Biometrics 000, 1–20 DOI: 000
000 000

A General Framework of Nonparametric Feature Selection in High-Dimensional Data

Hang Yu[1], Yuanjia Wang[2] and Donglin Zeng[3]

[1]Department of Statistics and Operation Research, University of North Carolina at Chapel Hill, Chapel Hill, North Carolina, U.S.A.

[2]Department of Biostatistics, Columbia University, New York, New York, U.S.A.

[3]Department of Biostatistics, University of North Carolina at Chapel Hill, Chapel Hill, North Carolina, U.S.A.

**email:* hangyu@live.unc.edu

***email:* yw2016@cumc.columbia.edu

****email:* dzeng@email.unc.edu

Summary: Nonparametric feature selection for high-dimensional data is an important and challenging problem in the fields of statistics and machine learning. Most of the existing methods for feature selection focus on parametric or additive models which may suffer from model misspecification. In this paper, we propose a new framework to perform nonparametric feature selection for both regression and classification problems. Under this framework, we learn prediction functions through empirical risk minimization over a reproducing kernel Hilbert space (RKHS). The space is generated by a novel tensor product kernel which depends on a set of parameters that determines the importance of the features. Computationally, we minimize the empirical risk with a penalty to estimate the prediction and kernel parameters simultaneously. The solution can be obtained by iteratively solving convex optimization problems. We study the theoretical property of the kernel feature space and prove the oracle selection property and Fisher consistency of our proposed method. Finally, we demonstrate the superior performance of our approach compared to existing methods via extensive simulation studies and applications to two real studies.

Key words: fisher consistency, oracle property, reproducing kernel Hilbert space, tensor product kernel, variable selection

This paper has been submitted for consideration for publication in *Biometrics*

1. Introduction

With fast technological advances in modern medicine, biomedical studies that collect complex data with a large number of features are becoming the norm. High-dimensional feature selection is an essential tool to allow using such data for disease prediction or precision medicine, for instance, to discover a subset of biomarkers that can predict treatment response to chronic disorders, or to determine predictive biomarkers for effective management of patient healthcare. Accurately identifying the subset of true important features and building an individualized treatment outcome prediction model is even more crucial and challenging in modern medicine and personalized healthcare. Specifically, in a motivating study presented in Section 4 (Trivedi et al., 2016), the goal is to identify behavioral and biological markers that can predict treatment response in patients affected by major depressive disorder (MDD) to guide clinical care. Treatment responses for mental disorders are inadequate and considerable heterogeneity is observed, in part because of a lack of knowledge on predictive markers that are informative of heterogeneous treatment effect. It is desirable to create a biosignature of treatment response for MDD by selecting and combining informative markers among a comprehensive candidate pool of clinical, behavioral, neuroimaging and electrophysiology measures.

High-dimensional feature selection has been extensively studied for linear model or generalized linear models in the past decades, and many methods have been developed including Lasso (Tibshirani, 1996), SCAD (Fan and Li, 2001), and MCP (Zhang, 2010). In these parametric models, the importance of individual features is characterized by non-null coefficients associated with them, so proper penalization can identify non-null coefficients with the probability tending to one when the sample size increases. However, parametric model assumptions are likely to be incorrect for many biomedical data due to potential higher-

order interactions among feature variables. In fact, applying these approaches to any simple transformation of feature variables may lead to very different feature selection results.

More recently, increasing efforts have been devoted to high-dimensional feature selection when parametric assumptions, especially linearity assumption, do not hold. Various approaches were proposed to select features based on measuring certain marginal dependency (Guyon and Elisseeff, 2003; Fan and Lv, 2008; Fan et al., 2011; Song et al., 2012; Yamada et al., 2014; Urbanowicz et al., 2018). For example, nonparametric association between each feature and outcome was used for screening (Fan and Lv (2008); Fan et al. (2011); Song et al. (2012). Li et al. (2012) adopted a robust rank correlation screening method based on marginal Kendall correlation coefficient. Yamada et al. (2014) considered a feature-wise kernelized Lasso, namely HSICLasso, for capturing nonlinear dependency between features and outcomes. In this approach, after a Lasso-type regression of an output kernel matrix on each feature-wise kernel matrix, non-important features with small marginal dependence in terms of a Hilbert-Schmidt independence criterion (HSIC) would be removed. However, all methods based on marginal dependence may fail to select true important variables since marginal dependency does not necessarily imply the significance of a feature when other features are also included for prediction, which is the case even for a simple linear model.

Alternatively, other approaches were proposed to relax parametric model assumptions and perform feature selection and prediction simultaneously. Lin and Zhang (2006) proposed COmponent Selection and Smoothing Operator (COSSO) to perform penalized variable selection based on smoothing spline ANOVA. Ravikumar et al. (2009) studied feature selection in a sparse additive model (SpAM), which assumed an additive model but allowed arbitrary nonparametric smoothers such as approximation in a reproducing kernel Hilbert space (RKHS) for each individual component function. Huang et al. (2010) considered spline approximation in the same model and adopted an adaptive group Lasso method to perform

feature selection. Wu and Stefanski (2015) also proposed a kernel based variable selection method as an extension of the additive model via local polynomial smoothing using a backfitting algorithm. Although COSSO, SpAM Wu and Stefanski (2015) allowed nonlinear prediction from each feature, they still imposed restrictive additive model structures, possible with some higher-order interactions. To allow arbitrary interactions among the features and perform a fully nonparametric prediction, Allen (2013) and Stefanski et al. (2014) proposed a procedure in which the feature input was constructed in a Gaussian RKHS in order to perform nonparametric prediction. Different weights were used for different features in the constructed Gaussian kernel function so that a larger weight implied a higher importance of the corresponding feature variable. However, due to high nonlinearity in the kernel function, estimating the weights was numerically unstable even when the dimension of the features was moderate. Finally, Yang et al. (2016) and Rosasco et al. (2013) considered model-free variable selection by examining the partial derivatives of regression functions with respect to each feature variable. Although theoretically an unimportant feature should yield a zero derivative, estimating the partial derivatives in a high-dimensional setting is known to be numerically unstable and such methods cannot be applied to non-continuous feature variables.

In this paper, we propose a general framework to perform nonparametric high-dimensional feature selection. We consider a general loss function which includes both regression models and classification as special cases. To perform nonparametric prediction, we construct a novel RKHS based on a tensor product of kernels for individual features. The constructed tensor product kernel, as discussed in Gao and Wu (2012), can handle any high-order nonlinear relationship between the features and outcome and any high-order interactions among the features. More importantly, each feature kernel depends on a non-negative parameter which determines the feature importance, so for feature selection, we further introduce a l_1-penalty of these parameters in the estimation. Computationally, coordinate descent algorithms are

used for updating parameters and each step involves simple convex optimization problems. Thus, our algorithm is numerically stable and can handle high-dimensional features easily. Theoretically, we first derive the approximation property of the proposed RKHS and characterize the complexity of the unit ball in this space in terms of bracket covering numbers. We then show that the estimated prediction function from our approach is consistent and moreover, we show that under some regularity conditions, the important features can be selected with the probability tending to one.

The rest of the paper is organized as follows. In Section 2, we introduce our proposed regularized tensor product kernel and lay out a penalized framework for both estimation and feature selection. We then provide detailed computational algorithms to solve the optimization problem. In Section 3, two simulation studies for regression and classification problems are conducted and we compare our method to existing methods. Applications to a microarray study and a depression study are given in Section 4. We conclude the paper with some discussion in Section 5.

2. Method

Suppose data are obtained from n independent subjects and consist of $(\mathbf{X}_i, Y_i), i = 1, ..., n$, where we let $\mathbf{X}$ denote p_n-dimensional feature variables and Y be the outcome which can be continuous, binary or ordinal. Our goal is to use the data to learn a nonparametric prediction function, $f(\mathbf{X})$, for the outcome Y.

We learn $f(\mathbf{X})$ through a regularized empirical risk minimization by assuming $f(\cdot)$ belongs to an RKHS associated with a kernel function, $\kappa(\mathbf{X}, \widetilde{\mathbf{X}})$, which will be described later. Specifically, if we denote the RKHS generated by $\kappa(\mathbf{X}, \widetilde{\mathbf{X}})$ by $\mathcal{H}_\kappa$, equipped with norm $\|\cdot\|_{\mathcal{H}_\kappa}$, then the empirical regularized risk minimization on RKHS for estimating $f(\mathbf{X})$ solves the

following optimization problem:

$$\min_f \mathbf{P}_n l(Y, f(\mathbf{X})) + \gamma_n \|f\|^2_{\mathcal{H}_\kappa},$$

where $l(y, f)$ is a pre-specified non-negative and convex loss function to quantify the prediction performance, $\mathbf{P}_n$ denotes the empirical measure from n observations, i.e., $\mathbf{P}_n g(Y, \mathbf{X}) = n^{-1}\sum_{i=1}^n g(Y_i, \mathbf{X}_i)$, and γ_n is a tuning parameter to control the complexity of f. For a continuous outcome, $l(y, f)$ is often chosen to be a L_2-loss given as $(y - f)^2$, while for a binary outcome, it can be one of the large-margin losses such as $\exp(-yf)$ in Adaboost since it yields the same classifier as the Bayesian classifier (Bartlett, 2006). There are many choices of kernel functions for $\kappa(\cdot, \cdot)$ so that the estimated $f(\mathbf{X})$ is nonlinear. One of the most commonly used kernel functions in machine learning is the Gaussian kernel function given by $\kappa(\mathbf{X}, \widetilde{\mathbf{X}}) = \exp\big(-\|\mathbf{X} - \widetilde{\mathbf{X}}\|^2/\sigma^2\big)$ for some bandwidth σ, where $\|\cdot\|$ is the Euclidean norm. To handle high-dimensional features, SpAM considered an additive kernel function by assuming $\kappa(\mathbf{X}, \widetilde{\mathbf{X}}) = \sum_{j=1}^{p_n} \exp(-|X_j - \widetilde{X}_j|^2/\sigma^2)$. In the KNIFE procedure, the kernel function is defined as $\kappa_{\boldsymbol{\omega}}(\mathbf{X}, \widetilde{\mathbf{X}}) = \exp\Big\{-\sum_{j=1}^{p_n} \omega_j (X_j - \widetilde{X}_j)^2/\sigma^2\Big\}$, where $\omega_j, j = 1, .., p_n$ are the additional weights to determine the feature importance.

To achieve the goal of both nonparametric prediction and feature selection, we propose a tensor product kernel as follows. For any given nonnegative vector $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \cdots \lambda_{p_n})^\intercal$, we define a $\boldsymbol{\lambda}$-regularized kernel function as

$$\kappa_{\boldsymbol{\lambda}, \sigma_n}(\mathbf{X}, \widetilde{\mathbf{X}}) = \prod_{m=1}^{p_n} \Big\{1 + \lambda_m \kappa_n(X_m, \widetilde{X}_m)\Big\}, \tag{1}$$

where $\kappa_n(x, y) = \exp\{-(x - y)^2/2\sigma_n^2\}$ with a pre-defined bandwidth σ_n in $\mathcal{R}$. There are two important observations for this new kernel function. First, it is the product of a univariate kernel function for each feature variable, which is given by $1 + \lambda_m \kappa_n(X_m, \widetilde{X}_m)$. Thus, the RKHS generated by $\kappa_{\boldsymbol{\lambda}, \sigma_n}$ is equivalent to the tensor product of the RKHS generated by each feature-specific space. Second, each univariate kernel function is essentially the same as the Gaussian kernel function when $\lambda_m \neq 0$. Consequently, the resulting tensor

product space is the same as the RKHS generated by the multivariate Gaussian kernel function from all features whose λ_m's are non-zero. Therefore, the closure for the RKHS generated by $\kappa_{\boldsymbol{\lambda},\sigma_n}$ consists of all functions that only depend on feature variables for which $\lambda_m \neq 0$. In other words, non-negative parameters, λ_m, completely capture and regularize the contribution of each feature X_m. In this way, feature selection can be achieved by estimating the regularization parameters, λ_m's, in the kernel function. In Web Figure 1, we present a tensor-product kernel function in two-dimensional space with different choices of λ_1 and λ_2. When increasing λ_1 (or λ_2) from zero to some positive number, the kernel function along X_1 (or X_2) direction becomes non-flat, indicating that such a kernel function can capture non-trivial functional form along this direction.

More specifically, using the proposed kernel function, we let $\mathcal{H}_{\boldsymbol{\lambda},\sigma_n}$ denote the RKHS corresponding to $\kappa_{\boldsymbol{\lambda},\sigma_n}$ so we aim to minimize

$$\begin{aligned} L_n(\boldsymbol{\lambda}, f) \equiv \quad & \mathbf{P}_n l\left\{Y, f(\mathbf{X})\right\} + \gamma_{1n}||f||^2_{\mathcal{H}_{\boldsymbol{\lambda},\sigma_n}} + \gamma_{2n} P(\boldsymbol{\lambda}) \\ \text{subject to} \quad & M \geqslant \lambda_1, \lambda_2, \cdots, \lambda_{p_n} \geqslant 0, \end{aligned} \tag{2}$$

where M is a pre-specified large constant. $P(\boldsymbol{\lambda}) = \sum_{m=1}^{p_n} P(\lambda_m) = \sum_{m=1}^{p_n} \lambda_m I(\lambda_m < M/2)$, which is a truncated Lasso, and γ_{1n}, γ_{2n} are tuning parameters. Here, we include an l_1 penalization term on the regularization vector to perform feature selection and restrict λ_m to be bounded. The latter bound is useful for numerical convergence to avoid the situation that some λ_m can diverge. Since our RKHS contains a constant and based on the representation theory (Aronszajn, 1950) for RKHS, solution for (2) takes form

$$f(\mathbf{X}) = \sum_{i=1}^{n} \alpha_i \kappa_{\boldsymbol{\lambda},\sigma_n}(\mathbf{X}, \mathbf{X}_i)$$

and

$$\|f\|^2_{\mathcal{H}_{\boldsymbol{\lambda},\sigma_n}} = \boldsymbol{\alpha}^T \mathbf{K}_{\boldsymbol{\lambda},\sigma_n} \boldsymbol{\alpha},$$

where $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_n)^T$ and $\mathbf{K}_{\boldsymbol{\lambda},\sigma_n}$ is an $n \times n$ matrix with entry $\kappa_{\boldsymbol{\lambda},\sigma_n}(\mathbf{X}_i, \mathbf{X}_j)$. Then the

optimization becomes solving

$$\min_{\alpha_1,\ldots,\alpha_n,\boldsymbol{\lambda}} \quad \mathbf{P}_n l\left\{Y, \sum_{i=1}^{n} \alpha_i \kappa_{\boldsymbol{\lambda},\sigma_n}(\mathbf{X}, \mathbf{X}_i)\right\} + \gamma_{1n}\boldsymbol{\alpha}^T \mathbf{K}_{\boldsymbol{\lambda},\sigma_n}\boldsymbol{\alpha} + \gamma_{2n}\sum_{m=1}^{p_n} \lambda_m I(\lambda_m < M/2)$$

$$\text{subject to} \quad M \geqslant \lambda_1, \lambda_2, \cdots, \lambda_{p_n} \geqslant 0.$$

We iterate between $\boldsymbol{\alpha}$ and $\boldsymbol{\lambda}$ to solve the above optimization problem. At the k-th iteration,

$$\boldsymbol{\alpha}^{k+1} = \min_{\boldsymbol{\alpha}} n^{-1}\sum_{j=1}^{n} l\left\{Y_j, \sum_{i=1}^{n} \alpha_i \kappa_{\boldsymbol{\lambda}^k,\sigma_n}(\mathbf{X}_j, \mathbf{X}_i)\right\} + \gamma_{1n}\boldsymbol{\alpha}^{\intercal}\mathbf{K}_{\boldsymbol{\lambda}^k,\sigma_n}\boldsymbol{\alpha} \tag{3}$$

$$\begin{aligned}\boldsymbol{\lambda}^{k+1} = \min_{0\leqslant\boldsymbol{\lambda}\leqslant M} & n^{-1}\sum_{j=1}^{n} l\left\{Y_j, \sum_{i=1}^{n} \alpha_i^{k+1} \kappa_{\boldsymbol{\lambda},\sigma_n}(\mathbf{X}_j, \mathbf{X}_i)\right\} \\ & + \gamma_{1n}(\boldsymbol{\alpha}^{k+1})^{\intercal}\mathbf{K}_{\boldsymbol{\lambda},\sigma_n}\boldsymbol{\alpha}^{k+1} + \gamma_{2n}\sum_{m=1}^{p_n} \lambda_m I(\lambda_m < M/2).\end{aligned} \tag{4}$$

Since the loss function is a convex loss, the optimization in (3) is a convex minimization problem, so many optimization algorithms can be applied. To solve (4) for $\boldsymbol{\lambda}$, we adopt a coordinate descent algorithm to update each λ_q $(q = 1, 2, \cdots, p_n)$ in turn. Specifically, to obtain λ_q^{k+1}, we fix $\lambda_1^{k+1}, \lambda_2^{k+1}, \cdots, \lambda_{q+1}^k, \lambda_{q+2}^k, \cdots, \lambda_{p_n}^k$ and then after simple calculation, the objective function takes the following form,

$$\min_{\lambda_q \geqslant 0} \frac{1}{n}\sum_{i=1}^{n} g(a_{iq} + b_{iq}\lambda_q) + d_q\lambda_q, \tag{5}$$

where $g(a_{jq} + b_{jq}\lambda_q)$ is equal to $l\left\{Y_j, \sum_{i=1}^{n} \alpha_i^{k+1} \kappa_{\boldsymbol{\lambda},\sigma_n}(\mathbf{X}_j, \mathbf{X}_i)\right\}$ as a function of λ_q, and a_{iq}, b_{iq}, d_q's are constants. By the construction of $\kappa_{\boldsymbol{\lambda},\sigma_n}$, $g(\cdot)$ is a differentiable and convex function so each step in the coordinating descent algorithm is a constrained convex minimization problem in a bounded interval, which is easy to solve.

We summarize the algorithm in the following table. At the convergence after k iterations, the final prediction function is given as

$$\widehat{f}_{\boldsymbol{\lambda}^{k+1}}(\mathbf{X}) = \sum_{i=1}^{n} \alpha_i^{k+1} \kappa_{\boldsymbol{\lambda}^{k+1},\sigma_n}(\mathbf{X}, \mathbf{X}_i).$$

For classification problem, the classification rule is

$$\text{sign}\left\{\widehat{f}_{\boldsymbol{\lambda}^{k+1}}(\mathbf{X})\right\} = \text{sign}\left\{\sum_{i=1}^{n} \alpha_i^{k+1} \kappa_{\boldsymbol{\lambda}^{k+1},\sigma_n}(\mathbf{X}, \mathbf{X}_i)\right\}.$$

We give details of our algorithm below (Algorithm 1).

Algorithm 1 Algorithm for learning $f(\mathbf{X})$

Input: Data $(\mathbf{X}, \mathbf{Y})$; Regularization parameter γ_{1n} and γ_{2n}; Former updating results, $\widehat{\boldsymbol{\alpha}}^k, \widehat{\boldsymbol{\lambda}}^k, \widehat{f}_{\widehat{\boldsymbol{\lambda}}^k}$;

Initialize For regression, $\widehat{\boldsymbol{\lambda}}_0 = \mathbf{0}$; For classification, $\widehat{\boldsymbol{\lambda}}_0 = (0, \cdots, 1, \cdots, 0)$, where all elements equal to 0, expect the one having largest margin correlation with outcome.

Iterate until convergence ($\delta = \text{abs}\left\{L_n(\widehat{\boldsymbol{\lambda}}^{k+1}, \widehat{f}_{\widehat{\boldsymbol{\lambda}}^{k+1}}) - L_n(\widehat{\boldsymbol{\lambda}}^{k}, \widehat{f}_{\widehat{\boldsymbol{\lambda}}^{k}})\right\} \leqslant c_1$, $e = \|\widehat{\boldsymbol{\lambda}}^{k+1} - \widehat{\boldsymbol{\lambda}}^{k}\|_1 \leqslant c_2$, where c_1 and c_2 are given cut points):

(i) Update $\widehat{\boldsymbol{\alpha}}^{k+1}$ for fix $\widehat{\boldsymbol{\lambda}}^{k}$, which can be solved explicitly for regression and via fminsearch function for classification.

(ii) Update $\widehat{\boldsymbol{\lambda}}^{k+1}$ for fixed $\widehat{\boldsymbol{\alpha}}^{k+1}$ via coordinate descent algorithm.

(iii) $\delta = \text{abs}\left\{L_n(\widehat{\boldsymbol{\lambda}}^{k+1}, \widehat{f}_{\widehat{\boldsymbol{\lambda}}^{k+1}}) - L_n(\widehat{\boldsymbol{\lambda}}^{k}, \widehat{f}_{\widehat{\boldsymbol{\lambda}}^{k}})\right\}$ and $e = \|\widehat{\boldsymbol{\lambda}}^{k+1} - \widehat{\boldsymbol{\lambda}}^{k}\|_1$.

Output: $\widehat{\boldsymbol{\alpha}}^{k+1}, \widehat{\boldsymbol{\lambda}}^{k+1}, \widehat{f}_{\widehat{\boldsymbol{\lambda}}^{k+1}}$.

Remark 1. When updating $\boldsymbol{\alpha}$ iteratively for regression, it can be solved in a closed form as $\widehat{\boldsymbol{\alpha}}^{k+1} = (\mathbf{K}^{\intercal}_{\widehat{\boldsymbol{\lambda}}^k,\sigma_n}\mathbf{K}_{\widehat{\boldsymbol{\lambda}}^k,\sigma_n} + n\gamma_{1n}\mathbf{K}_{\widehat{\boldsymbol{\lambda}}^k,\sigma_n})^{-1}\mathbf{K}^{\intercal}_{\widehat{\boldsymbol{\lambda}}^k,\sigma_n}\mathbf{Y}$. For classification, we apply one-step Newton method for updating. Tuning parameters in the algorithm are chosen via cross-validation over a grid of $2^{-15}, 2^{-13}, \cdots, 2^{13}, 2^{15}$. Although the kernel bandwidth, σ_n, can also be tuned, to save computation cost, we follow Jaakkola et al. (1999) to set it to be the median value of the paired distances. Following the same analysis as in Wright (2015), since g-function in (5) is convex and is twice-continuously differentiable, if we further assume that it is strictly convex and has bounded second derivatives in a neighborhood of the minimizer for $\boldsymbol{\lambda}$ in Step (ii), say $\mathcal{N}$, then the distance between the updated λ_q and its convergent value can be bounded by a mulitiplier of the distance for the previous λ_q. In addition, this multiplication factor is approximately $\text{abs}[1 - \{\max_{\boldsymbol{\lambda}\in\mathcal{N}} \sum_i g''(a_{iq} + b_{iq}\lambda_q)\}^{-1} \{\min_{\boldsymbol{\lambda}\in\mathcal{N}} \sum_i g''(a_{iq} + b_{iq}\lambda_q)\}]$ so is less than one when $\boldsymbol{\lambda}$ is sufficiently close to its convergent value. That is, the whole

coordinate descent algorithm has a linear convergence rate in a local neighborhood of the minimizer for $\boldsymbol{\lambda}$. Finally, since it is possible that the proposed algorithm will converge to a local minimum, we suggest to start with a few initial values, and consider the algorithm to converge once the objective function does not change more than a given threshold. In practice, we may also compare the prediction error from our algorithm with the ones from other methods such as SPAM or random forest, so it will give additional assurance when these errors are comparable.

Remark 2. In the supporting information accompanying this paper, we provide details for the properties of the proposed kernel function including its universal approximation property and complexity of the unit ball in its induced RKHS. Furthermore, we provide regularity conditions to show that our proposed prediction function leads to the best prediction performance asymptotically. We also establish the oracle property of variable selection using the proposed method in an ultra-high dimensional setting, i.e., when the dimension of the feature variables grows exponentially as a function of the sample size.

3. Simulation Study

We conducted two simulation studies, one for a regression problem with continuous Y and the other for classification with binary Y. In the first simulation study, we considered a continuous outcome model with a total number of p correlated feature variables, which were generated from a multivariate normal distribution, each with mean zero and variance one. Furthermore, X_1, X_2, X_3, X_4 were correlated with $corr(X_1, X_2) = 0.4$, $corr(X_1, X_3) = -0.3$, $corr(X_2, X_3) = 0.5$ and $corr(X_3, X_4) = 0.2$, while the others were all independent. The outcome variable, Y, was simulated from a linear model

$$Y = 0.9X_5^3 + 4X_1X_2X_3 + 2.3\exp(-X_3) + 4X_4 + \epsilon,$$

where $\epsilon \sim N(0,1)$. Thus, X_1 to X_5 were important variables but not any others. In the second simulation study, X's were generated similarly but with some different correlations: $corr(X_1, X_2) = -0.2$, $corr(X_1, X_4) = 0.2$, $corr(X_2, X_3) = 0.5$, $corr(X_2, X_4) = 0.3$ and $corr(X_3, X_4) = -0.4$. The binary outcome, Y, with values -1 and 1, was generated from a Bernoulli distribution with the probability of being one given by

$$\left\{1 + e^{-0.25+(X_2-1.1X_3+0.3X_4)^3}\right\}^{-1},$$

so only X_2 to X_4 were important variables. Since many biomedical applications (as well as our application in this work) have small to moderate sample sizes, in both simulation studies, we considered sample size $n = 100, 200$ and 400 and varied the feature dimension from $p = 200, 400$ to 1000. Each simulation setting was repeated 500 times.

For each simulated data, we used the proposed method to learn the prediction function. Initial values, tuning parameters and the optimization package used for binary case are chosen as in Remark 1 of Section 2, where 3-fold cross-validation was used for selecting the tuning parameters. The bound of regularized parameter M was chosen to be 10^5. We also centered the continuous outcome and re-weighted class label controlled to be balanced before iteration to make numerically stable. We reported the true positive rates, true negative rates and the average number of the selected variables for feature selection. We also reported the prediction errors or misclassification rates using a large and independent validation data. For comparison, we compared our proposed method with HSICLasso and SpAM since both methods were able to estimate nonlinear functions in high dimensional settings. In addition, we also compared the performance with LASSO in the first simulation study and l_1-SVM in the second simulation study, in order to study the impact due to model misspecification. In the simulations, our algorithm usually converged within 400 iterations for the continuous outcome and within 100 iterations for the binary outcome.

The results based on 500 replicates are summarized in Tables 1 and 2. From these tables,

we observe that for a fixed dimension, the performance of our method improves as sample size n becomes large in terms of the improved true positive and true negative rates for feature selection as well as decreasing prediction errors. In almost all cases, our true negative rate is close to 100%, which shows that noise variables can be identified with a very high chance. As expected, the performance deteriorates as the dimensionality increases. Interestingly, our method continues to select only a small number of feature variables. Comparatively, HSICLasso selected many more noise variables and had larger prediction errors, while SpAM also tended to select more features than our method. The performance of these methods become much worse when the feature dimension is 1000.

Clearly, LASSO and l_1-SVM did not yield reasonable variable selection results and their prediction errors are much higher due to model misspecification. We also give violin plots to visualize prediction performance of 500 replications in Figures 1 and 2. Since Lasso cannot provide stable prediction errors, its prediction errors from many replicates are out of the bound as shown in Figure 1. Figure 1 and 2 further confirm that our method is superior to all other methods, even when the dimension is as large as 1000 and the sample size is as small as $n = 100$, which is of similar size as our real data analysis example in Section 5. In the supporting information, we conducted an additional simulation study which had a similar setting to the first simulation study but allowed the dependence between the important and unimportant variables. Our method remains to perform better than the other methods.

[Table 1 about here.]

[Table 2 about here.]

[Figure 1 about here.]

[Figure 2 about here.]

4. Application

4.1 *Application to Microarray Study of Eye Disease in Animals*

We applied our proposed method to analyze a gene expression study in Scheetz et al. (2006). This study analyzed microarray RNAs of eye disease from 120 male rats, containing the expression levels from about $31,000$ gene probes. Gene TRIM32 is known to cause Bardet-Biedl syndrome (Chiang et al., 2006) but is responsible for less than half of cases. Therefore, identifying other genes associated with TRIM32 can help to identify additional novel disease genes, which provide a more complete gene signature to understand the disease mechanisms. The biological rationale behind this is that there is an evolutionary advantage in linking the expression of functionally related genes to the same biological system for which their function is needed. As a note, the association analysis between the causal gene and other genes using the same dataset has also been considered in Scheetz et al. (2006) and Huang et al. (2010) .

Since the expression value for TRIM 32 was skewed due to some extreme values, we dichotomized TRIM32 based on whether it was over expressed as compared to a reference sample in the dataset. We further restricted our feature variables to the top 1000 probe sets that were most correlated with TRIM32, while the analysis using all 31,000 probes was presented in the supporting information. All feature variables were on a log-scale and standardized in the analysis. To examine the performance of our method, we randomly divided the whole sample so that 70% was used for training and the rest was used for testing. This random splitting was then repeated 500 times to obtain reliable results. For each training data, we used 3-fold cross validation to choose tuning parameters. We also applied HSICLasso, SpAM and l_1-SVM for comparison.

The analysis results are shown in Table 3. We notice that our method gives almost the same classification error as l_1-SVM, which is the smallest on average. However, our method

selects a much smaller set of feature variables with an average of 5 variables. SpAM selects 13 variables on average, but its classification error is higher. In Table 3, we also report the top 10 most-frequently selected features among all 500 replications for each method. We notice that some features such as Fbxo7 and LOC102555217 were selected by at least three methods. In addition, Gene Sirt 3 was identified by all three nonlinear feature selection methods, but not l_1-SVM, indicating some possible nonlinear relationship between Sirt 3 and TRIM32. In the Online Supplementary, we provide a figure to reveal the nonlinear relationship between Sirt 3 and Fbxo7. We applied our method to analyze the whole sample and obtained a training error of 21.9% along five 5 genes identified (Fbxo7, Plekha6, Nfatc4, 1375872 and 1388656), which were all selected as the top 10 genes in the previous random splitting experiment. As a note, the average prediction errors when using only the top five feature variables were in turn 0.345, 0.344, 0.344, 0.367, and 0.294, indicating that each individual feature itself was not sufficient to achieve the best prediction.

[Table 3 about here.]

4.2 *Application to EMBARC Study*

In the second application, we applied our method to the Establishing Moderators and Biosignatures of Antidepressant Response for Clinical Care (EMBARC) study (Trivedi et al., 2016), which aimed to create a biosignature from clinical and biological markers to improve the response rates for MDD as well as to guide clinical care for patients. The study recruited early onset ($<$ 30 years) chronic or recurrent MDD patients and the study medication was Sertraline (an antidepressant). A comprehensive array of biomarkers were collected from participants: functional magnetic resonance imaging (fMRI) was used to assess key amygdala-ACC circuitry implicated in implicit emotion processing and regulation under task and to collect functional connectivity measures under resting state (a total of 232 variables including region of interest [ROI] biomarkers measuring regional brain activation under tasks and

functional connectivity during rest); electroencephalogram (EEG) was used to measure brain signals at several power spectral bands (e.g., alpha and theta power and loudness dependence of auditory evoked potential [LDAEP], a total of 11 variables); and structural MRI was used to collect diffusion tensor imaging white matter measures (2 variables). Other measures included demographics, quick inventory of depressive symptomatology (QIDS score) and behavioral phenotyping that measures psychomotor slowing, cognitive control (particularly post-error behavioral adjustments), working memory performance, and reward responsiveness. Details of the study measures were reported elsewhere (Trivedi et al., 2016).

Our study sample included 111 randomized patients. The outcome we considered was the treatment responder status assessed by a trained clinician (36 responders and 75 non-responders). Due to a large number of candidate feature variables (266 variables) compared to the sample size, it was essential to select features that can best predict patient response to create the biosignature. We standardized all the feature variables before fitting the model and treatment assignment was included as a feature variable. To further examine the performance of our method, we randomly divided the whole sample so that 70% was used for training and the rest was used for testing. We repeated 500 times of random splittings to obtain reliable results. We used 3-fold cross validation to choose tuning parameters. We compared with HSICLasso, SpAM and l_1-SVM as in the simulation studies.

The analysis results are shown in Table 4. Our proposed method has the smallest mean and median classification error among all the comparison methods. Regarding feature selection, our proposed method gives the sparsest results with only 3.36 variables selected, on average, among 500 replications. However, all the other three methods selected more than 40 variables with larger classification errors. The top most selected variables by our method include age, behavioral phenotypes (e.g., reaction time under various behavioral tasks), and neuroimaging biomarkers (e.g., mean activation in ROIs under various tasks). Moreover, one task fMRI

measure (i.e., mean activation of subgenual cingulate ROI under the conflict adaptation task) was chosen both by our method and SpAM, but not l_1-SVM, which indicates that there could be a nonlinear relationship between this moderator and the outcome. More interestingly, there are some overlaps regarding to the features selected by the proposed method and l_1-SVM, but our proposed method has a much smaller prediction error compared to l_1-SVM. This indicates that the proposed method successfully identified some predictors that have nonlinear relationships, which improves the classification error compared to a linear model. Note that the marginal classification errors for the top 5 features selected by our method range from 0.381 to 0.419, which are higher than the classification error when they are all included in our prediction method. These results provide empirical evidence to support using behavioral and neuroimaging measures to predict depression treatment responses, which has been suggested in theoretical models of depression biotypes (Williams, 2017).

[Table 4 about here.]

5. Discussion

In this work, we have proposed a general framework for nonparametric feature selection for both regression and classification in high dimensional settings. We introduced a novel tensor product kernel for empirical risk minimization. This kernel led to fully nonparametric estimation for the prediction function but allowed the importance of each feature to be captured by a non-negative parameter in the kernel function. Our approach is computationally efficient because it iteratively solves a convex optimization problem in a coordinate descent manner. We have shown that the proposed method has theoretical oracle property for variable selection. The superior performance of the proposed method was demonstrated via simulation studies and a real data application with a large number of feature variables.

We considered the l_2 loss function for regression and the exponential loss function for

classification as examples, respectively. Clearly, the proposed framework applies to feature selection under many different loss functions in the machine learning field. Another extension is to incorporate structures of feature variables in constructing the kernel function. For example, in integrative data analysis, feature variables arise from many different domains such as clinical domain, DNA, RNA, imaging and nutrition. It will be interesting to construct a hierarchical kernel function which can not only identify feature variables within each domain but also identify important domains at the same time. Furthermore, our theoretical results can be extended to not only derive the oracle selection of the important feature variables, but also obtain the convergence rate of the prediction error.

Our framework of nonparametric feature selection can be generalized to perform feature selection when estimating individualized treatment rules. We can adapt loss functions used for learning treatment rules in our proposed method to simultaneously accomplish nonparametric variable selection and discover optimal individualized treatment rules. Extensions to categorical outcomes and multi-stage treatment rule estimation can also be considered.

ACKNOWLEDGMENTS

We thank the co-editor, associate editor and reviewers for their constructive comments and NIMH Data Archive (study ID 2199). This work is partially supported by NIH grants NS073671, MH123487 and GM124104.

Data Availability Statement

The microarray data are publicly available and see Scheetz et al. (2006). The data for the Establishing Moderators and Biosignatures of Antidepressant Response for Clinical Care study is available to the public through https://nda.nih.gov/edit_collection.html?id=2199.

References

Allen, G. I. (2013). Automatic feature selection via weighted kernels and regularization. *Journal of Computational and Graphical Statistics* **22,** 284–299.

Aronszajn, N. (1950). Theory of reproducing kernels. *Transactions of the American Mathematical Society* **68,** 337–404.

Bartlett, P. (2006). Pattern classification and large margin classifiers. *Machine Learning Summer School* .

Chiang, A., Beck, J., Yen, H., Tayeh, M., Scheetz, T., Swiderski, R., and et al. (2006). Homozygosity mapping with snp arrays identifies trim32, an e3 ubiquitin ligase, as a bardet-biedl syndrome gene (bbs11). *Proceedings of the National Academy of Sciences of the United States of America* **103,** 6287–6292.

Fan, J., Feng, Y., and Song, R. (2011). Nonparametric independence screening in sparse ultra-high dimensional additive models. *Journal of the American Statistical Association* **106,** 544–557.

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association* **96,** 1348–1360.

Fan, J. and Lv, J. (2008). Sure independence screening for ultra-high dimensional feature space. *Journal of the Royal Statistical Society: Series B* **70,** 849–911.

Gao, C. and Wu, X. (2012). Kernel support tensor regression. *2012 International Workshop on Information and Electronics Engineering (IWIEE)* **29,** 3986–3990.

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

Huang, J., Horowitz, J. L., and Wei, F. (2010). Variable selection in nonparametric additive model. *The Annals of Statistics* **38,** 22822313.

Jaakkola, T., Diekhans, M., and Haussler, D. (1999). Using the fisher kernel method to

detect remote protein. *ISMB* **99,** 149–158.

Li, G., Peng, H., Zhang, J., and Zhu, L. (2012). Robust rank correlation based screening. *Annals of Statistics* **40,** 1846–1877.

Lin, Y. and Zhang, H. H. (2006). Component selection and smoothing in multivariate nonparametric regression. *The Annals of Statistics* **34,** 2272–2297.

Ravikumar, P., Lafferty, J., Liu, H., and Wasserman, L. (2009). Sparse additive models. *Journal of the Royal Statistical Society: Series B* **101,**.

Rosasco, L., Villa, S., Mosci, S., Santoro, M., and Verri, A. (2013). Nonparametric sparsity and regularization. *Journal of Machine Learning Research* **14,** 1665–1714.

Scheetz, T. E., Kim, K.-Y. A., Swiderski, R. E., and Philp, A. R. (2006). Regulation of gene expression in the mammalian eye and its relevance to eye disease. *Proc Natl Acad Sci U S A.* **103,** 14429–14434.

Song, L., Smola, A., Gretton, A., Bedo, J., and Borgwardt, K. (2012). Feature selection via dependence maximization. *Journal of Machine Learning Research* **13,** 1393–1434.

Stefanski, L. A., Wu, Y., and White, K. (2014). Variable selection in nonparametric classification via measurement error model selection likelihoods. *Journal of the American Statistical Association* **106,** 574–589.

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

Trivedi, M., McGrath, P., Fav, M., Parsey, R., Kurian, B., Phillips, M., and et al. (2016). Establishing moderators and biosignatures of antidepressant response in clinical care (embarc): Rationale and design. *J Psychiatr Res.* pages 11–23.

Urbanowicz, R. J., Meeker, M., Cava, W. L., and Olson, R. S. (2018). Relief-based feature selection: Introduction and review. *Journal of Biomedical Informatics* **85,** 189–203.

Williams, L. M. (2017). Defining biotypes for depression and anxiety based on large-scale

circuit dysfunction: A theoretical review of the evidence and future directions for clinical translation. *Depression and anxiety* **34,** 9–24.

Wright, S. (2015). Coordinate descent algorithms. *Mathematical Programming* **151,** 334.

Wu, Y. and Stefanski, L. A. (2015). Automatic structure recovery for additive models. *Biometrika* **102,** 381395.

Yamada, M., Jitkrittum, W., Sigal, L., Xing, E. P., and Sugiyama, M. (2014). High-dimensional feature selection by feature-wise non-linear lasso. *Neural Computation* **26,** 185–207.

Yang, L., Lv, S., and Wang, J. (2016). Model-free variable selection in reproducing kernel hilbert space. *Journal of Machine Learning Research* **17,** 1–24.

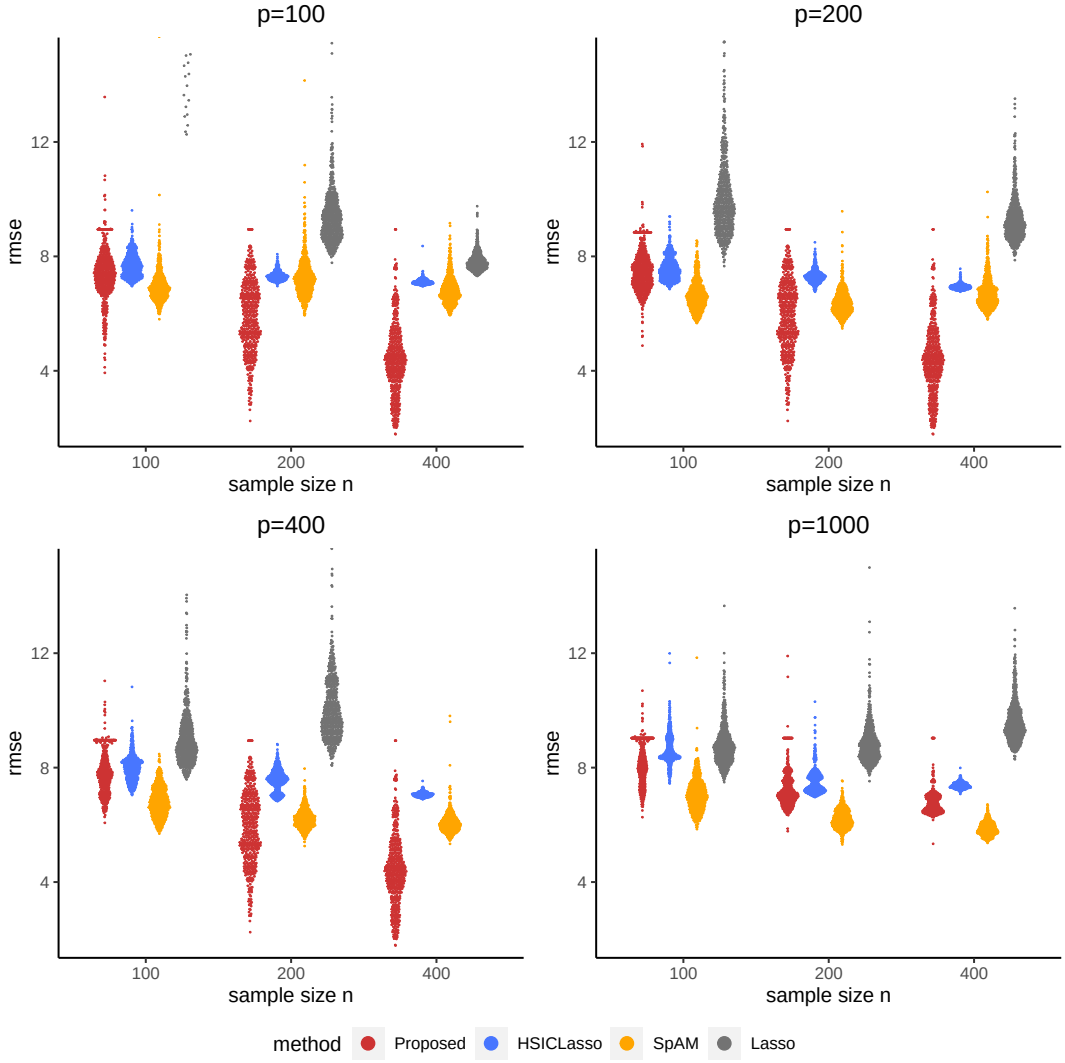
Zhang, C. (2010). Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics* **38,** 894–942.

Supporting Information

Web Appendices, Tables, and Figures referenced in Sections 2, 3 and 4.1 are available with this paper at the Biometrics website on Wiley Online Library. Web Appendix A the supporting information presents a plot of the kernel function with 2-dimensional feature variables. Web Appendix B in the supporting information includes conditions and theoretical justification for the oracle selection of the important features in the proposed method. Web Appendix C in the supporting information presents one additional plot for Section 4.1 and more results for analyzing the microarray example using all 31,000 probes. Web Appendix D in the supporting information presents one additional simulation study with same setting as the first simulation study in Section 3, but allowing the dependence between the important and unimportant variables. The codes and one simulated data are available in the supporting information and readers can also refer to the program for the link

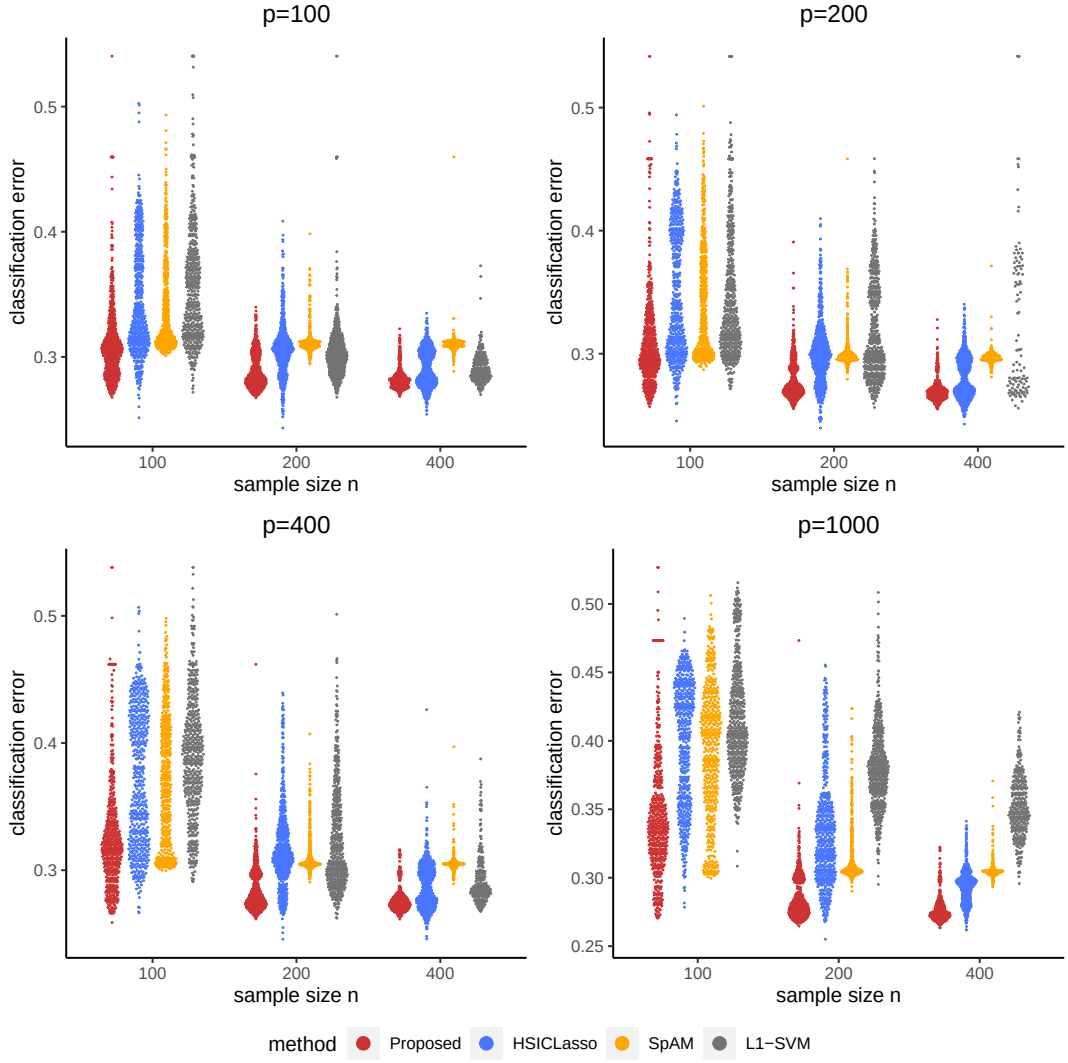
"https://github.com/Hyu4610/Nonparametric-Feature-Selection-in-High-Dimensional-Data".

Figure 1. Violin plots of Prediction Errors for Continuous Outcome



Note. The plots give the distribution of prediction errors among four competing methods. The comparing methods from left to right in each plot are our proposed method, HSICLasso, SpAM and Lasso. This figure appears in color in the electronic version of this article, and any mention of color refers to that version.

Figure 2. Violin plots of Misclassification Errors for Binary Outcome



Note. The plots give the distribution of misclassification rates among four competing methods. The comparing methods from left to right in each plot are our proposed method, HSICLasso, SpAM and l_1-SVM. This figure appears in color in the electronic version of this article, and any mention of color refers to that version.

Table 1
Results from The Simulation Study with Continuous Outcome

(a) Summary of Feature Selection Performance

		Proposed Method			HSICLasso			SPAM			LASSO		
p	n	TPR	TNR	Avg#	TPR	TNR	Avg#	TPR	TNR	Avg#	TPR	TNR	Avg#
100	100	60.9%	97.3%	5.6	81.5%	78.5%	24.5	99.6%	34.6%	67.1	98.8%	1.3%	98.8
	200	71.2%	99.0%	4.5	98.0%	60.4 %	42.5	100.0%	4.4%	95.8	100.0%	0.1%	99.9
	400	82.7%	98.4%	5.7	99.6 %	78.0 %	25.8	100.0%	0.3%	99.7	100.0%	0.1%	99.9
200	100	57.2%	98.7%	5.5	75.6%	88.8%	25.6	99.1%	63.0%	77.1	84.0%	52.2%	97.5
	200	66.6%	99.5%	4.2	94.0%	75.2%	53.1	100.0%	33.7%	134.1	99.1%	0.0%	198.6
	400	78.1%	99.4%	5.0	99.8 %	84.2%	35.8	100.0%	5.4%	189.5	100.0%	0.12%	199.8
400	100	47.3%	99.3%	5.2	68.5%	90.4%	41.5	98.2%	80.8%	80.8	79.4%	76.4%	97.1
	200	65.0%	99.7%	4.5	86.3%	89.0%	47.6	100.0%	62.1%	154.6	90.7%	51.4%	196.6
	400	73.1%	99.8%	4.4	99.7%	87.6%	54.0	100.0%	34.0%	265.8	99.1%	0.7%	397.3
1000	100	40.7%	99.7%	5.0	56.0%	91.8%	84.5	93.7%	92.2%	82.6	73.6%	90.6%	97.2
	200	61.2%	99.9%	4.5	78.2%	98.6%	18.4	99.9%	84.2%	162.0	85.5%	80.7%	196.1
	400	70.7%	99.9%	4.0	99.4%	91.0%	94.9	100.0%	68.7%	316.4	94.5%	60.7%	395.6

(b) Summary of Prediction Errors

p	n	Proposed Method	HSICLasso	SPAM	LASSO
100	100	7.405 (0.527)	7.695 (0.291)	6.985 (0.291)	41.663 (11.325)
	200	5.929 (0.950)	7.323 (0.098)	7.299 (0.389)	9.508 (0.575)
	400	4.424 (0.777)	7.115 (0.053)	6.868 (0.292)	7.840 (0.183)
200	100	7.567 (0.493)	7.603 (0.286)	6.672 (0.336)	10.176 (0.794)
	200	6.623 (0.412)	7.313 (0.130)	6.404 (0.279)	44.464 (9.125)
	400	5.661(0.580)	6.946 (0.054)	6.767 (0.305)	9.370 (0.433)
400	100	7.920 (0.670)	8.001 (0.284)	6.815 (0.399)	9.091 (0.532)
	200	7.008 (0.346)	7.563 (0.233)	6.222 (0.218)	10.151 (0.722)
	400	6.444 (0.199)	7.061 (0.049)	6.079 (0.192)	40.190 (6.402)
1000	100	8.215 (0.764)	8.638 (0.263)	7.067 (0.372)	8.851 (0.406)
	200	7.324 (0.368)	7.539 (0.252)	6.214 (0.242)	8.871 (0.379)
	400	6.818 (0.250)	7.376 (0.068)	5.870 (0.161)	9.652 (0.429)

Note. In (a), "TPR" is the true positive rate, "TNR" is the true negative rate, and "Avg#" is the average number of the selected variables from 500 replicates. In (b), the numbers are the mean squared errors from prediction, and the numbers within parentheses are the median absolute deviations from 500 replicates.

Table 2
Results from The Simulation Study with Binary Outcome

(a) Summary of Feature Selection Performance

		Proposed Method			HSICLasso			SPAM			l_1-SVM		
p	n	TPR	TNR	Avg#	TPR	TNR	Avg#	TPR	TNR	Avg#	TPR	TNR	Avg#
100	100	74.7%	99.0%	3.3	71.1%	79.4%	22.1	64.5%	89.6%	12.1	76.2%	75.1%	26.5
	200	83.9%	99.9%	2.6	80.7%	89.7 %	12.4	53.4%	98.9%	2.6	92.5%	80.4%	21.8
	400	86.0%	99.9%	2.6	87.8%	90.3%	12.1	50.6%	99.9%	1.5	98.8%	71.3%	30.8
200	100	70.4%	99.3%	3.5	71.3%	80.1%	41.3	63.6%	91.2%	19.2	71.3%	85.4%	31.0
	200	84.1%	99.8%	2.9	78.3%	95.0 %	12.2	54.5%	98.7%	4.1	90.7%	80.2%	41.8
	400	87.0%	100.0%	2.7	83.1 %	96.5%	9.3	50.6%	99.9%	1.6	89.3%	73.4%	55.0
400	100	68.5%	99.5%	3.9	70.9%	79.4%	84.0	63.7%	92.8%	30.6	65.9%	86.7%	54.7
	200	84.5%	99.9%	3.0	76.9%	95.5 %	20.2	57.7%	98.3%	8.3	87.0%	91.0%	38.1
	400	87.0%	100.0%	2.6	79.6 %	98.9%	6.8	51.9%	100.0%	1.8	99.1%	82.3%	73.0
1000	100	61.3%	99.8%	4.1	72.2%	77.4%	227.4	61.0%	95.7 %	45.0	58.4%	90.3 %	98.9
	200	86.3%	99.9%	3.3	75.5 %	95.9 %	43.6	54.3%	98.9 %	12.8	79.4%	91.4%	87.4
	400	87.7%	100.0%	2.8	73.9 %	99.6 %	6.6	50.0%	100.0%	1.9	96.8%	90.1 %	101.6

(b) Summary of Misclassification Errors

p	n	Proposed Method	HSICLasso	SPAM	l_1-SVM
100	100	0.314 (0.017)	0.345 (0.028)	0.343 (0.018)	0.359 (0.032)
	200	0.290 (0.009)	0.307 (0.012)	0.312 (0.002)	0.305 (0.011)
	400	0.283 (0.004)	0.292 (0.012)	0.297 (0.002)	0.292 (0.007)
200	100	0.316 (0.019)	0.351 (0.042)	0.344 (0.034)	0.352 (0.031)
	200	0.280 (0.008)	0.302 (0.015)	0.302 (0.003)	0.321 (0.028)
	400	0.270 (0.004)	0.282 (0.014)	0.297 (0.002)	0.326 (0.025)
400	100	0.331 (0.024)	0.372 (0.047)	0.369 (0.046)	0.390 (0.031)
	200	0.286(0.010)	0.319 (0.018)	0.311 (0.003)	0.327 (0.026)
	200	0.277 (0.004)	0.288 (0.014)	0.305 (0.001)	0.295 (0.010)
1000	100	0.352 (0.027)	0.397 (0.037)	0.390 (0.036)	0.416 (0.027)
	200	0.287 (0.008)	0.335 (0.024)	0.315 (0.003)	0.381 (0.020)
	400	0.277 (0.004)	0.294 (0.008)	0.305 (0.001)	0.353 (0.016)

Note. See Table 1.

Table 3
Top 10 Most Selected Genes for Each Method with Summary of Feature Selection Results in The Real Data Application Based on 500 Random Splittings

	Proposed Method	HSICLasso	SpAM	l_1-SVM
Top 10 Most Selected Genes	**Fbxo7** (67.5%)	Ska1 (76.6%)	1388491 (46.2%)	1376747 (99.1%)
	Plekha6 (47.3%)	**Sirt3** (76.2%)	**Fbxo7** (37.8%)	**1390538** (98.9%)
	LOC102555217 (24.5%)	Ddx58 (76.2%)	Slco1c1 (36.6%)	RragB (98.6%)
	Nfatc4 (22.7%)	1371610 (76.0%)	Stmn1 (35.4%)	Atl1 (97.9%)
	1390538 (20%)	LOC100912578 (73.2%)	1373944 (32.4%)	**Fbxo7** (97.3%)
	1375872 (20%)	Ttll7 (70.4%)	Ufl1 (32.2%)	**Plekha6** (95.1%)
	RGD1306148 (13.4%)	Decr1 (70.4%)	LOC100912578 (31.0%)	**1375872** (94.8%)
	Sirt3 (11.6%)	Mff (68.0%)	LOC100911357 (28.6%)	**RGD1306148** (94.1%)
	Prpsap2 (11.4%)	Pkn2 (67.0%)	**LOC102555217** (26.8%)	Ska1 (93.6%)
	1388656 (10.2%)	Taf11 (65.0%)	**Sirt3** (22.4%)	**LOC102555217** (93.2%)
min #	2	1	1	7
max #	13	1000	26	990
avg #	5.1	250.3	12.3	448.7
classification error	0.286 (0.057)	0.293 (0.046)	0.316 (0.057)	0.283 (0.058)

Note. For the top 10 most selected genes part, the numbers within parentheses are the frequencies to be selected in 500 random splittings. The genes also selected by the proposed method are highlighted in boldface. The last four rows give the summary of feature selection results. The numbers are the means of misclassification rates from 500 replicates. The numbers within parentheses are the median absolute deviations from 500 replicates. "min#" is the minimum number of the selected features, "max#" is the max number of the selected features, and "avg.#" is the average number of the selected features. This table appears in color in the electronic version of this article, and any mention of color refers to that version.

Table 4

Top 10 Most Selected Markers for Each Method with Summary of Feature Selection Results in The Real Data Application Based on 500 Random Splittings

	Proposed Method	HSICLasso	SpAM	l_1-SVM
Top 10 Most Selected Markers	**Task.median.reaction. time.mean** (40.4%)	Right.insula. perfusion.std (91.6%)	Dorsal.cingulate.conflict. adaptation.activation.mean (78.4%)	**Task.median.reaction. time.mean** (86.9%)
	Conflict.reaction. time.effect.mean (31.6%)	Left.Insula...Right.Insula. resting.coupling.mean (90.2%)	A/B choice test accuracy (77.2%)	**Conflict.reaction. time.effect.mean** (85.6%)
	Conflict.adaptation.reaction. time.effect.mean (26.0%)	RightAmygdala...Subgenual.cingulate. resting.coupling.mean (89.4%)	RightAmygdala...Subgenual.cingulate. resting.coupling.mean (68.8%)	Conflict.reaction. time.effect.mean (84.9%)
	Task.median.reaction. time.std (25.0%)	Left.insula.conflict. activation.std (88.6%)	Left.Insula...Right.Insula.resting.coupling.mean (76.0%)	**Task.median.reaction. time.std** (83.6%)
	Rabbit test reaction time (21.2%)	Ventral.striatum..seed..to.Perigenual.cingulate. coupling.anticipation.std (86.2%)	Left.ventral.striatum...Pregenual. cingulate.resting.coupling.mean (66.2%)	**Age** (81.9%)
	Age (17.6%)	Right..Insula...posterior.cingulate. resting.coupling.mean (85.4%)	Left.amygdala. perfusion.mean (65.2%)	**Rabbit test reaction time** (80.6%)
	Subgenual.cingulate.conflict. adaptation.activation.mean (11.2%)	Left.insula.conflict. activation.std (84.6%)	BP_w0_1924 (64.6%)	**Gratton test reaction time** (79.6%)
	Left.insula. conflict.activation.std (8.6%)	Ventral.striatum..seed..to. Perigenual.cingulate.coupling.anticipation.std (84.0%)	Rabbit test reaction time (63.8%)	Conflict.adaptation.reaction. time.effect.std (72.5%)
	Gratton test reaction time (7.2%)	Left.ventral.striatum. outcome.mean (83.2%)	Left.ventral.striatum. outcome.mean (62.0%)	Flanker test reaction time (71.8%)
	Right.ventral.striatum. outcome.std (5.6%)	Dorsal.anterior.cingulate. outcome.mean (82.6%)	**Subgenual.cingulate.conflict. adaptation.activation.mean** (58.0%)	Conflict.reaction. time.effect.std (69.9%)
min #	0	2	25	2
max #	32	267	140.6	266
avg #	3.36	140.6	41.2	41.6
mean of classification error	0.346 (0.045)	0.356 (0.058)	0.412 (0.065)	0.441 (0.123)
median of classification error	0.324	0.325	0.417	0.412

Note. See Table 3. This table appears in color in the electronic version of this article, and any mention of color refers to that version.