High-Dimensional Feature Selection by Feature-Wise Kernelized Lasso

Makoto Yamada
makotoy@yahoo-inc.com
Yahoo Labs, 701 1ˢᵗ Ave., Sunnyvale, CA 94098, U.S.A.

Wittawat Jitkrittum
wittawatj@gmail.com
University College London, Alexandra House, 17 Queen Square,
London, WC1N 3AR, U.K.

Leonid Sigal
lsigal@disneyresearch.com
Disney Research Pittsburgh, Pittsburgh, PA 15213, U.S.A.

Eric P. Xing
epxing@cs.cmu.edu
Carnegie Mellon University, Pittsburgh, PA 15213, U.S.A.

Masashi Sugiyama
sugi@cs.titech.ac.jp
Tokyo Institute of Technology O-okayama, Meguro-ku,
Tokyo, 152-8552, Japan

The goal of supervised feature selection is to find a subset of input features that are responsible for predicting output values. The least absolute shrinkage and selection operator (Lasso) allows computationally efficient feature selection based on linear dependency between input features and output values. In this letter, we consider a feature-wise kernelized Lasso for capturing nonlinear input-output dependency. We first show that with particular choices of kernel functions, nonredundant features with strong statistical dependence on output values can be found in terms of kernel-based independence measures such as the Hilbert-Schmidt independence criterion. We then show that the globally optimal solution can be efficiently computed; this makes the approach scalable to high-dimensional problems. The effectiveness of the proposed method is demonstrated through feature selection experiments for classification and regression with thousands of features.

1 Introduction

Finding a subset of features in high-dimensional supervised learning is an important problem with many real-world applications such as gene selection from microarray data (Xing, Jordan, & Karp, 2001; Ding & Peng, 2005; Suzuki, Sugiyama, Kanamori, & Sese, 2009; Huang, Horowitz, & Wei, 2010), document categorization (Forman, 2008), and prosthesis control (Shenoy, Miller, Crawford, & Rao, 2008).

1.1 Problem Description. Let $\mathcal{X}(\subset \mathbb{R}^d)$ be the domain of input vector x and $\mathcal{Y}(\subset \mathbb{R})$ be the domain of output data y.[1] Suppose we are given n independent and identically distributed (i.i.d.) paired samples,

$$\{(x_i, y_i)|x_i \in \mathcal{X}, y_i \in \mathcal{Y}, i = 1, \ldots, n\},$$

drawn from a joint distribution with density $p_{x,y}(x, y)$. We denote the original data by

$$X = [x_1, \ldots, x_n] \in \mathbb{R}^{d \times n},$$
$$y = [y_1, \ldots, y_n]^\top \in \mathbb{R}^n,$$

where $^\top$ denotes the transpose.

The goal of supervised feature selection is to find m features ($m < d$) of input vector x that are responsible for predicting output y.

1.2 Lasso. The least absolute shrinkage and selection operator (Lasso) (Tibshirani, 1996) allows computationally efficient feature selection based on the assumption of linear dependency between input features and output values.

The Lasso optimization problem is given as

$$\min_{\alpha \in \mathbb{R}^d} \frac{1}{2} \left\| y - X^\top \alpha \right\|_2^2 + \lambda \|\alpha\|_1,$$

where $\alpha = [\alpha_1, \ldots, \alpha_d]^\top$ is a regression coefficient vector, α_k denotes the regression coefficient of the kth feature, $\|\cdot\|_1$ and $\|\cdot\|_2$ are the ℓ_1- and ℓ_2-norms, and $\lambda > 0$ is the regularization parameter. The ℓ_1-regularizer in Lasso tends to produce a sparse solution, which means that the regression coefficients for irrelevant features become zero. Lasso is particularly useful when the number of features is larger than the number of training samples

---

[1] $\mathcal{Y}$ could be either continuous (i.e., regression) or categorical (i.e., classification). Structured outputs can also be handled in our proposed methods.

(Tibshirani, 1996). Furthermore, various optimization software packages were developed for efficiently computing the Lasso solution (Boyd & Vandenberghe, 2004; Daubechies, Defrise, & De Mol, 2004; Combettes & Wajs, 2005; Kim, Koh, Lustig, Boyd, & Gorinvesky, 2007; Yin, Osher, Goldfarb, & Darbon, 2008; Wright, Nowak, & Figueiredo, 2009; Tomioka, Suzuki, & Sugiyama, 2011).

However, a critical limitation of Lasso is that it cannot capture nonlinear dependency.

1.3 Instance-Wise Nonlinear Lasso. To handle nonlinearity, the instance-wise non-linear Lasso was introduced (Roth, 2004), where the original instance x is transformed by a nonlinear function $\psi(\cdot) : \mathbb{R}^d \to \mathbb{R}^{d'}$. Then the Lasso optimization problem is expressed as

$$\min_{\beta \in \mathbb{R}^n} \frac{1}{2} \|y - A\beta\|_2^2 + \lambda \|\beta\|_1,$$

where $A_{i,j} = \psi(x_i)^\top \psi(x_j) = A(x_i, x_j)$, $\beta = [\beta_1, \ldots, \beta_n]^\top$ is a regression coefficient vector and β_j is a coefficient of the jth basis $A(x, x_j)$.

The instance-wise nonlinear Lasso gives a sparse solution in terms of instances but not features. Therefore, it cannot be used for feature selection.

1.4 Feature-Wise Nonlinear Lasso (Feature Vector Machine). To obtain sparsity in terms of features, the feature-wise nonlinear Lasso was proposed (Li, Yang, & Xing, 2006).

The key idea is to apply a nonlinear transformation in a feature-wise manner, not in an instance-wise manner. More specifically, let us represent the sample matrix X in a feature-wise manner as

$$X = [u_1, \ldots, u_d]^\top \in \mathbb{R}^{d \times n},$$

where $u_k = [x_{k,1}, \ldots, x_{k,n}]^\top \in \mathbb{R}^n$ is the vector of the kth feature for all samples. Then the feature vector u_k and the output vector y are transformed by a nonlinear function $\phi(\cdot) : \mathbb{R}^n \to \mathbb{R}^p$. The Lasso optimization problem in the transformed space is given as

$$\min_{\alpha \in \mathbb{R}^d} \frac{1}{2} \|\phi(y) - \sum_{k=1}^d \alpha_k \phi(u_k)\|_2^2 + \lambda \|\alpha\|_1, \tag{1.1}$$

where $\alpha = [\alpha_1, \ldots, \alpha_d]^\top$ is a regression coefficient vector and α_k denotes the regression coefficient of the kth feature. By using the kernel trick (Schölkopf & Smola, 2002), equation 1.1 was shown to be equivalently expressed as the

following quadratic programming (QP) problem:

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^d} \ \frac{1}{2} \boldsymbol{\alpha}^\top \boldsymbol{D} \boldsymbol{\alpha},$$

$$\text{s.t.} \ \ \forall k, |\boldsymbol{\alpha}^\top \boldsymbol{d}_k - D(\boldsymbol{u}_k, \boldsymbol{y})| \leq \frac{\lambda}{2}, \tag{1.2}$$

where $D_{k,l} = \boldsymbol{\phi}(\boldsymbol{u}_k)^\top \boldsymbol{\phi}(\boldsymbol{u}_l) = D(\boldsymbol{u}_k, \boldsymbol{u}_l)$ and $\boldsymbol{D} = [\boldsymbol{d}_1, \dots, \boldsymbol{d}_d]$. This formulation is called the feature vector machine (FVM). Note that, since FVM uses the $d \times d$-dimensional Hessian matrix $\boldsymbol{D}$, it is especially useful when the number of training samples n is much bigger than that of features d.

In the original FVM, mutual information (Cover & Thomas, 2006) was used as the kernel function $D(\boldsymbol{u}, \boldsymbol{u}')$. However, the matrix $\boldsymbol{D}$ obtained from mutual information is not necessarily positive definite (Seeger, 2002), and thus the objective function, equation 1.2, can be nonconvex. Furthermore, when the number of training samples is smaller than that of features (which is often the case in high-dimensional feature selection scenarios), the matrix $\boldsymbol{D}$ is singular. This can cause numerical instability. Another restriction of FVM is that regardless of regression or classification, output $\boldsymbol{y}$ should be transformed by the same nonlinear function $\boldsymbol{\phi}(\cdot)$ as feature vector $\boldsymbol{u}$. This highly limits the flexibility of capturing nonlinear dependency. Finally, it is not statistically clear what kinds of features are found by this FVM formulation.

1.5 Contribution of This Paper. To overcome the limitations of FVM, we propose an alternative feature-wise nonlinear Lasso. More specifically, we propose to use particular forms of universal reproducing kernels (Steinwart, 2001) as feature and output transformations and solve the optimization problem in the primal space.

An advantage of this new formulation is that the global optimal solution can be computed efficiently. Thus, it is scalable to high-dimensional feature selection problems. To the best of our knowledge, this is the first convex feature selection method that is able to deal with high-dimensional nonlinearly related features. Furthermore, this new formulation has a clear statistical interpretation that nonredundant features with strong statistical dependence on output values are found using kernel-based independence measures such as the Hilbert-Schmidt independence criterion (HSIC) (Gretton, Bousquet, Smola, & Schölkopf, 2005) and the criterion based on the normalized cross-covariance operator (NOCCO) (Fukumizu, Gretton, Sun, & Schölkopf, 2008). Thus, the proposed methods can be regarded as a minimum redundancy maximum relevance–based feature selection method (Peng, Long, & Ding, 2005). In addition, the proposed methods are simple to implement, a highly preferable property for practitioners.

We also discuss the relation between the proposed method and existing feature selection approaches such as minimum redundancy maximum relevance (mRMR) (Peng et al., 2005), HSIC-based greedy feature selection

Table 1: Feature Selection Methods.

Method	Dependency	Optimization	Primal/Dual	Scalability with Regard to Number of Features
Lasso	Linear	**Convex**	**Primal**	**Highly scalable**
mRMR	**Nonlinear**	Greedy	—	Scalable
Greedy HSIC	**Nonlinear**	Greedy	—	Scalable
HSFS	**Nonlinear**	Nonconvex	—	Not scalable
FVM	**Nonlinear**	Nonconvex[a]	Dual	Not scalable
QPFS/KTA	**Nonlinear**	Nonconvex[a]	Dual	Not scalable
SpAM	Additive nonlinear	**Convex**	**Primal**	Scalable
Proposed	**Nonlinear**	**Convex**	**Primal**	**Highly scalable**

Notes: The methods that have preferable properties for high-dimensional feature selection problems are in bold.
[a]In practice, positive constants may be added to the diagonal elements of the Hessian matrix to guarantee the convexity, although the validity of selected features by this modification is not statistically clear.

(Song, Smola, Gretton, Bedo, & Borgwardt, 2012), quadratic programming feature selection (QPFS) (Rodriguez-Lujan, Huerta, Elkan, & Cruz, 2010), kernel target alignment (KTA) (Shawe-Taylor & Kandola, 2002; Cortes, Mohri, & Rostamizadeh, 2012), Hilbert-Schmidt feature selection (HSFS) (Masaeli, Fung, & Dy, 2010), and sparse additive models (SpAM) (Ravikumar, Lafferty, Liu, & Wasserman, 2009; Liu, Lafferty, & Wasserman, 2009; Raskutti, Wainwright, & Yu, 2012). See Table 1 for the summary of feature selection methods.

Through experiments on real-world feature selection problems, we show that the proposed methods compare favorably with existing feature selection methods.

2 Proposed Methods

In this section, we propose alternative implementations of the nonlinear feature-wise Lasso.

2.1 HSIC Lasso. We propose a feature-wise nonlinear Lasso of the following form, which we call the HSIC Lasso:[2]

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^d} \ \frac{1}{2} \Big\| \bar{L} - \sum_{k=1}^{d} \alpha_k \bar{K}^{(k)} \Big\|_{\text{Frob}}^2 + \lambda \|\boldsymbol{\alpha}\|_1,$$

$$\text{s.t.} \ \ \alpha_1, \ldots, \alpha_d \geq 0, \tag{2.1}$$

---

[2]A Matlab implementation of the proposed algorithm is available online at http://www.makotoyamada-ml.com/hsiclasso.html .

where $\| \cdot \|_{\mathrm{Frob}}$ is the Frobenius norm, $\bar{K}^{(k)} = \Gamma K^{(k)} \Gamma$ and $\bar{L} = \Gamma L \Gamma$ are centered Gram matrices, $K_{i,j}^{(k)} = K(x_{k,i}, x_{k,j})$ and $L_{i,j} = L(y_i, y_j)$ are Gram matrices, $K(x, x')$ and $L(y, y')$ are kernel functions, $\Gamma = I_n - \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^\top$ is the centering matrix, I_n is the n-dimensional identity matrix, and $\mathbf{1}_n$ is the n-dimensional vector with all ones. Note that we employ nonnegativity constraint for α so that meaningful features are selected (see section 2.2 for details). In addition, since we use the output Gram matrix L to select features in HSIC Lasso, we can naturally incorporate structured outputs using kernels. Moreover, we can perform feature selection even if the training data set consists of input x and its affinity information L such as link structures between inputs.

Differences from the original formulation, equation 1.1 are that we allow the kernel functions K and L to be different and the nonnegativity constraint is imposed. The first term in equation 2.1 means that we are regressing the output kernel matrix $\bar{L}$ by a linear combination of feature-wise input kernel matrices $\{\bar{K}^{(k)}\}_{k=1}^d$.

2.2 Interpretation of HSIC Lasso. Here we show that HSIC Lasso can be regarded as a minimum redundancy maximum relevancy (mRMR)–based feature selection method (Peng et al., 2005), a popular feature selection strategy in machine learning and artificial intelligence communities.

The first term in equation 2.1 can be rewritten as

$$\frac{1}{2} \left\| \bar{L} - \sum_{k=1}^d \alpha_k \bar{K}^{(k)} \right\|_{\mathrm{Frob}}^2 = \frac{1}{2} \mathrm{HSIC}(y, y) - \sum_{k=1}^d \alpha_k \mathrm{HSIC}(u_k, y)$$
$$+ \frac{1}{2} \sum_{k,l=1}^d \alpha_k \alpha_l \mathrm{HSIC}(u_k, u_l), \tag{2.2}$$

where $\mathrm{HSIC}(u_k, y) = \mathrm{tr}(\bar{K}^{(k)} \bar{L})$ is a kernel-based independence measure called the (empirical) Hilbert-Schmidt independence criterion (HSIC) (Gretton et al., 2005) and $\mathrm{tr}(\cdot)$ denotes the trace. $\mathrm{HSIC}(y, y)$ is a constant and can be ignored. HSIC always takes a nonnegative value and is zero if and only if two random variables are statistically independent when a universal reproducing kernel (Steinwart, 2001) such as the gaussian kernel is used. Note that the empirical HSIC asymptotically converges to the true HSIC with $O(1/\sqrt{n})$ (see theorem 3 in Gretton et al., 2005). In addition, HSIC can be regarded as the centered version of the kernel target alignment (KTA) (Shawe-Taylor & Kandola, 2002).

If the kth feature u_k has high dependence on output y, $\mathrm{HSIC}(u_k, y)$ takes a large value and thus α_k should also take a large value so that equation 2.1 is minimized. On the other hand, if u_k is independent of y, $\mathrm{HSIC}(u_k, y)$ is close to zero, and thus such α_k tends to be eliminated by the ℓ_1-regularizer.

This means that relevant features that have strong dependence on output y tend to be selected by HSIC Lasso.

Furthermore, if u_k and u_l are strongly dependent (i.e., redundant features), $\text{HSIC}(u_k, u_l)$ takes a large value, and thus either of α_k and α_l tends to be zero. This means that redundant features tend to be eliminated by HSIC Lasso.

Overall, HSIC Lasso tends to find nonredundant features with strong dependence on output y, which is the idea of feature selection methods based on minimum redundancy maximum relevancy (mRMR) (Peng et al., 2005). This is a preferable property in feature selection.

Note that it is possible to remove the nonnegativity constraint in equation 2.1 and select features that have nonzero coefficients α. However, if we allow negative values in α, it is hard to interpret selected features. Indeed, interpretability is one of important properties in feature selection, and thus we include the nonnegativity constraint for HSIC Lasso.

2.3 Kernel Selection. In theory, a universal kernel such as the gaussian kernel or the Laplace kernel permits HSIC to detect dependence between two random variables (Gretton et al., 2005). Moreover, it has been proposed to use the delta kernel for multiclass classification problems (Song et al., 2012). Thus, in this letter, we employ the gaussian kernel for inputs. For output kernels, we use the gaussian kernel for regression cases and the delta kernel for classification problems.

For input x, we first normalize the input x to have unit standard deviation, and we use the gaussian kernel,

$$K(x, x') = \exp\left(-\frac{(x - x')^2}{2\sigma_x^2}\right),$$

where σ_x is the gaussian kernel width.

In regression scenarios (i.e., $y \in \mathbb{R}$), we normalize an output y to have unit standard deviation, and we use the gaussian kernel,

$$L(y, y') = \exp\left(-\frac{(y - y')^2}{2\sigma_y^2}\right),$$

where σ_y is the gaussian kernel width.

In classification scenarios (i.e., y is categorical), we use the delta kernel for y,

$$L(y, y') = \begin{cases} 1/n_y & \text{if } y = y', \\ 0 & \text{otherwise,} \end{cases}$$

where n_y is the number of samples in class y. Note that it is also possible to use the gaussian kernel in classification scenarios, but it tends to perform poorly (see Figure 4).

2.4 Computational Properties of HSIC Lasso. An important computational property of HSIC Lasso is that the first term in equation 2.1 can be rewritten as

$$\frac{1}{2}\left\| \text{vec}(\bar{L}) - [\text{vec}(\bar{K}^{(1)}), \dots, \text{vec}(\bar{K}^{(d)})]\boldsymbol{\alpha} \right\|_2^2,$$

where $\text{vec}(\cdot)$ is the vectorization operator. This is the same form as plain Lasso with n^2 samples and d features.

If $d > n^2$ (i.e., high-dimensional feature selection from a small number of training samples), the Lasso optimization technique called the dual augmented Lagrangian (DAL) was shown to be computationally highly efficient (Tomioka et al., 2011).[3] Because DAL can also incorporate the non-negativity constraint without losing its computational advantages, we can directly use DAL to solve our HSIC Lasso problem. In contrast, when $n^2 \geq d$, we may use the either cKTM or FVM (i.e., dual) formulation.

If the number of samples n is relatively large, the gaussian kernel computation in HSIC Lasso is expensive. Therefore, the overall computation cost of HSIC Lasso is high. In addition, since naive implementation of HSIC Lasso requires n^2d memory space, it is not practical if both d and n are large (e.g., $d > 10000$ and $n > 1000$). Here, we propose a table lookup approach to reduce the computation time and memory size.

More specifically, based on the fact that the gaussian kernel values depend on only the difference of two input values, we normalize every feature x to have mean zero and unit standard deviation and discretize the difference of two input values into B values (we use $B = 4096$ in our implementation). Then we prepare in advance a lookup table of B elements that contain gaussian kernel values and refer to these values when we compute the gaussian kernels. The centered Gram matrix $\bar{K}$ can be rewritten as

$$\bar{K} = \left(I - \frac{1}{n}\mathbf{1}\mathbf{1}^\top\right)K\left(I - \frac{1}{n}\mathbf{1}\mathbf{1}^\top\right)$$
$$= K - \mathbf{1}m^\top - m\mathbf{1}^\top + s\mathbf{1}\mathbf{1}^\top,$$

where $m = \frac{1}{n}K\mathbf{1} \in \mathbb{R}^n$ and $s = \frac{1}{n^2}\mathbf{1}^\top K\mathbf{1} \in \mathbb{R}$. Thus, to compute $\bar{K}$, we only need to store m and s. Namely, the required memory size for the table lookup–based approach is $O(dn + B)$, which is much smaller than the memory size required for the naive implementation, $O(dn^2)$.

---

[3]http://www.ibis.t.u-tokyo.ac.jp/ryotat/dal/.

Another approach to deal with large sample size is using stability selection, which consists of running HSIC Lasso many times with subsampling and computing the number of times each feature is selected across the runs (Meinshausen & Bühlmann, 2010; Bach, 2008).

2.5 Variation: NOCCO Lasso. Instead of $\bar{K}^{(k)}$ and $\bar{L}$, we use $\widetilde{K}^{(k)} = \bar{K}^{(k)}(\bar{K}^{(k)} + \epsilon n I_n)^{-1}$ and $\widetilde{L} = \bar{L}(\bar{L} + \epsilon n I_n)^{-1}$, where $\epsilon > 0$ is a regularization parameter. Then our optimization problem is expressed as

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^d} \frac{1}{2} \sum_{k,l=1}^{d} \alpha_k \alpha_l D^{\text{NOCCO}}(\boldsymbol{u}_k, \boldsymbol{u}_l) - \sum_{k=1}^{d} \alpha_k D^{\text{NOCCO}}(\boldsymbol{u}_k, \boldsymbol{y}) + \lambda \|\boldsymbol{\alpha}\|_1,$$

$$\text{s.t. } \alpha_1, \ldots, \alpha_d \geq 0,$$

where $D^{\text{NOCCO}}(\boldsymbol{u}_k, \boldsymbol{y}) = \text{tr}(\widetilde{K}^{(k)}\widetilde{L})$ is the kernel-based dependence measure based on the (empirical) normalized cross-covariance operator (NOCCO) (Fukumizu et al., 2008). We call this formulation the NOCCO Lasso.

Because D_{NOCCO} was shown to be asymptotically independent of the choice of kernels, NOCCO Lasso is expected to be less sensitive to the kernel parameter choice than HSIC Lasso, although ϵ needs to be tuned in practice.

2.6 Other Types of Regularizers. The proposed method is amenable to most of the popular regularizers such as group Lasso and elastic net regularizers (Meier, Van De Geer, & Bühlmann, 2008; Zou & Hastie, 2005), as well as to other feature selection problems. For example, the group Lasso regularizer can be easily incorporated into our framework as

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^d} \frac{1}{2} \|\bar{L} - \sum_{k=1}^{d} \alpha_k \bar{K}^{(k)}\|_{\text{Frob}}^2 + \lambda \sum_{g=1}^{G} \|\boldsymbol{\alpha}_g\|_2,$$

$$\text{s.t. } \alpha_1, \ldots, \alpha_d \geq 0,$$

where $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1^\top, \ldots, \boldsymbol{\alpha}_G^\top]^\top$, $\boldsymbol{\alpha}_g$ is the gth group of variables and G is the number of groups. This group Lasso problem can also be efficiently solved by the DAL package with the nonnegativity constraint.

2.7 Relation to Two-Stage Multiple Kernel Learning. The proposed method is closely related to the two-stage multiple kernel learning (MKL) method called centered kernel target alignment (cKTA) (Cortes et al., 2012), which was originally proposed for learning a kernel Gram matrix (not used for feature selection problems).

If we adopt cKTA for supervised feature selection problems, the optimization problem of cKTA can be written as

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^d} \frac{1}{2} \sum_{k,l=1}^{d} \alpha_k \alpha_l \mathrm{HSIC}(\boldsymbol{u}_k, \boldsymbol{u}_l) - \sum_{k=1}^{d} \alpha_k \mathrm{HSIC}(\boldsymbol{u}_k, \boldsymbol{y})$$

$$\text{s.t. } \alpha_1, \ldots, \alpha_d \geq 0.$$

Differences from the HSIC Lasso are that cKTA solves the optimization problem in the dual space and does not have the ℓ_1 regularization term.

An advantage of cKTA is that feature selection can be performed by solving a nonnegative least-squares problem. Moreover, since cKTA has $d \times d$ dimensional Hessian matrix ($D_{kl} = \mathrm{HSIC}(\boldsymbol{u}_k, \boldsymbol{u}_l)$), cKTA is computationally efficient for feature selection problems with large n and small d. However, the Hessian matrix $\boldsymbol{D}$ is not necessarily positive definite (Seeger, 2002) and is singular in high-dimensional problems. More specifically, $\boldsymbol{D}$ can be written as

$$\boldsymbol{D} = [\mathrm{vec}(\bar{\boldsymbol{K}}^{(1)}), \ldots, \mathrm{vec}(\bar{\boldsymbol{K}}^{(d)})]^\top [\mathrm{vec}(\bar{\boldsymbol{K}}^{(1)}), \ldots, \mathrm{vec}(\bar{\boldsymbol{K}}^{(d)})],$$

and $\boldsymbol{D}$ is singular when $d > n^2$ (i.e., high-dimensional feature selection from a small number of training samples). Thus, solving the nonnegative least-squares problem in high-dimensional feature selection problems can be cumbersome in practice.

For the above reason, the proposed feature selection method is more suited than cKTA for high-dimensional feature selection problems. In contrast, if we want to solve a small d and large n feature selection problem, cKTA is more suited than the HSIC Lasso.

3 Existing Methods

In this section, we review existing feature selection methods and discuss their relation to the proposed approach. (See Table 1 for the summary of feature selection methods.)

3.1 Minimum Redundancy Maximum Relevance. Minimum redundancy maximum relevance (mRMR) (Peng et al., 2005) is a mutual information–based feature selection criterion.

Let $\boldsymbol{V} = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m]^\top \in \mathbb{R}^{m \times n}$ be a submatrix of $\boldsymbol{X} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_d]^\top \in \mathbb{R}^{d \times n}$, where m features are extracted from d features. Then mRMR for $\boldsymbol{V}$ is defined as

$$\mathrm{mRMR}(\boldsymbol{V}) = \frac{1}{m} \sum_{k=1}^{m} \widehat{\mathrm{MI}}(\boldsymbol{v}_k, \boldsymbol{y}) - \frac{1}{m^2} \sum_{k,l=1}^{m} \widehat{\mathrm{MI}}(\boldsymbol{v}_k, \boldsymbol{v}_l), \tag{3.1}$$

where $\widehat{\mathrm{MI}}(\boldsymbol{v}, \boldsymbol{y})$ is an empirical approximator of mutual information given as

$$\widehat{\mathrm{MI}}(\boldsymbol{v}, \boldsymbol{y}) = \iint \widehat{p}_{\mathrm{v,y}}(\boldsymbol{v}, \boldsymbol{y}) \log \frac{\widehat{p}_{\mathrm{v,y}}(\boldsymbol{v}, \boldsymbol{y})}{\widehat{p}_{\mathrm{v}}(\boldsymbol{v})\widehat{p}_{\mathrm{y}}(\boldsymbol{y})} \mathrm{d}\boldsymbol{v}\mathrm{d}\boldsymbol{y}.$$

$\widehat{p}_{\mathrm{v,y}}(\boldsymbol{v}, \boldsymbol{y})$ denotes a Parzen window estimator of the joint density of $\boldsymbol{v}$ and $\boldsymbol{y}$, and $\widehat{p}_{\mathrm{v}}(\boldsymbol{v})$ and $\widehat{p}_{\mathrm{y}}(\boldsymbol{y})$ denotes Parzen window estimators of marginal densities of $\boldsymbol{v}$ and $\boldsymbol{y}$, respectively.

The first term in mRMR measures the dependency between chosen feature $\boldsymbol{v}_k$ and output $\boldsymbol{y}$, while the second term is a penalty for selecting redundant features. Thus, mRMR finds nonredundant features with strong dependence on outputs. A very fast implementation of mRMR that is available and can deal with high-dimensional feature selection problems.

mRMR-based feature selection is performed by finding a submatrix $\boldsymbol{V}$ that maximizes equation 3.1. However, since there are 2^d possible feature subsets, the brute force approach is computationally intractable. Hence, greedy search strategies such as forward selection or backward elimination are used in practice (Peng et al., 2005). However, the greedy approaches tend to produce a locally optimal feature set.

Another potential weakness of mRMR is that mutual information is approximated by Parzen window estimation. Parzen window–based mutual information estimation is unreliable when the number of training samples is small (Suzuki et al., 2009).

3.2 Greedy Feature Selection with HSIC. The HSIC-based feature selection criterion (Song et al., 2012) is defined as

$$\mathrm{tr}(\bar{\boldsymbol{M}}\bar{\boldsymbol{L}}), \tag{3.2}$$

where $\bar{\boldsymbol{M}} = \boldsymbol{\Gamma}\boldsymbol{M}\boldsymbol{\Gamma}$ is a centered Gram matrix, $M_{i,j} = M(\boldsymbol{v}_i, \boldsymbol{v}_j)$ is a Gram matrix, $M(\boldsymbol{v}, \boldsymbol{v}')$ is a kernel function, and $(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m) = \boldsymbol{V} \in \mathbb{R}^{m \times n}$.

HSIC-based greedy feature selection is performed by finding a submatrix $\boldsymbol{V}$ that maximizes equation 3.2. An advantage of HSIC-based feature selection is its simplicity; it can be implemented very easily. However, since the maximization problem equation 3.2 is NP-hard, forward selection or backward elimination strategies are used for finding a locally optimal solution in practice (Song et al., 2012).

3.3 Hilbert-Schmidt Feature Selection. Hilbert-Schmidt feature selection (HSFS) (Masaeli et al., 2010) is defined as

$$\min_{\boldsymbol{W} \in \mathbb{R}^{d \times d}} -\mathrm{HSIC}(\boldsymbol{W}\boldsymbol{X}, \boldsymbol{y}) + \lambda \sum_{j=1}^{d} \|\boldsymbol{w}_j\|_{\infty},$$

where $W = [w_1, \ldots, w_d]$ is a transformation matrix, $\lambda > 0$ is the regularization parameter, and $\| \cdot \|_\infty$ is the ℓ_∞-norm.

HSFS can be regarded as a continuously relaxed version of the HSIC-based feature selection (Song et al., 2012). Thanks to this continuous formulation, the HSFS optimization problem can be solved by limited-memory BFGS (L-BFGS) (Nocedal & Wright, 2003). However, since HSFS is a nonconvex method, restarting from many different initial points would be necessary to select good features, which is computationally expensive. Moreover, HSFS attempts to optimize a projection matrix that has d^2 parameters. That is, following the original HSFS implementation based on a quasi-Newton method, the total computational complexity of HSFS is $O(d^4)$, which can be unacceptably large in high-dimensional feature selection problems. To reduce the computational cost, it may be able to approximately solve the HSFS optimization problem by reducing the size of the transformation matrix to $d \times q$ for $q \ll d$. However, this approximation leads to an additional tuning parameter that cannot be chosen objectively.

3.4 Quadratic Programming Feature Selection. Quadratic programming feature selection (QPFS) (Rodriguez-Lujan et al., 2010) tries to find features by solving a QP problem.

The QPFS optimization problem is defined as

$$\min_{\theta \in \mathbb{R}^d} \frac{(1-\gamma)}{2} \theta^\top D \theta - \gamma \theta^\top d,$$

$$\text{s.t.} \sum_{k=1}^{d} \theta_k = 1, \theta_1, \ldots, \theta_d \geq 0,$$

where $\theta = [\theta_1, \ldots, \theta_d]^\top$, $D_{k,l} = D(u_k, u_l)$, $d = [D(u_1, y), \ldots, D(u_d, y)]^\top$, $D(\cdot)$ is a dependency measure, and $\gamma \in [0, 1]$ is a tuning parameter. In QPFS, an empirical estimator of mutual information is used as a dependency measure. Note that if we employ HSIC as a dependency measure in QPFS and remove the sum-to-one constraint, QPFS is equivalent to the centered KTA (cKTA) (Cortes et al., 2012), a multiple kernel learning method originally proposed for learning a kernel matrix (see section 2.7 for details).

Similar to cKTA, an advantage of QPFS is that feature selection can be performed just by solving a QP problem. Moreover, since QPFS has $d \times d$-dimensional Hessian matrix, QPFS is computationally efficient for feature selection problems with large n and small d. However, the Hessian matrix D is not necessarily positive definite (Seeger, 2002) and is singular in high-dimensional problems.

3.5 Sparse Additive Models. The sparse additive model (SpAM) is useful for high-dimensional feature selection (Ravikumar et al., 2009; Liu et al., 2009; Raskutti et al., 2012; Suzuki & Sugiyama, 2013).

The SpAM optimization problem can be expressed as

$$\min_{\boldsymbol{\beta}_1,\ldots,\boldsymbol{\beta}_d \in \mathbb{R}^n} \left\| \boldsymbol{y} - \sum_{k=1}^{d} \boldsymbol{K}^{(k)} \boldsymbol{\beta}_k \right\|_2^2 + \lambda \sum_{k=1}^{d} \sqrt{\frac{1}{n} \left\| \boldsymbol{K}^{(k)} \boldsymbol{\beta}_k \right\|_2^2},$$

where $\boldsymbol{\beta}_k = [\beta_{k,1}, \ldots, \beta_{k,n}]^\top, k = 1, \ldots, d$ are regression coefficient vectors, $\beta_{k,j}$ is a coefficient for $[K(x_{k,1}, x_{k,j}), \ldots, K(x_{k,n}, x_{k,j})]^\top$, and $\lambda > 0$ is a regularization parameter. This problem can be efficiently solved by the back-fitting algorithm (Ravikumar et al., 2009). Note that SpAM is closely related to the hierarchical multiple kernel learning (Bach, 2009), which employs a sparse additive model with an alternative sparsity-inducing regularizer.

An advantage of SpAM is that it is a convex method and can be efficiently optimized by the backfitting algorithm. Moreover, statistical properties of the SpAM estimator are well studied (Ravikumar et al., 2009). A potential weakness of SpAM is that it can deal only with additive models. That is, if data follow a nonadditive model, SpAM may not work well (see Figure 1b). Another weakness of SpAM is that it needs to optimize *nd* variables, while the proposed methods contain only *d* variables. Thus, SpAM optimization tends to be computationally more expensive than the proposed methods. Finally, an output *y* should be a real number in SpAM, meaning that SpAM cannot deal with structured outputs such as multilabel and graph data.

4 Experiments

In this section, we experimentally investigate the performance of the proposed and existing feature selection methods using synthetic and real-world data sets.

4.1 Setup. We compare the performance of the proposed methods with mRMR (Peng et al., 2005), QPFS (Rodriguez-Lujan et al., 2010), cKTA (Cortes et al., 2012), forward selection with HSIC (FHSIC), FVM (Li et al., 2006), and SpAM (Ravikumar et al., 2009).[4] Note that since it has been reported that the performance of FHSIC is comparable to HSFS and HSFS is computationally expensive for high-dimensional data, we decided to compare the proposed method only to FHSIC. For FVM, QPFS, and mRMR, the C++ implementation of a mutual information estimator is used.[5] Then a QP solver SeDuMi is used to solve the FVM and QPFS optimization problems.[6] We observed that

---

[4]We thank the authors of Ravikumar et al. (2009) for providing us the code used in their paper.

[5]http://penglab.janelia.org/proj/mRMR/.

[6]http://sedumi.ie.lehigh.edu/.

(a) Data1 (Additive model)

(b) Data2 (Non-additive model)

(c) Computation time for Data2

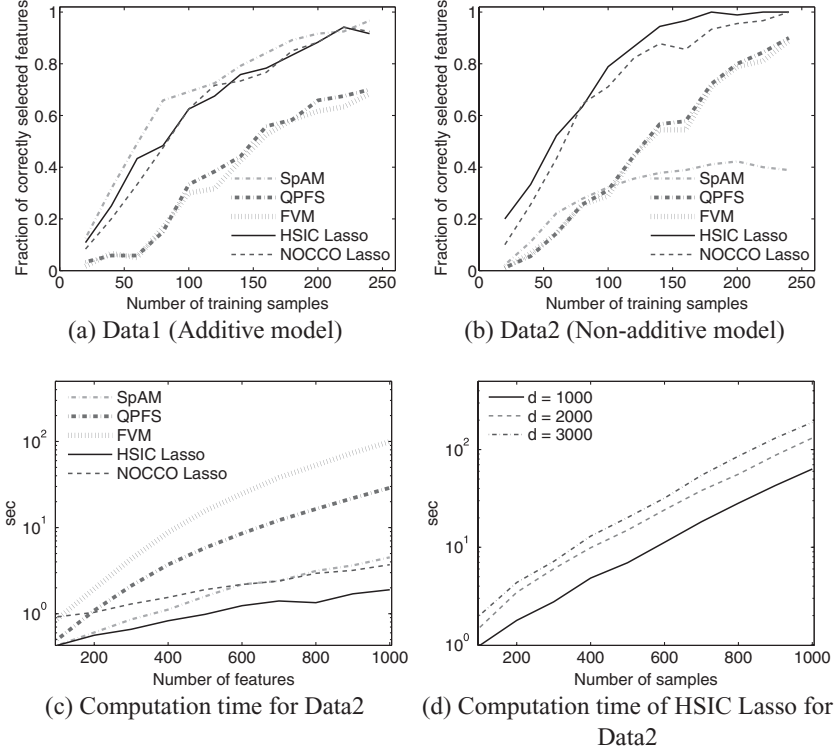(d) Computation time of HSIC Lasso for Data2

Figure 1: (a, b) Feature selection results for artificial data sets over 30 runs. The horizontal axis denotes the number of training samples, and the vertical axis denotes the fraction of correctly selected features. In HSIC Lasso and NOCCO Lasso, the regularization parameter λ is set so that the number of nonzero coefficients is in $\{d^*, d^* + 1, \ldots, d^* + 10\}$ where d^* is the number of true features. Then we use top d^* features by ranking regression coefficients. In QPFS, FVM, and SpAM, we use top d^* features by ranking coefficients. (c) Comparison of computation time for Data2. The horizontal axis denotes the number of entire features d, and the vertical axis denotes the computation time in log-scale. (d) Comparison of computation time for Data2 for HSIC Lasso. The horizontal axis denotes the number of training samples n, and the vertical axis denotes the computation time in log-scale.

the matrices D in FVM, QPFS, and cKTA tend not to be positive definite. In our experiments, we added a small constant to the diagonal elements of D so that the objective function becomes convex. For all experiments, we set $\lambda = 1$ in FVM, $\gamma = 0.5$ in QPFS and cKTA, and $\epsilon = 10^{-3}$ in NOCCO Lasso. For proposed methods, we experimentally use $\sigma_x = 1$.

4.2 Synthetic Data Sets. First, we illustrate the behavior of the proposed HSIC Lasso and NOCCO Lasso using the following two synthetic data sets:

- **Data1 (additive model):**

$$Y = -2\sin(2X_1) + X_2^2 + X_3 + \exp(-X_4) + E,$$

  where $(X_1, \ldots, X_{256})^\top \sim N(\mathbf{0}_{256}, \mathbf{I}_{256})$ and $E \sim N(0, 1)$. Here, $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the multivariate gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$.
- **Data2 (nonadditive model):**

$$Y = X_1 \exp(2X_2) + X_3^2 + E,$$

  where $(X_1, \ldots, X_{1000})^\top \sim N(\mathbf{0}_{1000}, \mathbf{I}_{1000})$ and $E \sim N(0, 1)$.

Figure 1 shows the feature selection accuracy of each method over 30 runs as functions of the number of samples, where the accuracy is measured by the fraction of correctly selected features under the assumption that the number of true features is known. As the figure clearly shows, the proposed HSIC Lasso and NOCCO Lasso methods select good features in both additive and nonadditive model cases. SpAM also works very well for Data1, but it performs poorly for Data2 because the additivity assumption is violated in Data2. QPFS and FVM behave similarly but tend to be outperformed by the proposed methods.

Next, we compare the computation time of each method. Here, we change the number of features in Data2 to $d = 100, 200, \ldots, 1000$, while we fix the number of samples to $n = 100$. Figure 1(c) shows the average computation time for each method over 30 runs. As can be observed, the computation time of HSIC Lasso and NOCCO Lasso increases mildly with respect to the number of features compared to that of SpAM, FVM, and QPFS. Figure 1(d) shows the average computation time of HSIC Lasso for 30 runs. In this experiment, we use the table lookup trick for HSIC Lasso to deal with a relatively large number of samples. This shows that the lookup trick allows us to handle relatively large data sets.

4.3 Real-World Data Sets. Next, we compare the performance of feature selection methods using real-world data sets.

4.3.1 Multi-Class Classification. We use four image data sets and two microarray datasets.[7] Detailed information on the data sets is summarized in Table 2.

In this experiment, we use 80% of samples for training and the rest for testing. We repeat the experiments 100 times by randomly shuffling training

---

[7]http://featureselection.asu.edu/datasets.php.

Table 2: Summary of Real-World Data Sets.

Type	Data Set	Number of Features (d)	Number of Samples (n)	Number of Classes
Image	AR10P	2400	130	10
	PIE10P	2400	210	10
	PIX10P	10,000	100	10
	ORL10P	10,000	100	10
Microarray	TOX	5748	171	4
	CLL-SUB	11340	111	3

and test samples and evaluate the performance of feature selection methods by the average classification accuracy. We use multiclass ℓ_2-regularized kernel logistic regression (KLR) (Hastie, Tibshirani, & Friedman, 2001; Yamada, Sugiyama, & Matsui, 2010) with the gaussian kernel for evaluating the classification accuracy when the top $m = 10, 20, \ldots, 50$ features selected by each method are used. In this letter, we first choose 50 features and then use the top $m = 10, 20, \ldots, 50$ features having the largest absolute regression coefficients. In KLR, all tuning parameters such as the gaussian width and the regularization parameter are chosen based on three-fold cross-validation.

We also investigate the redundancy rate (RED) (Zhao, Wang, & Li, 2010),[8]

$$\text{RED} = \frac{1}{m(m-1)} \sum_{u_k, u_j, k > l} |\rho_{k,l}|,$$

where $\rho_{k,l}$ is the correlation coefficient between the kth and lth features. A large RED score indicates that selected features are more strongly correlated to each other, that is, many redundant features are selected. Thus, as a feature selection method, a small redundancy rate is preferable.

Results. Figure 2 shows the mean classification accuracy over 100 runs as functions of the number of selected features. Table 3 shows the average classification accuracy rates for the top $m = 50$ features selected by each method. In this experiment, since the computation cost of FVM was too high for data sets with a large number of features, we included the FVM results only for the data sets with a small number of features (i.e., AR10P, PIE10P, and TOX). The graphs in Figure 2 clearly show that HSIC Lasso and NOCCO Lasso compare favorably with existing methods for the image data sets (i.e., AR10P, PIE10P, PIX10P, and ORL10P) in terms of the classification

---

[8]The original redundancy rate was defined with a plain correlation coefficient (Pearson, 1920), not the absolute correlation coefficient (Zhao et al., 2010). However, this is not appropriate as an error metric because negative correlation decreases RED. For this reason, we decided to use the absolute correlation coefficient.
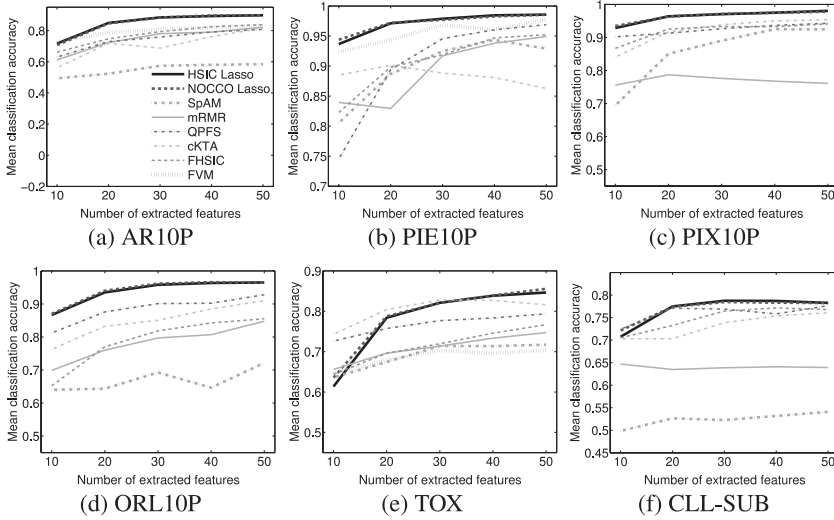
Figure 2: Mean classification accuracy for real-world data. The horizontal axis denotes the number of selected features, and the vertical axis denotes the mean classification accuracy.

Table 3: Mean Classification Accuracy.

Data Set	HSIC Lasso	NOCCO Lasso	SpAM	FVM	mRMR	QPFS	cKTA
AR10P	**.85** (.11)	**.85** (.11)	.55 (.11)	.80 (.12)	.75 (.14)	.74 (.14)	.71 (.21)
PIE10P	**.97** (.03)	**.97** (.03)	.90 (.11)	.96 (.06)	.90 (.12)	.95 (.07)	.88 (.22)
PIX10P	**.96** (.04)	**.96** (.04)	.86 (.15)	— (—)	.77 (.12)	.92 (.07)	.92 (.09)
ORL10P	**.94** (.07)	**.94** (.07)	.67 (.12)	— (—)	.78 (.14)	.88 (.10)	.85 (.14)
TOX	.78 (.12)	.79 (.11)	.69 (.09)	.69 (.09)	.71 (.08)	.77 (.08)	**.80** (.11)
CLL-SUB	**.77** (.09)	**.77** (.08)	.52 (.11)	— (—)	.64 (.10)	.76 (.12)	.71 (.10)

Note: Standard deviations in brackets for real-world data.

accuracy, and they are comparable to existing methods for the microarray data sets (i.e., TOX and CLL-SUB).

Table 4 shows the RED values for the top $m = 50$ features selected by each method. As can be observed, HSIC Lasso and NOCCO Lasso tend to have smaller RED values, and thus they select less redundant features.

Role of the Gaussian Width and the Output Kernel. In the proposed methods, the gaussian width and the output kernel $L(y, y')$ must be chosen manually. We carried out a set of experiments to show the sensitivity of choosing the gaussian width and the output kernel in Figures 3 and 4. Note that since the

Table 4: Mean Redundancy Rate for Real-World Data.

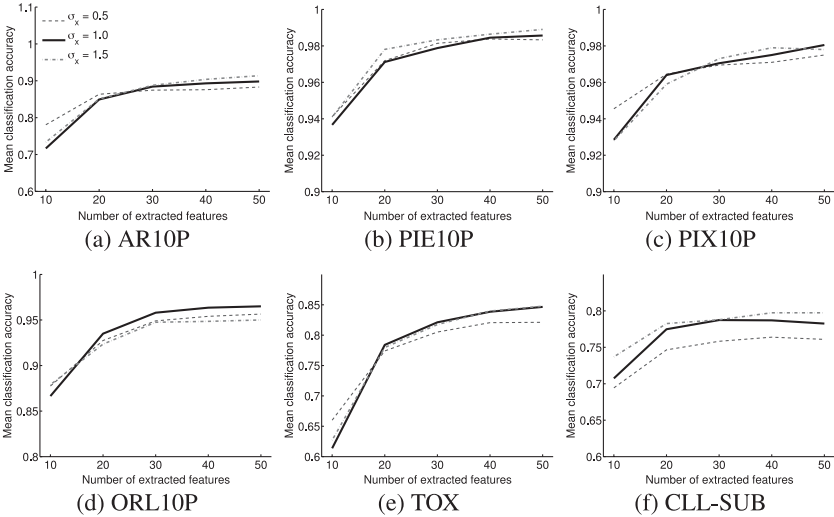| Data Set	HSIC Lasso	NOCCO Lasso	SpAM	FVM	mRMR	QPFS	cKTA
AR10P	**.20** (.03)	**.20** (.03)	.26 (.04)	.26 (.04)	.27 (.04)	.22 (.05)	.24 (.03)
PIE10P	**.14** (.01)	**.14** (.02)	.25 (.042)	.19 (.03)	.23 (.04)	.18 (.03)	.16 (.02)
PIX10P	.18 (.02)	**.17** (.02)	.34 (.11)	— (—)	.20 (.07)	.29 (.06)	.20 (.04)
ORL10P	**.19** (.03)	**.19** (.03)	.30 (.05)	— (—)	.29 (.10)	.20 (.03)	.19 (.03)
TOX	.38 (.03)	.38 (.03)	.39 (.03)	.42 (.03)	.39 (.03)	.38 (.03)	**.37** (.04)
CLL-SUB	.34 (.03)	.35 (.03)	.40 (.06)	— (—)	.33 (.04)	.32 (.03)	**.28** (.05)

Note: Standard deviations in brackets.

Figure 3: Mean classification accuracy of HSIC Lasso with different gaussian widths. The horizontal axis denotes the number of selected features, and the vertical axis denotes the mean classification accuracy.

performances of HSIC Lasso and NOCCO Lasso are comparable, we here evaluate only HSIC Lasso. As can be seen in Figure 3, the proposed method is not so sensitive to the gaussian width. From the output kernel comparison in Figure 4, we found that HSIC Lasso with delta kernel clearly outperforms that with gaussian kernel. Thus, using different input and output kernels is important for feature selection in classification scenarios.

   4.3.2 High-Dimensional Regression. We also evaluate our proposed method with the Affymetric GeneChip Rat Genome 230 2.0 Array data
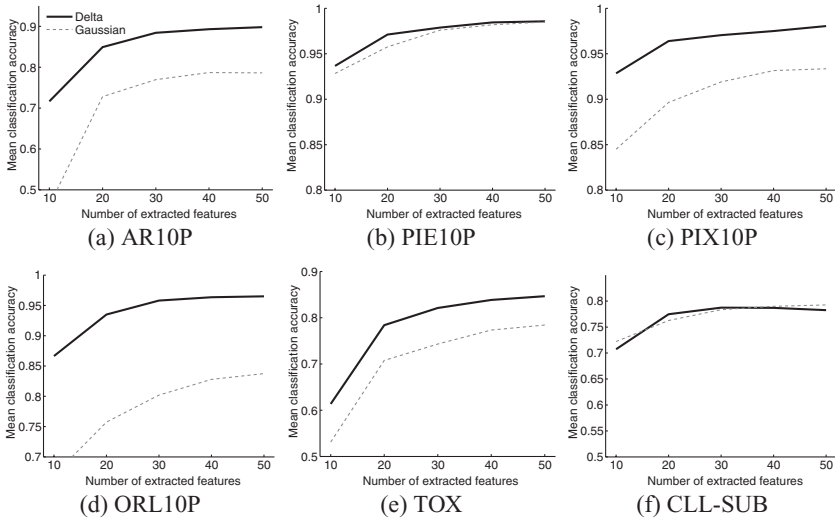
Figure 4: Mean classification accuracy of HSIC Lasso with different output kernels. Here, we use the delta kernel and the gaussian kernel. The horizontal axis denotes the number of selected features, and the vertical axis denotes the mean classification accuracy.

set (Scheetz et al., 2006). The data set consists of 120 rat subjects with 31098 genes, which were measured from eye tissue. Similar to Huang et al. (2010), we focus on finding genes that are related to the TRIM32 gene, which was recently found to cause the Bardet-Biedl syndrome. Note that since TRIM32 takes real values, this is a regression problem.

In this experiment, we use 80% of samples for training and the rest for testing. We repeat the experiments 100 times by randomly shuffling training and test samples and evaluate the performance of feature selection methods by the mean squared error. In addition, we use the correlation coefficient between the predicted and the true TRIM32 values, a popular performance metric in biology community. We use kernel regression (KR) (Schölkopf and Smola, 2002) with the gaussian kernel for evaluating the mean squared error and the mean correlation when the top $m = 10, 20, \ldots, 50$ features selected by each method are used. We first choose 50 features and then use top $m = 10, 20, \ldots, 50$ features having the largest absolute regression coefficients. In KR, all tuning parameters such as the gaussian width and the regularization parameter are chosen based on three-fold cross-validation.

Results. Figure 5 shows the mean squared error and the mean correlation coefficient over 100 runs as functions of the number of selected features. As can be observed, the proposed HSIC Lasso and NOCCO Lasso compare favorably with existing methods.

(a) MSE

(b) MSE (HSIC Lasso)

(c) MSE (NOCCO Lasso)

(d) Mean correlation
coefficient

(e) Mean correlation
coefficient (HSIC Lasso)

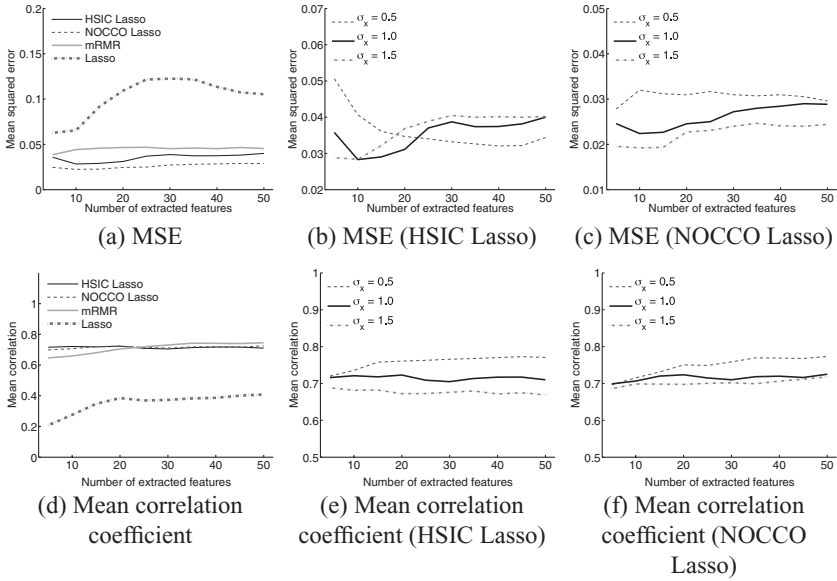(f) Mean correlation
coefficient (NOCCO
Lasso)

Figure 5: (a) Mean squared error for the biology data. The horizontal axis denotes the number of selected features and the vertical axis the mean squared error. Here, we use the kernel parameter $\sigma_x = 1.0$ for both HSIC Lasso and NOCCO Lasso. (b, c) Mean squared error of HSIC Lasso and NOCCO Lasso with respect to different kernel parameters. (d) Mean correlation for the biology data. The horizontal axis denotes the number of selected features and the vertical axis the mean correlation. (e, f) Mean correlation coefficient of HSIC Lasso and NOCCO Lasso with respect to different kernel parameters. The average redundancy rate of HSIC Lasso, NOCCO Lasso, mRMR, and Lasso are 0.44, 0.45, 0.42, and 0.43.

5 Conclusion

In this letter, we proposed novel nonlinear feature selection methods called HSIC Lasso and NOCCO Lasso. In the proposed methods, global optimal solutions can be obtained by solving a Lasso optimization problem with a nonnegativity constraint, which can be efficiently performed by the dual augmented Lagrangian algorithm (Tomioka et al., 2011). Furthermore, the proposed methods have clear statistical interpretation that nonredundant features with strong statistical dependence on output values can be found via kernel-based independence measures (Gretton et al., 2005; Fukumizu et al., 2008). We applied the proposed methods to real-world image and biological feature selection tasks and experimentally showed that they are promising.

The usefulness of the proposed method will be investigated on more real-world applications such as computer vision, bioinformatics, and speech and signal processing in our future work. Moreover, extending the proposed model to multitask learning and prediction and investigating theoretical properties of the proposed formulation are important issues to investigate.

Acknowledgments

References

Bach, F. (2008). Bolasso: Model consistent lasso estimation through the bootstrap. In *International Conference on Machine learning (ICML)* (pp. 33–40). New York: ACM.

Bach, F. (2009). Exploring large feature spaces with hierarchical multiple kernel learning. In D. Koller, D. Schuurmans, Y. Bengio, & L. Bottou (Eds.), *Advances in neural information processing systems, 21* (pp. 105–112). Cambridge, MA: MIT Press.

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge: Cambridge University Press.

Combettes, P. L., & Wajs, V. R. (2005). Signal recovery by proximal forward-backward splitting. *Multiscale Modeling and Simulation*, 4(4), 1168–1200.

Cortes, C., Mohri, M., & Rostamizadeh, A. (2012). Algorithms for learning kernels based on centered alignment. *Journal of Machine Learning Research*, 13, 795–828.

Cover, T. M., & Thomas, J. A. (2006). *Elements of information theory* (2nd ed.). New York: Wiley.

Daubechies, I., Defrise, M., & De Mol, C. (2004). An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57(11), 1413–1457.

Ding, C., & Peng, H. (2005). Minimum redundancy feature selection from microarray gene expression data. *Journal of Bioinformatics and Computational Biology*, 3(2), 185–205.

Forman, G. (2008). BNS feature scaling: An improved representation over TF-IDF for SVM text classification. In *ACM Conference on Information and Knowledge Management* (pp. 263–270). New York: ACM.

Fukumizu, K., Gretton, A., Sun, X., & Schölkopf, B. (2008). Kernel measures of conditional dependence. In J. Platt, D. Koller, Y. Singer, & S. Roweis (Eds.), *Advances in Neural Information Processing Systems, 20* (pp. 489–496). Cambridge, MA: MIT Press.

Gretton, A., Bousquet, O., Smola, A., & Schölkopf, B. (2005). Measuring statistical dependence with Hilbert-Schmidt norms. In *Algorithmic learning theory (ALT)* (pp. 63–77). New York: Springer.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning: Data mining, inference, and prediction*. New York: Springer.

Huang, J., Horowitz, J. L., & Wei, F. (2010). Variable selection in nonparametric additive models. *Annals of Statistics*, *38*(4), 2282–2313.

Kim, S.-J., Koh, K., Lustig, M., Boyd, S., & Gorinvesky, D. (2007). An interior-point method for large-scale ℓ_1-regularized least squares. *IEEE Journal of Selected Topics in Signal Processing*, *1*(4), 606–617.

Li, F., Yang, Y., & Xing, E. (2006). From Lasso regression to feature vector machine. In Y. Weiss, B. Schölkopf, & J. Platt (Eds.), In *Advances in neural information processing systems, 18* (pp. 779–786). Cambridge, MA: MIT Press.

Liu, H., Lafferty, J., & Wasserman, L. (2009). Nonparametric regression and classification with joint sparsity constraints. In D. Koller, D. Schuurmans, Y. Bengio, & L. Bottou (Eds.), *Advances in neural information processing systems, 21* (pp. 969–976). Cambridge, MA: MIT Press.

Masaeli, M., Fung, G., & Dy, J. G. (2010). From transformation-based dimensionality reduction to feature selection. In *International Conference on Machine learning (ICML)* (pp. 751–758). Madison, WI: Omnipress.

Meier, L., Van De Geer, S., & Bühlmann, P. (2008). The group lasso for logistic regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *70*(1), 53–71.

Meinshausen, N., & Bühlmann, P. (2010). Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *72*(4), 417–473.

Nocedal, J., & Wright, S. (2003). *Numerical optimization* (2nd ed.). New York: Springer-Verlag.

Pearson, K. (1920). Notes on the history of correlation. *Biometrika*, *13*(1), 25–45.

Peng, H., Long, F., & Ding, C. (2005). Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*, 1226–1237.

Raskutti, G., Wainwright, M., & Yu, B. (2012). Minimax-optimal rates for sparse additive models over kernel classes via convex programming. *Journal of Machine Learning Research*, *13*, 389–427.

Ravikumar, P., Lafferty, J., Liu, H., & Wasserman, L. (2009). Sparse additive models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *71*(5), 1009–1030.

Rodriguez-Lujan, I., Huerta, R., Elkan, C., & Cruz, C. S. (2010). Quadratic programming feature selection. *Journal of Machine Learning Research (JMLR)*, *11*, 1491–1516.

Roth, V. (2004). The generalized Lasso. *IEEE Transactions on Neural Networks*, *15*(1), 16–28.

Scheetz, T. E., Kim, K.-Y. A., Swiderski, R. E., Philp, A. R., Braun, T. A., Knudtson, K. L., et al. (2006). Regulation of gene expression in the mammalian eye and its relevance to eye disease. *Proceedings of the National Academy of Sciences*, *103*, 14429–14434.

Schölkopf, B., & Smola, A. J. (2002). *Learning with kernels*. Cambridge, MA: MIT Press.

Seeger, M. (2002). Covariance kernels from Bayesian generative models. In T. K. Leen, T. G. Dietterich, & V. Tresp (Eds.), *Advances in neural information processing systems* (pp. 905–912). Cambridge, MA: MIT Press.

Shawe-Taylor, N., & Kandola, A. (2002). On kernel target alignment. In T. K. Leen, T. G. Dietterich, & V. Tresp (Eds.), *Advances in neural information processing systems, 13* (pp. 367–373).

Shenoy, P., Miller, K. J., Crawford, B., & Rao, R. N. (2008). Online electromyographic control of a robotic prosthesis. *IEEE Transactions on Biomedical Engineering*, *55*(3), 1128–1135.

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

Steinwart, I. (2001). On the influence of the kernel on the consistency of support vector machines. *Journal of Machine Learning Research*, *2*, 67–93.

Suzuki, T., & Sugiyama, M. (2013). Fast learning rate of multiple kernel learning: Trade-off between sparsity and smoothness. *Annals of Statistics*, *41*, 1381–1405.

Suzuki, T., Sugiyama, M., Kanamori, T., & Sese, J. (2009). Mutual information estimation reveals global associations between stimuli and biological processes. *BMC Bioinformatics, 10*, S52.

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

Tomioka, R., Suzuki, T., & Sugiyama, M. (2011). Super-linear convergence of dual augmented Lagrangian algorithm for sparsity regularized estimation. *Journal of Machine Learning Research*, *12*, 1537–1586.

Wright, S. J., Nowak, R. D., & Figueiredo, M. A. T. (2009). Sparse reconstruction by separable approximation. *IEEE Transactions on Signal Processing*, *57*(7), 2479–2493.

Xing, E. P., Jordan, M. I., & Karp, R. M. (2001). Feature selection for high-dimensional genomic microarray data. In *International Conference on Machine Learning (ICML)* (pp. 601–608). Madison, WI: Omnipress.

Yamada, M., Sugiyama, M., & Matsui, T. (2010). Semi-supervised speaker identification under covariate shift. *Signal Processing*, *90*(8), 2353–2361.

Yin, W., Osher, S., Goldfarb, D., & Darbon, J. (2008). Bregman iterative algorithms for L1-minimization with applications to compressed sensing. *SIAM Journal of Imaging Sciences*, *1*(1), 143–168.

Zhao, Z., Wang, L., & Li, H. (2010). Efficient spectral feature selection with minimum redundancy. In *AAAI Conference on Artificial Intelligence (AAAI)* (pp. 673–678). Cambridge, MA: MIT Press.

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *67*(2), 301–320.