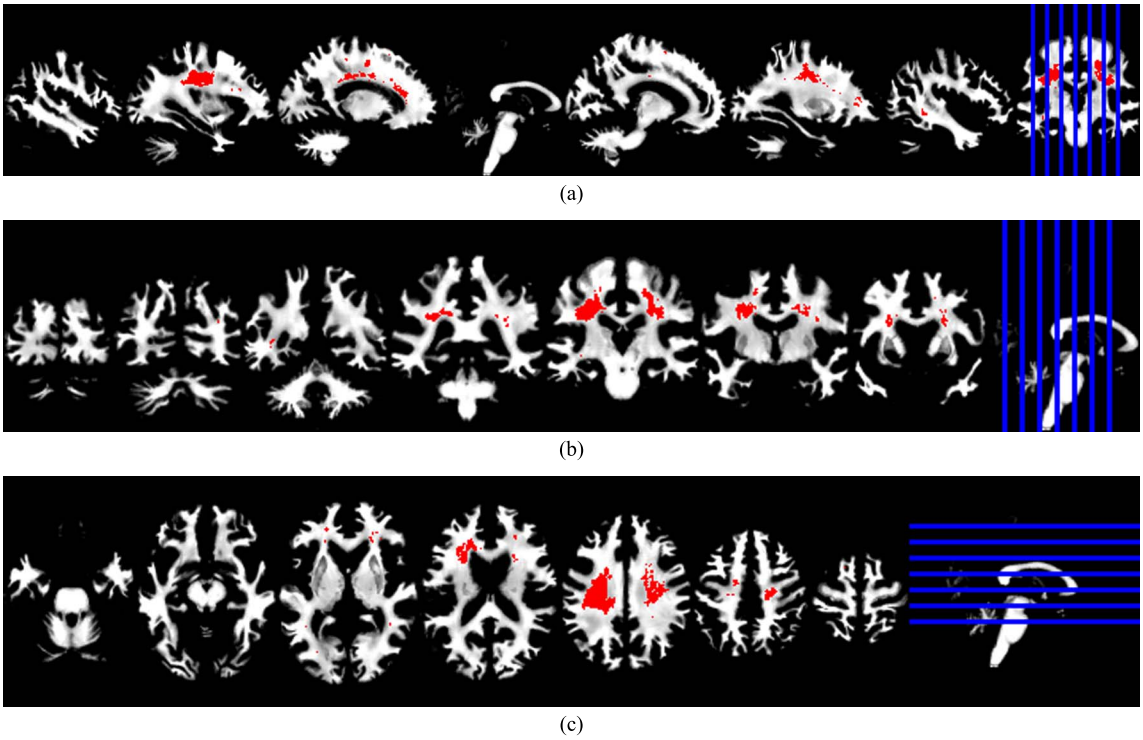


**Fig. 2.** *Discriminating voxels (red) for the LDA-svnPCA$_0$ method for the NoCNV vs. 16p13.1 dup experiment (white matter). The vertical and horizontal bars on the pictures to the right indicate what brain slices are plotted to the left. (a) Sagittal. (b) Coronal. (c) Axial.*

NoCNV vs. CNV duplication experiments to eliminate gender bias, and we also split the data into training and testing datasets using the same proportions. Table 10 shows a summary of the two classes, while the results for this experiment are summarized in Table 11.

Considering the results in Table 11, it is readily seen that these data are much more challenging than the NoCNV vs. duplication data. For white matter, all the comparison methods are performing at 50% accuracy or even worse. The LDA-svnPCA$_0$ method achieves an TE$_{opt}$ score of 7 out of 18. However, it is quite obvious that it is difficult to obtain optimal tuning parameter values for LDA-svnPCA$_0$ using CV for this dataset.

For gray matter, the results are somewhat similar. Again the comparison methods fail to find any discriminating features, while the LDA-svnPCA$_0$ method has a TE$_{opt}$ value of 5 out of 18, which translates to roughly 72% accuracy, which is very acceptable, especially when given the poor performance of the comparison methods. As for the white matter, the high TE values are in stark contrast to the lower TE$_{opt}$ values. By using SIS to reduce the number of features down to 5000, we see that for white matter, the optimal misclassification rate goes down from 7 out of 18 to 5 out of 18, which is quite good. However, for the gray matter, SIS did not improve the results.

Table 9 NoCNV vs 22q11.2 Dup Misclassification Results. The Parentheses After LDA-svnPCA$_0$ Indicate What $r$ was Used for White and Gray Matter, Respectively

| | White matter | | | | Gray matter | | | |
|---|---|---|---|---|---|---|---|---|
| Methods | CV err | Nonzeros | TE | TE$_{opt}$ | CV err | Nonzeros | TE | TE$_{opt}$ |
| SVM | N/A | N/A | 4/12 | 4/12 | N/A | N/A | 5/12 | 5/12 |
| SCRDA | 9/24 | 9 | 4/12 | 1/12 | 9/24 | 349893 | 6/12 | 4/12 |
| NSC | 7/24 | 145 | 4/12 | 1/12 | 9/24 | 7 | 5/12 | 3/12 |
| PLDA | 11/24 | 52536 | 4/12 | 4/12 | 14/24 | 104758 | 5/12 | 5/12 |
| LDA-svnPCA$_0$ (2,2) | 5/24 | 72 | 2/12 | 1/12 | 8/24 | 1960 | 4/12 | 2/12 |
| SIS (5000) | | | | | | | | |
| LDA-svnPCA$_0$ (3,3) | 9/24 | 83 | 4/12 | 0/12 | 7/24 | 11 | 5/12 | 4/12 |

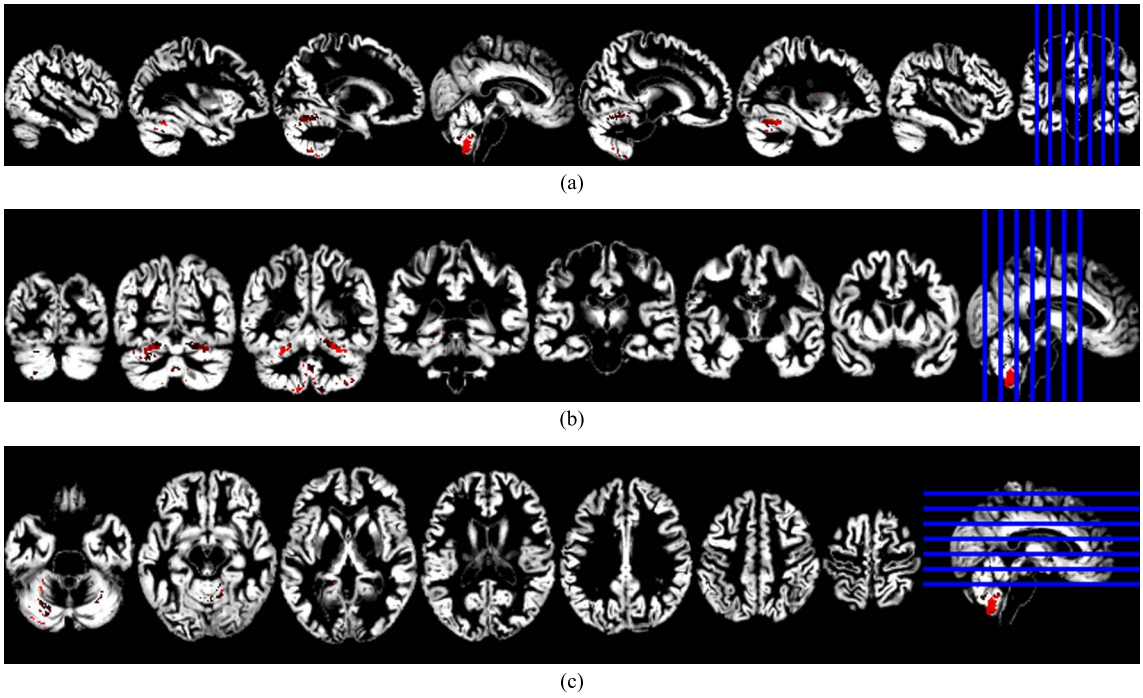


**Fig. 3.** *Discriminating voxels (red) for the LDA-svnPCA$_0$ method for the NoCNV vs. 22q11.2 dup experiment (gray matter). The vertical and horizontal bars on the pictures to the right indicate what brain slices are plotted to the left. (a) Sagittal. (b) Coronal. (c) Axial.*

To summarize the findings of this experiment, it is clear that the comparison methods totally fail to discriminate between the two classes. The proposed method can do so, with up to 72% accuracy, indicating that there are some detectable morphological differences in the brain between the low risk and high risk groups.

## VII. CONCLUSION

In this paper, we have considered the classification of big data where $p \gg n$. In particular, we have focused on classification of 3-D MRI images of the human brain in a genetic context.

To address this issue we have proposed a novel algorithm, based on LDA from the normal model viewpoint, where the estimation of the covariance of the observed data is performed by using the nPCA covariance model. Another important feature of the proposed method is its inherent ability to drop out variables that do not contribute to the class separation, using a vector $\ell_0$ penalty. The proposed method depends on two tuning parameters, i.e., the number of noisy principal components and a sparsity tuning parameter, which were selected based on the cross-validation. We conducted a number of experiments using simulated data, real microarray data, and structural MRI data of both white and gray matter.

The experiment results indicate that when dealing with data of very high dimensionality with complex covariance structure such as the MRI data, our method compares well to other state-of-the-art big data methods. An important item for future work is to develop methods, which are faster than cross-validation, for tuning parameter selection. ■

Table 10 Summary of the Low-Risk and High-Risk Groups in PRS Study

| | Low PRS | | High PRS | |
|---|---|---|---|---|
| | PRS | AGE | PRS | AGE |
| min | -78.81 | 25 | -65.24 | 22 |
| max | -74.41 | 63 | -63.00 | 66 |
| mean | -75.50 | 47.48 | -64.30 | 47.85 |
| sd | 1.02 | 10.91 | 0.68 | 11.01 |
| males | 14 | | 11 | |
| females | 13 | | 16 | |

## APPENDIX

By expanding the quadratic of (14) and combining terms, we can write

$$\mathrm{EM}(\boldsymbol{\theta}_0, \boldsymbol{\theta}) = -\frac{p}{2}\log\sigma^2 - \frac{\mathrm{tr}(\boldsymbol{S}_x)}{2\sigma^2} - \frac{\mathrm{tr}\left(\boldsymbol{G}\boldsymbol{B}_0^T\right)}{\sigma^2} - \frac{\mathrm{tr}(\boldsymbol{G}\boldsymbol{A}_0\boldsymbol{G}^T)}{2\sigma^2} - \frac{h}{2}\sum_{j=1}^{p} |||\boldsymbol{a}_j|||_0$$

Table 11 PRS Study—27 Subjects From Each Tail of the PRS Distribution. The Parentheses After LDA-svnPCA$_0$ Indicate What $r$ was Used for White and Gray Matter, Respectively

| | White matter | | | | Gray matter | | | |
|---|---|---|---|---|---|---|---|---|
| Methods | CV err | Nonzeros | TE | $TE_{opt}$ | CV err | Nonzeros | TE | $TE_{opt}$ |
| SVM | N/A | N/A | 9/18 | 9/18 | N/A | N/A | 9/18 | 9/18 |
| SCRD | 18/36 | 0 | 9/18 | 9/18 | 13/36 | 323173 | 11/18 | 8/18 |
| NSC | 14/36 | 0 | 10/18 | 10/18 | 14/36 | 15 | 9/18 | 9/18 |
| PLDA | 17/36 | 109933 | 10/18 | 9/18 | 18/36 | 0 | 9/18 | 9/18 |
| LDA-svnPCA$_0$ (1,1) | 15/36 | 196 | 11/18 | 7/18 | 16/36 | 179 | 11/18 | 5/18 |
| SIS (5000) | | | | | | | | |
| LDA-svnPCA$_0$ (1,3) | 17/36 | 6 | 10/18 | 5/18 | 15/36 | 60 | 11/18 | 6/18 |

where

$$\boldsymbol{W}_0 = \boldsymbol{G}_0^T\boldsymbol{G}_0 + \sigma_0^2\boldsymbol{I}_r$$

$$\boldsymbol{u}_{i0} = \boldsymbol{W}_0^{-1}\boldsymbol{G}_0^T(\tilde{\mathbf{x}}_i - \boldsymbol{d}_{k0}), \quad i = 1,\ldots,r$$

$$\boldsymbol{U} = \left[\boldsymbol{u}_{i0}^T\right]$$

$$\boldsymbol{A}_0 = \sigma_0^2\boldsymbol{W}_0^{-1} + \frac{1}{n}\boldsymbol{U}^T\boldsymbol{U}$$

$$\boldsymbol{B}_0 = \frac{1}{n}\tilde{\boldsymbol{X}}^T\boldsymbol{U}$$

$$\boldsymbol{S}_x = \frac{1}{n}\sum_{k=1}^{K}\sum_{i\in C_k}(\tilde{\mathbf{x}}_i - \boldsymbol{d}_k)(\tilde{\mathbf{x}}_i - \boldsymbol{d}_k)^T.$$

Maximization w.r.t. $\boldsymbol{G}$ is equivalent to minimization of the following cost function w.r.t. $\boldsymbol{G}$:

$$J_1 = \sum_{j=1}^{p} J(\boldsymbol{g}_j)$$

$$= \sum_{j=1}^{p}\left(\frac{1}{2}\boldsymbol{g}_j^T\boldsymbol{A}_0\boldsymbol{g}_j - \boldsymbol{g}_j^T\boldsymbol{b}_{j0} + \frac{h\sigma^2}{2}|||\boldsymbol{a}_j|||_0\right).$$

The cost function is separable in the rows of $\boldsymbol{G}$, so we optimize it for each $\boldsymbol{g}_j$ individually. Due to the $l_0$ penalty, the cost function is not differentiable at zero, so we proceed in two steps. First, we assume that $\boldsymbol{g}_j \neq \mathbf{0}$ and find the optimal solution, then we compare the resulting cost to the cost when $\boldsymbol{g}_j = \mathbf{0}$. Assuming that $\boldsymbol{g}_j \neq \mathbf{0}$ and differentiating and setting $J$ equal to zero yields

$$\boldsymbol{g}_j = \boldsymbol{A}_0^{-1}\boldsymbol{b}_{j0}.$$

A comparison to the $\boldsymbol{g}_j = \mathbf{0}$ solution yields

$$J(\boldsymbol{g}_j) - J(\mathbf{0}) = -\tau_j^2 + h\sigma^2 \geq 0$$

where

$$\tau_j^2 = \boldsymbol{b}_{j0}^T\boldsymbol{A}_0^{-1}\boldsymbol{b}_{j0} + \sum_{k=1}^{K}\frac{n_k}{n}(\hat{\mu}_{kj} - \hat{\mu}_j)^2.$$

The $\boldsymbol{g}_j = \mathbf{0}$ solution is picked if $h\sigma_0^2 > \tau_j^2\sigma^2$, and the minimizer is given by

$$\boldsymbol{g}_{j1} = \boldsymbol{A}_0^{-1}\boldsymbol{b}_{j0}I\left(\tau_j^2 \geq h\sigma_0^2\right), \qquad j = 1,\ldots,p.$$

By using the same optimization method, we can optimize the EM functional w.r.t. $d_{kj}$ (note that it depends on the EM functional through $\boldsymbol{S}_x$). The optimization yields

$$d_{kj1} = (\hat{\mu}_{kj} - \hat{\mu}_j)I\left(\tau_j^2 \geq h\sigma_0^2\right), \qquad j=1,\ldots,p, \;\; k=1,\ldots,K.$$

The optimization of the EM functional w.r.t. $\sigma^2$ easily yields

$$\sigma_1^2 = \frac{1}{p}\left[\mathrm{tr}(\boldsymbol{S}_x) - 2\mathrm{tr}\left(\boldsymbol{B}_0^T\boldsymbol{G}_1\right) + \mathrm{tr}\left(\boldsymbol{A}_0\boldsymbol{G}_1^T\boldsymbol{G}_1\right)\right]$$

which can be simplified to

$$\sigma_1^2 = \frac{1}{p}\sum_{j\in\mathcal{I}}\left(S_{xjj} - \boldsymbol{b}_{j0}^T\boldsymbol{A}_0^{-1}\boldsymbol{b}_{j0}\right) + \frac{1}{p}\sum_{j\in\mathcal{I}^c}S_{xjj}.$$

## REFERENCES


[1] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. New York, NY, USA: Springer, 2009.

[2] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford, U.K.: Clarendon Press, 1995.

[3] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, 2nd ed. New York, NY, USA: Wiley, 2001.

[4] L. Kaelbling, M. Littman, and A. Moore, "Reinforcement learning: A survey," *J. Artif. Intell. Res.*, vol. 4, pp. 237–285, 1996.

[5] M. Fauvel, Y. Tarabalka, J. Benediktsson, J. Chanussot, and J. Tilton, "Advances in spectral-spatial classification of hyperspectral images," *Proc. IEEE*, vol. 101, no. 3, pp. 652–675, Mar. 2013.

[6] Y. LeCun, L. Bottov, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[7] R. Chellappa, C. Wilson, and S. Sirohey, "Human and machine recognition of faces: A survey," *Proc. IEEE*, vol. 83, no. 5, pp. 705–741, May 1995.

[8] T. Golub *et al.*, "Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, no. 5439, pp. 531–537, 1999.

[9] F. Pereira, "Machine learning classifiers and fMRI: A tutorial overview," *Neuroimage*, vol. 45, no. 1, pp. S199–S209, 2009.

[10] S. Kloppel *et al.*, "Automatic classification of MR scans in Alzheimer's disease," *Brain*, vol. 131, no. 3, pp. 681–689, 2008.

[11] I. Gould *et al.*, "Multivariate neuroanatomized classification of cognitive subtypes in schizophrenia: A support vector machine learning approach," *Neuroimage: Clinical*, vol. 6, pp. 229–236, 2014.

[12] Y. Kawasaki *et al.*, "Multivariate voxel-based morphometry succesfully differentiates schizophrenia patients from healthy controls," *Neuroimage*, vol. 34, pp. 235–242, 2007.

[13] M. Niewenhuis *et al.*, "Classification of schizophrenia patients and healthy controls from structural MRI scans in two large independent samples," *Neuroimage*, vol. 61, pp. 606–612, 2012.

[14] H. Schnack *et al.*, "Can structural MRI aid in clinical classification? A machine learning study in two independent samples of patients with schizophrenia, bipolar disorder and healthy subjects," *Neuroimage*, vol. 84, pp. 299–306, 2014.

[15] E. Lander *et al.*, "Initial sequencing and analysis of the human genome," *Nature*, vol. 409, pp. 860–921, 2001.

[16] T. Jonsson *et al.*, "A mutation in APP protects against Alzheimer's disease and age related cognitive decline," *Nature*, vol. 488, pp. 96–99, 2012.

[17] D. Gudbjartsson *et al.*, "Many sequence variants affecting diversity of adult human height," *Nature Genetics*, vol. 40, pp. 609–615, 2008.

[18] A. Mayer-Lindenberg and D. Weinberger, "Intermediate phenotypes and genetic mechanism of psychiatric disorders," *Nat. Rev. Neurosci.*, vol. 10, pp. 818–827, 2006.

[19] J. Stein *et al.*, "Voxelwise genomewide association study (vGWAS)," *Neuroimage*, vol. 53, no. 3, pp. 1160–1174, 2010.

[20] M. Vounou, T. Nichols, and G. Montana, "Discovering genetic associations with high-dimensional neuroimaging phenotypes: A sparse reduced-rank regression approach," *Neuroimage*, vol. 53, pp. 1147–1159, 2010.

[21] M. Vounou *et al.*, "Sparse reduced-rank regression detects genetic associations with voxel-wise longitudinal phenotypes in Alzheimer's disease," *Neuroimage*, vol. 60, pp. 700–716, 2012.

[22] H. Stefansson *et al.*, "CNVs conferring risk of autism or schizophrenia affect cognition in controls," *Nature*, vol. 505, p. 361, 2014.

[23] J. Stein *et al.*, "Identification of common variants associated with human hippocampal and intracranial volume," *Nature Genetics*, vol. 44, pp. 552–561, 2012.

[24] D. Hilbar *et al.*, "Common genetic variants influence human subcortical brain structures," *Nature*, vol. 520, no. 7546, pp. 224–229, 2015.

[25] L. Breiman, "Random forests," *Mach. Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[26] J. Zhu and T. Hastie, "Classification of gene microarrays by penalized logistic regression," *Biostatistics*, vol. 5, pp. 427–443, 2004.

[27] V. Vapnik, *The Nature of Statistical Learning*. New York, NY, USA: Springer-Verlag, 1996.

[28] B. Scholkopf and A. Smola, *Learning With Kernels: Support Vector Machines, Regularization, Optimization, Beyond*. Cambridge, MA, USA: MIT Press, 2001.

[29] G. Wahba, Y. Lin, and H. Zhang, "GACV for support vector machines," in *Advances in Large Margin Classifiers*, A. Smola, P. Bartlett, B. Scholkopf, and D. Schuurmans, Eds. Cambridge, MA, USA: MIT Press, 2000.

[30] T. Hastie, S. Rosset, R. Tibshirani, and J. Zhu, "The entire regularization path for the support vector machine," *J. Mach. Learning Res.*, vol. 5, pp. 1391–1415, 2004.

[31] V. Solo, "Selection of tuning parameters for support vector machines," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Philadelphia, PA, USA, 2005, vol. 5, pp. v/237–v/240.

[32] O. Chapelle, "Training a support vector machine in the primal," *Neural Comput.*, vol. 19, no. 5, pp. 1155–1178, 2007.

[33] Y. Lee and O. Mangasarian, "Ssvm: A smooth support vector machine for classification," *Comput. Optim. Appl.*, vol. 20, pp. 5–22, 2001.

[34] T. Zhou, D. Tao, and X. Wu, "NESVM: A fast gradient method for support vector machines," in *Proc. IEEE Int Conf Data Mining*, Sydney, Australia, 2010, pp. 679–688.

[35] G. Fung and O. Mangasarian, "A feature selection Newton method for support vector machine classification," *Comput. Optim. Appl.*, vol. 28, no. 2, pp. 185–202, 2004.

[36] O. Mangasarian, "Exact 1-norm support vector machines via unconstrained convex differential minimization," *J. Mach. Learning Res.*, vol. 7, no. 2, pp. 1517–1530, 2006.

[37] L. Wang, J. Zhu, and H. Zou, "The doubly regularized support vector machine," *Statistica Sinica*, vol. 16, no. 2, pp. 589–615, 2006.

[38] H. Zhang, J. Ahn, X. Lin, and C. Park, "Gene selection using support vector machines with non-convex penalty," *Bioinformatics*, vol. 22, no. 1, pp. 88–95, 2006.

[39] Y. Liu, H. Zhang, C. Park, and J. Ahn, "Support vector machines with adaptive lq penalty," *Comput. Statist. Data Anal.*, vol. 51, no. 12, pp. 6380–6394, 2007.

[40] Y. Liu and Y. Wu, "Variable selection via a combination of the l0 and l1 penalties," *J. Comput. Graphical Statist.*, vol. 16, no. 4, pp. 782–794, 2007.

[41] K. Mardia, J. Kent, and J. Bibby, *Multivariate Analysis*. London, U.K.: Academic, 1979.

[42] R. Tibshirani, T. Hastie, B. Narasimka, and G. Chu, "Diagnosis of multiple cancer types by shrunken centroids of gene expression," *Proc. Nat. Acad. Sci.*, vol. 99, pp. 6567–6572, 2002.

[43] P. Bickel and E. Levina, "Some theory for Fisher's linear discriminant function, naive Bayes, some alternatives where there are more variables than observations," *Bernoulli*, vol. 10, no. 6, pp. 989–1010, 2004.

[44] H. Pang, T. Tong, and H. Zhao, "Shrinkage-based diagonal discriminant analysis and its application in high dimensional data," *Biometrika*, vol. 65, pp. 1021–1029, 2009.

[45] Y. Guo, T. Hastie, and R. Tibshirani, "Regularized linear discriminant analysis and its application in microarrays," *Biostatistics*, vol. 8, no. 1, pp. 86–100, 2007.

[46] R. Tibshirani, T. Hastie, N. Balasubramanian, and G. Chu, "Class prediction by nearest shrunken centroids, with application to DNA microarrays," *Statist. Sci.*, vol. 18, no. 1, pp. 104–117, 2003.

[47] R. Li and B. Wu, "Sparse regularized discriminant analysis with application to microarrays," *Comput. Biol. Chem.*, vol. 39, pp. 14–19, 2012.

[48] L. Clemmensen, T. Hastie, D. Witten, and B. Ersboll, "Sparse discriminant analysis," *Technometrics*, vol. 53, no. 4, pp. 406–413, 2011.

[49] D. Witten and R. Tibshirani, "Penalized classification using Fisher's linear discriminant," *J. Royal Statist. Soc., Ser. B, Statist. Methodol.*, vol. 73, no. 5, pp. 753–772, 2012.

[50] Y. Wu, D. Wipf, and J.-M. Yun, "Understanding and evaluating sparse linear discriminant analysis," in *Proc. 18th Int. Conf. Artif. Intell. Statist. (AISTATS)*, San Diego, CA, USA, 2015, pp. 1070–1078.

[51] M. Wax and T. Kailath, "Detection of signals by information theoretic criteria," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-33, no. 2, pp. 387–392, Apr. 1985.

[52] M. Ulfarsson and V. Solo, "Sparse variable PCA using geodesic steepest descent," *IEEE Trans. Signal Process.*, vol. 56, no. 12, pp. 5823–5832, Dec. 2008.

[53] D. Lawley, "A modified method of estimation in factor analysis and some large sample results," in *Proc. Uppsala Symp. Psychol. Factor Anal.*, Uppsala, Sweden, 1953, pp. 35–42.

[54] M. Tipping and C. Bishop, "Probabilistic principal component analysis," *J. Royal Statist. Soc., Ser. B*, vol. 61, no. 3, pp. 611–622, 1999.

[55] M. Ulfarsson and V. Solo, "Vector $l_0$ sparse variable PCA," *IEEE Trans. Signal Process.*, vol. 59, no. 5, pp. 1949–1958, May 2011.

[56] D. Bartholomew, *Latent Variable Models and Related Methods*. London, U.K.: Charles Griffin, 1987.

[57] J. Fan, Y. Liao, and M. Mincheva, "High-dimensional covariance matrix estimation in approximate factor models," *Ann. Statist.*, vol. 39, no. 6, pp. 3320–3356, 2011.

[58] F. Palsson, M. Ulfarsson, and J. Sveinsson, "Sparse Gaussian noisy independent component analysis," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Florence, Italy, 2014, pp. 4224–4228.

[59] A. Sereviratne and V. Solo, "On vector $l_0$ penalized multivariate regression," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Kyoto, Japan, 2012, pp. 3613–3616.

[60] F. Palsson, M. Ulfarsson, and J. Sveinsson, "Hyperspectral image denoising using a sparse low rank model and dual-tree complex wavelet transform," in *Proc. IEEE Int. Conf. Geosci. Remote Sens. (IGARSS)*, 2014, pp. 3670–3673.

[61] M. Luessi, M. Hamalainen, and V. Solo, "Vector $l_0$ latent-space principal component analysis," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Florence, Italy, 2014, pp. 4229–4233.

[62] B. Cassidy and V. Solo, "Spatially sparse, temporally smooth MEG via vector $l_0$," *IEEE Trans. Med. Imag.*, vol. 34, no. 6, pp. 1282–1293, Jun. 2015.

[63] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Royal Statist. Soc., Ser. B*, vol. 39, no. 1, pp. 1–38, 1977.

[64] T. Moon, "The expectation-maximization algorithm," *IEEE Signal Process. Mag.*, vol. 13, no. 6, pp. 47–60, Nov. 1996.

[65] C. Wu, "On the convergence properties of the EM algorithm," *Ann. Statist.*, vol. 11, pp. 95–103, 1983.

[66] K. Lange, "A gradient algorithm locally equivalent to the EM algorithm," *J. Royal Statist. Soc., Ser. B*, vol. 57, no. 2, pp. 425–437, 1995.

[67] G. Snedecor and W. Cochran, *Statistical Methods*. Ames, IA, USA: Iowa State Univ. Press, 1989.

[68] J. Ashburner and K. Friston, "Unified segmentation," *Neuroimage*, vol. 26, pp. 839–851, 2005.

[69] E. Candes and T. Tao, "The Dantzig selector: Statistical estimation when $p$ is much larger then $n$," *Ann. Statist.*, vol. 35, no. 6, pp. 2313–2351, 2007.

[70] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Royal Statist. Soc., Ser. B*, vol. 58, no. 1, pp. 267–288, 1996.

[71] H. Zou, "The adaptive Lasso and its oracle properties," *J. Amer. Statist. Assoc.*, vol. 101, no. 476, pp. 1418–1429, 2006.

[72] J. Fan and R. Li, "Variable selection via nonconcave penalized likelihood and its oracle properties," *J. Amer. Statist. Assoc.*, vol. 96, no. 456, pp. 1348–1360, 2001.

[73] C.-H. Zhang, "Nearly unbiased variable selection under minimax concave penalty," *Ann. Statist.*, vol. 38, no. 2, pp. 894–942, 2010.

[74] D. Ruppert and M. Wand, "Multivariate locally weighted least squares regression," *Ann. Statist.*, vol. 22, no. 3, pp. 1346–1370, 1994.

[75] J. Fan and J. Lv, "Sure independent screening for ultrahigh dimensional feature space," *J. Royal Statist. Soc., Ser. B, Statist. Methodol.*, vol. 70, no. 5, pp. 849–911, 2008.

[76] R. Tibshirani, T. Hastie, B. Narasimhan, and G. Chu, "Diagnosis of multiple cancer types by shrunken centroids of gene expression," *Proc. Nat. Acad. Sci.*, vol. 99, no. 10, pp. 6567–6572, 2002.

[77] R. Tibshirani, T. Hastie, B. Narasimhan, and G. Chu, "Class prediction by nearest shrunken centroids, with applications to DNA microarrays," *Statist. Sci.*, vol. 18, no. 1, pp. 104–117, 2003.

[78] R Core Team, "R: A language and environment for statistical computing," R Foundation for Statistical Computing, Vienna, Austria, 2015. [Online]. Available: http://www.R-project.org/

[79] D. Meyer, E. Dimitriadou, K. Hornik, A. Weingessel, and F. Leisch, "e1071: Misc Functions of the Department of Statistics (e1071), TU Wien," R Package Version 1.6-4, 2014. [Online]. Available: http://CRAN.R-project.org/package=e1071

[80] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, pp. 27:1–27:27, 2011. [Online]. Available: http://www.csie.ntu.edu.tw/cjlin/libsvm

[81] "Y. Guo, T. Hastie, and R. Tibshirani, RDA: Shrunken Centroids Regularized Discriminant Analysis, 2012, R Package Version 1.0.2-2. [Online]. Available: http://CRAN.R-project.org/package=rda

[82] T. Hastie, R. Tibshirani, B. Narasimhan, and G. Chu, "PAMR: PAM: Prediction analysis for microarrays," R Package Version 1.55, 2014. [Online]. Available: http://CRAN.R-project.org/package=pamr

[83] D. Witten, "penalizedLDA: Penalized classification using Fisher's linear discriminant," r Package Version 1.0, 2011. [Online]. Available: http://CRAN.R-project.org/package=penalizedLDA

[84] H. Akaike, "A new look at the statistical model identification," *IEEE Trans. Autom. Control*, vol. AC-19, no. 6, pp. 716–723, Dec. 1974.

[85] G. Schwartz, "Estimating the dimension of a model," *Ann. Statist.*, vol. 6, no. 2, pp. 461–464, 1978.

[86] J. Chen and Z. Chen, "Extended Bayesian information criteria for model selection with large model spaces," *Biometrika*, vol. 95, no. 3, pp. 759–771, 2008.

[87] M. Stone, "Cross-validatory choice and assessment of statistical predictions," *J. Royal Statist. Soc.*, vol. 39, pp. 44–47, 1974.

[88] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. New York, NY, USA: Springer-Verlag, 2001.

[89] T. R. Golub *et al.*, "Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, no. 5439, pp. 531–537, 1999.

[90] F. Dudbridge, "Power and predictive accuracy of polygenic risk scores," *PLOS Genetic*, vol. 9, no. 3, pp. 1–17, 2013.

## ABOUT THE AUTHORS

**Magnus Orn Ulfarsson** (Member, IEEE) received the B.S. and M.S. degrees from the University of Iceland, Reykjavik, Iceland, in 2002, and the Ph.D. degree from the University of Michigan, Ann Arbor, MI, USA, in 2007.

He joined the University of Iceland in 2007, where he is currently a Professor. He has been affiliated with deCODE Genetics, Reykjavik, Iceland, since 2013. His research interests include statistical signal processing, genomics, medical imaging, and remote sensing.

**Frosti Palsson** (Student Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from the University of Iceland, Reykjavik, Iceland, in 2012 and 2013, respectively, and is currently pursuing the Ph.D. degree at the University of Iceland.

His research interests include image fusion in remote sensing, and image and signal processing.

**Jakob Sigurdsson** (Member, IEEE) received the B.S. and M.S. degree and Ph.D. degree from the University of Iceland, Reykjavik, Iceland, in 2011 and 2015, respectively.

His research interests include statistical signal processing, remote sensing, and image processing.

**Johannes R. Sveinsson** (Senior Member, IEEE) received the B.S. degree from the University of Iceland, Reykjavík, Iceland, and the M.S. and Ph.D. degrees from Queen's University, Kingston, ON, Canada, all in electrical engineering.

He is currently a Professor with the Department of Electrical and Computer Engineering, University of Iceland. He was with the Laboratory of Information Technology and Signal Processing from 1981 to 1982 and the Engineering Research Institute and the Department of Electrical and Computer Engineering as a Senior Member of research staff and a Lecturer, respectively, from 1991 to 1998. He was a Visiting Research Student with the Imperial College of Science and Technology, London, U.K., from 1985 to 1986. At Queen's University, he held teaching and research assistantships. His current research interests are in systems and signal theory.

Dr. Sveinsson received the Queen's Graduate Awards from Queen's University. He is a co-recipient of the 2013 IEEE GRSS Highest Impact Paper Award.