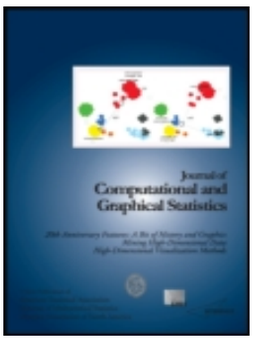

Journal of Computational and Graphical Statistics

ISSN: 1061-8600 (Print) 1537-2715 (Online) Journal homepage: http://www.tandfonline.com/loi/ucgs20

Automatic Feature Selection via Weighted Kernels and Regularization

Genevera I. Allen

To cite this article: Genevera I. Allen (2013) Automatic Feature Selection via Weighted Kernels and Regularization, Journal of Computational and Graphical Statistics, 22:2, 284-299, DOI: 10.1080/10618600.2012.681213

To link to this article: http://dx.doi.org/10.1080/10618600.2012.681213

Supplementary materials for this article are available online.
Please go to www.tandfonline.com/r/JCGS

Automatic Feature Selection via Weighted Kernels and Regularization

Genevera I. ALLEN

Selecting important features in nonlinear kernel spaces is a difficult challenge in both classification and regression problems. This article proposes to achieve feature selection by optimizing a simple criterion: a feature-regularized loss function. Features within the kernel are weighted, and a lasso penalty is placed on these weights to encourage sparsity. This feature-regularized loss function is minimized by estimating the weights in conjunction with the coefficients of the original classification or regression problem, thereby automatically procuring a subset of important features. The algorithm, KerNel Iterative Feature Extraction (KNIFE), is applicable to a wide variety of kernels and high-dimensional kernel problems. In addition, a modification of KNIFE gives a computationally attractive method for graphically depicting nonlinear relationships between features by estimating their feature weights over a range of regularization parameters. The utility of KNIFE in selecting features through simulations and examples for both kernel regression and support vector machines is demonstrated. Feature path realizations also give graphical representations of important features and the nonlinear relationships among variables. Supplementary materials with computer code and an appendix on convergence analysis are available online.

Key Words: Reproducing kernel Hilbert space; Kernel ridge regression; Lasso; Nonlinear regression; Regularization paths; Support vector machine.

1. INTRODUCTION

The merits of L_1-regularized linear regression and classification are well known. Regression methods such as the lasso and the elastic net, and classification methods such as the L_1-SVM (support vector machine) and the L_1-logistic regression enjoy immense popularity due to their ability to select important variables in high dimensions. The L_1-norm penalty, as the direct convex relaxation of the L_0-norm or best subsets penalty, also has many desirable theoretical properties (Candes and Plan 2009). In addition, the linear L_1-norm regularized problem has an appealing form that is simply an extension of the original problem formulation, namely a loss function plus an L_1 norm on the coefficients.

Genevera I. Allen, Department of Pediatrics-Neurology, Baylor College of Medicine, Jan and Dan Duncan Neurological Research Institute, Texas Children's Hospital, Houston, TX 77030, and Department of Statistics, Rice University, Houston, TX 77005 (E-mail: *gallen@rice.edu*).

Journal of Computational and Graphical Statistics, Volume 22, Number 2, Pages 284–299
DOI: 10.1080/10618600.2012.681213

This direct approach via the L_1-norm penalty gives a disciplined method and criterion for feature selection in linear problems.

In nonlinear problems, however, and especially kernel regression and classification problems, the L_1-norm penalty has had limited use due to numerical and computational challenges. Instead, indirect or heuristic methods are often employed. Filtering methods and subset methods, especially the popular recursive feature elimination (RFE) (Guyon et al. 2002), which removes features in a backward stepwise manner, are the most commonly used methods. Other methods such as RODEO (Lafferty and Wasserman 2008) and the method of Bertin and Lecué (2008) perform feature selection for nonlinear functions via local regressions. The latter uses a lasso penalty to select features in a local polynomial kernel. The COSSO (Component Selection and Smoothing Operator) method (Lin and Zhang 2007) also uses an L_1 penalty on the features, estimated in conjunction with smoothing splines. In addition, there are several regularization methods that perform feature selection specific to the linear SVM (Zhu et al. 2003; Bach, Lanckriet, and Jordan 2004; Lanckriet et al. 2004; Neumann, Schnörr, and Steidl 2005; Xu et al. 2009), and some also that can be adapted for kernel SVMs (Weston et al. 2000; Guyon 2003; Navot and Tishby 2004; Wang 2008). Many of the former, however, do not directly incorporate feature selection into the original problem formulation.

Weighting features within the kernel to achieve feature selection has been proposed in several methods (Weston et al. 2000; Grandvalet and Canu 2002; Navot and Tishby 2004; Argyriou et al. 2006; Li, Yang, and Xing 2006; Cao et al. 2007). Most of these, however, do not directly optimize the original regression or classification problem, but instead seek to find a good set of weights on the features for later use within the kernel of the model. Grandvalet and Canu (2002), however, formulated a direct optimization problem for the SVM. They placed an L_p-norm penalty on the feature weights, and developed an algorithm for the L_2-norm penalty with nonnegative feature weights. An L_2 penalty, however, often does not encourage sufficient sparsity in the feature weights. They commented that an L_1 penalty would achieve greater feature selection, but never pursue this approach algorithmically, as the problem is nonconvex and hence is not conducive to simple optimization. This, however, is the approach that has been employed here.

The regularization approach to kernel feature selection discussed here is summarized by giving the simple optimization criterion. Given a response, $\mathbf{Y}$, a feature-weighted kernel, $\mathbf{K}_\mathbf{w}$ (defined in Section 2), coefficients, $\boldsymbol{\alpha}$, and feature weights, $\mathbf{w}$ (defined in Section 2), the following criterion is optimized:

$$\begin{array}{lccccc} \text{minimize}_{\boldsymbol{\alpha},\mathbf{w}} & L(\mathbf{Y}, \mathbf{K}_\mathbf{w}\boldsymbol{\alpha}) & + & \lambda_1 \boldsymbol{\alpha}^T \mathbf{K}_\mathbf{w} \boldsymbol{\alpha} & + & \lambda_2 ||\mathbf{w}||_1 \\ & \text{loss function} & + & \text{coefficient regularization} & + & \text{feature-weighted regularization} \end{array} \tag{1}$$

Thus, feature selection is integrated into the original problem formulation through the feature-weighted loss function, with an L_1 penalty on the feature weights. Selecting relevant features is achieved automatically by optimizing this criterion.

The main contribution of this article is a novel algorithm that finds a local minimum (1) for general kernel problems, meaning that it is applicable to any kernel problem that can be formulated via a loss function. These include, but are not limited to kernel ridge regression, kernel SVMs, kernel logistic regression, kernel discriminant analysis, and kernel principal components analysis. As an extension, a method for visualizing nonlinear

relationships between features by estimating the feature weights over a range of regularization parameters is given, similar to coefficient regularization paths. Hence, a disciplined, automatic feature selection in high-dimensional kernel regression and classification problems is achieved.

The article is organized as follows. In Section 2, weighted kernels and the optimization problem studied here along with its mathematical and computational challenges are discussed. The main algorithm, KerNel Iterative Feature Extraction (KNIFE), is presented in Section 3 and minimization through kernel linearization is discussed. (Convergence results and further technical properties of KNIFE are given in the Supplementary Materials.) An extension of KNIFE is presented to visualizing nonlinear feature relationships in Section 3.4. Section 4 gives simulation results and real data examples on gene selection for microarray data and feature selection in vowel recognition and ozone prediction data. Section 5 concludes with a discussion.

2. KNIFE PROBLEM

This article proposes to select important features by forming a regularized loss function that involves a set of weights on the features within a kernel. Before establishing the KNIFE problem, the need for feature selection in nonlinear kernel regression and classification problems is briefly motivated.

Many have noted that nonlinear kernel methods perform particularly poorly when irrelevant features are present (Weston et al. 2000; Grandvalet and Canu 2002; Navot and Tishby 2004; Lin and Zhang 2007; Bertin and Lecué 2008; Lafferty and Wasserman 2008; Wang 2008). Consider this mathematically, using the example of the SVM with polynomial kernels. Given data $\mathbf{x}_i \in \Re^p$ for $i = 1, \ldots, n$ observations and p features with the response $\mathbf{Y} \in \{-1, 1\}^n$, the kernel matrix, $\mathbf{K} \in \Re^{n \times n}$, is defined by $\mathbf{K}(i, i') = k(\mathbf{x}_i, \mathbf{x}_{i'}) = \left(\sum_{j=1}^p x_{ij} x_{i'j} + 1\right)^d$, for example, with polynomial kernels. Recall that the SVM, presented in its primal form, can be written as an unconstrained minimization problem with the hinge loss (Wahba, Lin, and Zhang 1999): $\text{minimize}_{\boldsymbol{\alpha}, \alpha_0} \sum_{i=1}^n [1 - y_i \alpha_0 - y_i (\mathbf{K}\boldsymbol{\alpha})_i]_+ + \lambda \boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha}$. Here, the coefficients are $\boldsymbol{\alpha} \in \Re^n$ and $\alpha_0 \in \Re$. Note that each feature is given the same weight in the kernel matrix, thus explaining the poor performance of SVMs for data with many irrelevant features. It is, then, proposed to place feature weights within the kernels to differentiate between the true and the noise features.

2.1 FEATURE-WEIGHTED KERNELS

Data $\mathbf{x}_i \in \Re^p$ for $i = 1, \ldots, n$ observations that can be written as a data matrix $\mathbf{X} \in \Re^{n \times p}$ are observed. It is assumed that $\mathbf{x}_i$ is standardized so that it has mean zero and variance one. For regression, a response, $\mathbf{Y} \in \Re^n$, and for classification, $\mathbf{Y} \in \{-1, 1\}^n$ are considered. The feature-weighted kernels discussed here place a weight, $\mathbf{w} \in \Re^{p+}$, on each feature of the data within the kernel. Some examples for three common kernels are inner-product kernel, $k_{\mathbf{w}}(\mathbf{x}, \mathbf{x}') = \sum_{j=1}^p (w_j x_j)(w_j x'_j)$, Gaussian or radial kernel, $k_{\mathbf{w}}(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \sum_{j=1}^p (w_j x_j - w_j x'_j)^2)$, and polynomial kernel, $k_{\mathbf{w}}(\mathbf{x}, \mathbf{x}') = (\sum_{j=1}^p (w_j x_j)(w_j x'_j) + c)^d$. With these feature-weighted kernels, one can define the weighted kernel matrix as $\mathbf{K}_{\mathbf{w}} \in \Re^{n \times n}$ such that $\mathbf{K}_{\mathbf{w}}(i, i') = k_{\mathbf{w}}(\mathbf{x}_i, \mathbf{x}_{i'})$. Note that

the weights multiply each data feature in the kernel instead of placing a single feature weight within the kernel. (For example, in linear kernels, $(w_j x_j)(w_j x'_j)$ instead of $w_j x_j x'_j$.) This seemingly minor detail is important for the kernel linearization part of the algorithm discussed in Section 3.2. The weights also play the role of a scaling parameter and a feature weight, similar to the adaptive scaling approach of Grandvalet and Canu (2002).

2.2 KNIFE Optimization Problem

These feature-weighted kernels are incorporated into the regression or classification model. The response $\mathbf{Y}$ is modeled by $\hat{\mathbf{Y}} = f(\mathbf{X})$, where $f(\mathbf{x}_i) = \sum_{i'=1}^{n} \alpha_i \mathbf{K}_w(i, i')$ for regression, or $f(\mathbf{x}_i) = \text{sign}(\alpha_0 + \sum_{i'=1}^{n} \alpha_i \mathbf{K}_w(i, i'))$ for classification with $\boldsymbol{\alpha} \in \Re^n$—the coefficients that must be estimated. For a positive definite kernel, $\mathbf{K}_\mathbf{w}$, and $f(\mathbf{X})$, a member of the reproducing kernel Hilbert space, $\mathcal{H}_{K_\mathbf{w}}$, this problem can be written as a minimization problem of the form: minimize$_{f \in \mathcal{H}_{K_\mathbf{w}}}[L(\mathbf{Y}, f(\mathbf{X})) + \lambda ||f||^2_{\mathcal{H}_{K_\mathbf{w}}}]$, where $L(\mathbf{Y}, f(\mathbf{X}))$ is the loss function. Some common examples of loss functions include the hinge loss (SVM), squared error loss (regression), and binomial deviance loss (logistic regression).

To obtain a selection of important variables in a problem of this form, one needs the weights to be both nonnegative and sparse. To this end, an L_1 penalty is added on the weights and is optimized over the set of nonnegative weights that are less than one. (Note that the magnitude of the weights are limited to ensure nondegenerate scaling between $\boldsymbol{\alpha}$ and $\mathbf{w}$.) This gives the KNIFE optimization problem:

$$\begin{aligned} \underset{\boldsymbol{\alpha}, \mathbf{w}}{\text{minimize}} \quad & f(\boldsymbol{\alpha}, \mathbf{w}) = L(\mathbf{Y}, \mathbf{K}_\mathbf{w}\boldsymbol{\alpha}) + \lambda_1 \boldsymbol{\alpha}^T \mathbf{K}_\mathbf{w} \boldsymbol{\alpha} + \lambda_2 \mathbf{1}^T \mathbf{w} \\ \text{subject to} \quad & 0 \le w_j < 1, \text{ for all } j = 1, \ldots, p. \end{aligned} \tag{2}$$

Here, λ_1 and λ_2 are regularization parameters such that $\lambda_1 > 0$ and $\lambda_2 \ge 0$.

The KNIFE optimization problem, (2), is nonconvex and it is therefore difficult to find a minimum, even for problems of small dimensions. One could approach this as a difference of convex programming problem (convex-concave programming), a direction taken by Argyriou et al. (2006). This approach splits the optimization problem with respect to $\mathbf{w}$ into a convex part associated with positive α_i's and a concave part associated with negative α_i's. It then optimizes the criterion by majorization minimization (MM), with full optimization achieved by alternating between MM steps using a cutting plane method to estimate $\mathbf{w}$ and traditional kernel regression or classification to estimate $\boldsymbol{\alpha}$. Thus, solving (2), is computationally prohibitive in high-dimensional settings.

This article seeks to optimize the feature-regularized loss function with two main objectives: (1) to give a computationally tractable algorithm for nonlinear feature selection in high dimensions that leads to (2) an efficient method of visualizing nonlinear relationships among the features and the response. To achieve these, circumventing the computational challenges of minimizing the nonconvex KNIFE problem by linearizing nonlinear kernels with respect to the feature weights is proposed. This approach is discussed in detail in the following sections.

3. KNIFE ALGORITHM

Given the KNIFE optimization problem based on the feature-weighted kernels, an algorithm to minimize the feature-regularized loss function for general kernel classification and regression methods is proposed. The approach is to find a minimum by alternating between minimizing with respect to the coefficients, $\boldsymbol{\alpha}$, and the feature weights, $\mathbf{w}$.

3.1 Linear Kernels

To begin with, the KNIFE algorithm for linear kernels, which forms the foundation of the approach to solving (2) nonlinear kernels, is outlined. With linear kernels, the kernel matrix becomes $\mathbf{K}_\mathbf{w} = \mathbf{X}\mathbf{W}\mathbf{X}^T$, where $\mathbf{W} = \text{diag}(\mathbf{w}^2)$. This gives the following objective function: $f(\boldsymbol{\alpha}, \mathbf{w}) = L(\mathbf{Y}, \mathbf{X}\mathbf{W}\mathbf{X}^T\boldsymbol{\alpha}) + \lambda_1\boldsymbol{\alpha}^T\mathbf{X}\mathbf{W}\mathbf{X}^T\boldsymbol{\alpha} + \lambda_2\mathbf{1}^T\mathbf{w}$. Letting $\boldsymbol{\beta} = \mathbf{X}^T\boldsymbol{\alpha}$, one arrives at $f(\boldsymbol{\beta}, \mathbf{w}) = L(\mathbf{Y}, \mathbf{X}\mathbf{W}\boldsymbol{\beta}) + \lambda_1\boldsymbol{\beta}^T\mathbf{W}\boldsymbol{\beta} + \lambda_2\mathbf{1}^T\mathbf{w}$. Here, note that $f(\boldsymbol{\beta}, \mathbf{w})$ is a biconvex function of $\boldsymbol{\beta}$ and $\mathbf{w}$, meaning that if one fixes $\boldsymbol{\beta}$, $f(\cdot, \mathbf{w})$ is convex in $\mathbf{w}$, and if one fixes $\mathbf{w}$, $f(\boldsymbol{\beta}, \cdot)$ is convex in $\boldsymbol{\beta}$. Recall that $\mathbf{w} < 1$, $\lambda_1 > 0$, and $\lambda_2 \geq 0$ so that the scaling between the parameters $\boldsymbol{\beta}$ and $\mathbf{w}$ does not permit degenerate solutions.

This biconvex property leads to a simple blockwise algorithm for minimization: minimize with respect to $\boldsymbol{\beta}$ and then with respect to $\mathbf{w}$. This block descent algorithm is monotonic, meaning that each iteration decreases the objective function $f(\boldsymbol{\beta}, \mathbf{w})$ and the algorithm converges (supplementary materials).

Hence, for linear kernels, one can obtain a simple algorithm for minimizing the KNIFE problem because of the biconvex property between the coefficients and the feature weights.

3.2 KNIFE Algorithm

For nonlinear kernels, it is proposed to linearize kernels with respect to the feature weights to obtain a function convex in the weights and hence conducive to a simple minimization of the KNIFE objective (2). From the previous section, it was observed that if the kernel is linear in the feature weights, then one can apply a blockwise algorithm, estimating the coefficients and then estimating the feature weights. This, then, is the motivation for the nonlinear kernel algorithm that estimates coefficients, linearizes the kernel to obtain a surrogate objective, and estimates the feature weights from this surrogate function.

The kernels are linearized as follows. Given the current estimate of the weights, $\mathbf{w}^{(t)}$, define the linearized kernel, $\tilde{k}_\mathbf{w}$, as $\tilde{k}_{\mathbf{w}^{(t)}}(i, i') \triangleq k_{\mathbf{w}^{(t-1)}}(i, i') + \bigtriangledown k_{\mathbf{w}^{(t-1)}}(i, i')^T(\mathbf{w}^{(t)} - \mathbf{w}^{(t-1)})$. Note that $\tilde{k}_\mathbf{w}(i, i')$ is the linearization of the (i, i')th element of the kernel matrix $\mathbf{K}_\mathbf{w}$. Here, $\mathbf{w}^{(t-1)}$ is the weight vector from the previous iteration of the algorithm.

As previously mentioned in Section 2.1, a weight on each data feature is placed within the kernel. This is integral to the linearization step discussed here. The advantages can be seen with an example of the gradient of a polynomial kernel: $\bigtriangledown k_{\mathbf{w}^{(t-1)}}(i, i')_k = 2dw_k^{(t-1)}x_{ik}x_{i'k}(\sum_{j=1}^p (w_j^{(t-1)})^2 x_{ij}x_{i'j} + 1)^{d-1}$. Note that the gradient is scaled by the weights of the previous iteration, $\mathbf{w}^{(t-1)}$. Thus, if several weights were previously set to zero, the gradient in those directions is zero, meaning that the weights will remain zero in all subsequent iterations of the algorithm (i.e., the weights are *sticky* at zero). This property, first, maintains sparsity in the feature weights throughout the algorithm, and second, limits the number of directions in which the weight vector can move in succeeding iterations. The

former property allows one to approximate continuous feature paths (see Section 3.4) and the latter property can be critical to algorithm convergence (supplementary materials).

With this linearization step, a surrogate objective function, $\tilde{f}(\boldsymbol{\alpha}, \mathbf{w})$, that is convex in the feature weights is formed. This surrogate function is defined as follows: let $\mathbf{B} \in \Re^{n \times n} : \mathbf{B}_{ii'} \triangleq k_{\mathbf{w}^{(t-1)}}(i, i') - \nabla k_{\mathbf{w}^{(t-1)}}(i, i')^T \mathbf{w}^{(t-1)}$, and $\mathbf{A} \in \Re^{n \times p} : \mathbf{A}_{ii'} \triangleq \sum_{i'=1}^{n} \alpha_{i'} \bigtriangledown k_{\mathbf{w}^{(t-1)}}(i, i')^T$, then the surrogate objective is $\tilde{f}(\boldsymbol{\alpha}, \mathbf{w}) \triangleq L(\mathbf{Y}, \mathbf{B}\boldsymbol{\alpha} + \mathbf{A}\mathbf{w}) + \lambda_1 \boldsymbol{\alpha}^T \mathbf{A}\mathbf{w} + \lambda_2 \mathbf{1}^T \mathbf{w}$. Hence, the surrogate function is convex in $\mathbf{w}$, allowing for simple minimization. The KNIFE algorithm for nonlinear kernels is outlined in Algorithm 1:

Algorithm 1 KNIFE algorithm

1. Initialize $\boldsymbol{\alpha}^{(0)}$ and $\mathbf{w}^{(0)}$, where $0 < w_j^{(0)} < 1 \quad \text{for } j = 1, \ldots, p.$
2. Set $\boldsymbol{\alpha}^{(t)} = \text{argmin}_{\boldsymbol{\alpha}} \left\{ L(\mathbf{Y}, \mathbf{K}_{\mathbf{w}^{(t-1)}}\boldsymbol{\alpha}) + \lambda_1 \boldsymbol{\alpha}^T \mathbf{K}_{\mathbf{w}^{(t-1)}}\boldsymbol{\alpha} \right\}$.
3. Set $\mathbf{w}^{(t)}$ to the solution of:

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & L(\mathbf{Y}, \mathbf{B}\boldsymbol{\alpha}^{(t)} + \mathbf{A}\mathbf{w}) + \lambda_1 (\boldsymbol{\alpha}^{(t)})^T \mathbf{A}\mathbf{w} + \lambda_2 \mathbf{1}^T \mathbf{w} \\ \text{subject to} \quad & 0 \le w_j < 1, \, j = 1, \ldots, p, \end{aligned}$$

   where $\mathbf{B} \in \Re^{n \times n} : \mathbf{B}_{ii'} = k_{\mathbf{w}^{(t-1)}}(i, i') - \bigtriangledown k_{\mathbf{w}^{(t-1)}}(i, i')^T \mathbf{w}^{(t-1)}$, $\mathbf{A} \in \Re^{n \times p} : \mathbf{A}_{ii'} = \sum_{i'=1}^{n} \alpha_{i'} \bigtriangledown k_{\mathbf{w}^{(t-1)}}(i, i')^T$, and $\nabla k_{\mathbf{w}^{(t-1)}}(i, i')$ is the gradient of the (i, i') element of $\mathbf{K}_{\mathbf{w}^{(t-1)}}$ with respect to $\mathbf{w}^{(t-1)}$.
4. Repeat Steps 2–3 until convergence.

The KNIFE algorithm iterates between finding the coefficients of the regression or classification problem and finding the sparse set of feature weights. Kernel linearization has allowed one to circumvent the difficulties associated with direct minimization of the feature-regularized kernel loss function and formulate a computationally efficient algorithm for kernel feature selection.

Here, an interesting attribute of the KNIFE algorithm is noted. By linearizing kernels with respect to the weights, the iterative optimization to estimate coefficients and weights are both problems of the same form but in different spaces. For example, with squared error loss, minimization with respect to the coefficients is a least-square problem in n-dimensional space, whereas minimization with respect to the weights in the linearized kernel is also a least-square problem in p-dimensional feature space. Further properties of the KNIFE problem and algorithm through comparisons with other methods are discussed in the next section.

3.3 Connections With Other Methods

While the KNIFE optimization problem may appear unfamiliar, there are several variations that are the same or similar in form to existing methodology. Consider linear kernels for regression problems with squared error loss, which can be written as

$$\text{minimize} \;\; ||\mathbf{Y} - \mathbf{X}\mathbf{W}\boldsymbol{\beta}||_2^2 + \lambda_1 \boldsymbol{\beta}^T \mathbf{W}\boldsymbol{\beta} + \lambda_2 ||\mathbf{w}||_1 \;\; \text{subject to} \;\; \mathbf{1} > \mathbf{w} \ge 0, \tag{3}$$

$$\text{or} \;\; ||\mathbf{Y} - \mathbf{X}\tilde{\boldsymbol{\beta}}||_2^2 + \lambda_1 \tilde{\boldsymbol{\beta}}^T \mathbf{W}^{-1} \tilde{\boldsymbol{\beta}} + \lambda_2 ||\mathbf{w}||_1 \;\; \text{subject to} \;\; \mathbf{1} > \mathbf{w} \ge 0. \tag{4}$$

These are closely related to several common regression methods. (While in the problem discussed here, $\lambda_1 > 0$ and $\lambda_2 \geq 0$, for comparison purposes, these constraints are relaxed.) First, if one let $\lambda_2 = 0$ and $\mathbf{w} = \mathbf{1}$ in (3), then one obtains ridge regression. If one let $\lambda_1 = 0$, then KNIFE is closely related to the form of the nonnegative garrote (Breiman 1995). The form of both (3) and (4) is very similar to the elastic net that places an L_1 and an L_2 penalty on the coefficients (Zou and Hastie 2005). In KNIFE, however, the L_1 penalty is not on the coefficients, but on the weights that multiply the coefficients. Letting $\lambda_1 = 0$, one obtains a problem very similar in structure and intent to the lasso (Tibshirani 1996). Also, if one let $\lambda_2 = 0$, then one gets a problem that puts weights on the L_2 penalty of the coefficients. This is similar to the adaptive lasso that places weights on the L_1 penalty on the coefficients (Zou 2006).

These similarities between other regression methods and KNIFE hold with other forms of loss functions also. For SVMs, a problem similar to the L_0, L_1, and L_2 SVMs is obtained in the same way that the inner-product squared error loss KNIFE relates to ridge and lasso regression. The same is true of L_0-, L_1-, and L_2-regularized logistic regression.

In addition to these methods, the COSSO, which performs variable selection in smoothing splines, gives (3) and (4) exactly (Lin and Zhang 2007) when the splines are linear. This method minimizes the sums of squares between the response and a function with an L_2 penalty on the projection of the function scaled by the inverse of a nonnegative weight. This proposed form is also the form of (4). Additionally, the COSSO employs an algorithm that first fits a smoothing spline and then fits a nonnegative garrote, noting that these steps can be repeated. This approximate descent algorithm is similar to the KNIFE approach.

For SVMs with the hinge loss, KNIFE with linear kernels is also closely related to multiple kernel learning (Bach, Lanckriet, and Jordan 2004; Lanckriet et al. 2004) . Multiple kernel learning seeks to "learn" the kernel by replacing the typical kernel matrix in SVMs with the weighted sums of a set of linear kernels. Thus, one can form a linear kernel for each feature and combine them in a weighted sum (Xu et al. 2009), creating something similar to the feature-weighted kernels. (Note that the feature-weighted kernels contain the weights squared, instead of linear weights used in multiple kernel learning.) Lanckriet et al. (2004) showed how this problem can be transformed into a convex optimization problem, namely a semidefinite program, while Bach, Lanckriet, and Jordan (2004) and Xu et al. (2009) showed that this is equivalent to a quadratically constrained quadratic program. Hence, if one reparameterizes the feature-weighted linear kernels, KNIFE can also be written in the dual form as a convex problem via the multiple kernel learning framework. We note that as a reviewer pointed out, (4) is also jointly convex in $\tilde{\boldsymbol{\beta}}$ and $\mathbf{W}$.

3.4 Graphically Illustrating Nonlinear Feature Relationships

Since the KNIFE method uses regularization to extract important features, one can extend the KNIFE algorithm to graphically depict nonlinear feature relationships. The KNIFE solution is computed using warm starts over a grid of regularization parameters. Connecting these solutions allows one to visualize the nonlinear relationships among features as they relate to the response. These are loosely called feature paths, noting that as a reviewer pointed out, these are not the result of a path algorithm such as Hastie et al. (2004), but instead a series of connected solutions.

With KNIFE, two penalty parameters, λ_1 and λ_2, are obtained and both of these parameters penalize the feature weights. The first penalty, λ_1, affects the feature weights through the kernel matrix, $K_{\mathbf{w}}$, and also penalizes the coefficients, $\boldsymbol{\alpha}$, while λ_2 places a direct L_1 penalty on the weights. The latter encourages sparsity in the feature weights. Hence, when formulating an algorithm to estimate feature paths, the focus is on λ_2, fixing the value of λ_1. In general, setting $\lambda_1 = 1$, or if the loss function is given as $\frac{1}{n}L(\mathbf{Y}, f(\mathbf{X}))$, then $\lambda_1 = \frac{1}{n}$ performs well in all the simulations and examples, and is thus the default value for the remainder of the article. Also, fixing λ_1 at a small value may especially be of interest in SVMs as this permits a large margin and then allows the weights to both select features and further restrict the margin size.

Setting $\lambda_2 = 0$ gives no direct penalty on the feature weights and thus all features are permitted to be nonzero. The algorithm discussed here computes solutions starting from $\lambda_2 = 0$, where all feature weights are nonzero, to $\lambda_2 = M$, where M is the value at which all weights become zero. The grid of values between can be taken as 100 log-spaced values, an approach used by Friedman, Hastie, and Tibshirani (2010). The feature path algorithm is outlined in Algorithm 2.

Algorithm 2 KNIFE feature path algorithm

1. Fix λ_1, set $\lambda_2 = 0$, and initialize $\boldsymbol{\alpha}^{(0)}$ and $0 < \mathbf{w}^{(0)} < 1$.
2. Fit KNIFE with $\boldsymbol{\alpha}^{(t-1)}$ and $\mathbf{w}^{(t-1)}$ as warm starts.
3. Increase λ_2.
4. Repeat Steps 2–3 until $\mathbf{w}_j^{(k)} = \mathbf{0}$ for all $j = 1, \ldots, p$.

Two attributes of the KNIFE algorithm permit us to estimate feature paths in this manner. First, recall that kernels are linearized with respect to the weights, and in doing so, an algorithm that is *sticky* at zero is created. Thus, as λ_2 is increased once a particular feature's weight is set to zero, it cannot ever become nonzero. This attribute permits one to efficiently use warm starts for the coefficients and weights, speeding computational time considerably. Additionally, it ensures that subsequent solutions are close to the previous solutions, allowing one to connect solutions as to visualize the feature paths. Also, with warm starts and a small increase in λ_2, one can use a single update of the coefficients and weights at each iteration to approximate the feature paths. In addition, the *sticky* property allows one to limit the features under current consideration in the algorithm to the active set, or the current set of features with nonzero weights. Hence, with all of these advantages, the computational time does not dramatically increase from that of the original KNIFE algorithm.

The feature path algorithm discussed here is briefly compared with the well-known coefficient paths of the lasso and LAR (least angle regression) algorithms (Osborne, Presnell, and Turlach 2000; Efron et al. 2004). In these regularization paths, the algorithm begins with no variables in the model and incrementally includes variables that are most correlated with the response. The algorithm discussed here, however, begins with all features in the model and incrementally eliminates the features that are uncorrelated in the kernel

space with the response. Thus, the KNIFE path estimation algorithm can be thought of as a coherent regularization approach to the more heuristic backward elimination for kernels. Also, the lasso regularization paths permit coefficient paths to cross zero and enter and reenter the model. However, KNIFE does not allow this because of the *sticky* property of the feature weights, meaning that once a feature weight is set to zero it cannot move away from zero.

4. RESULTS

In this section, the performance of the KNIFE algorithm and the KNIFE feature path algorithm on both real and simulated data is investigated.

4.1 SIMULATIONS

Two simulation examples are presented demonstrating the performance of KNIFE with kernel regression and kernel SVMs for nonlinear regression and nonlinear classification, respectively. For both simulations, 50 training sets of size 100×10 and test sets of size 1000×10 were generated. Parameters for KNIFE and all comparison methods were selected on a separate validation set prior to training.

4.1.1 Nonlinear Regression. A sinusoidal, nonlinear regression problem with five true features and five noise features is simulated as follows. Let $\mathbf{x} \in \Re^p$ be standard normal with true coefficients, $\boldsymbol{\beta}^{\text{true}} = [6, -4, 3, 2, -2, 0, 0, 0, 0, 0]^T$. Then, the response is given by $y = \sin(\mathbf{x})\boldsymbol{\beta}^{\text{true}} + \epsilon$, where $\epsilon \sim N(0, 1)$. KNIFE with squared error loss and radial and second-order polynomial kernels is compared with linear ridge regression, kernel ridge regression, the filtering method sure independence screening (SIS) (Fan and Lv 2008), RFE (Guyon et al. 2002) (both with kernel ridge regression), and COSSO (Lin and Zhang 2007). For a fair comparison, one parameter was validated for each method. For KNIFE, λ_2 is selected and λ_1 is fixed at $\lambda_1 = 1$, a default used in the remainder of the examples in this article. For kernel ridge methods, the scale parameter, γ, for the radial kernel is set to $\gamma = 1/p$, a commonly used default.

In Table 1, the mean squared error for the training and test sets over the 50 simulations is reported. It is observed that KNIFE with radial kernels outperforms competing methods both in terms of mean squared error and in the correct selection of true features. KNIFE with radial kernels gives better error rates than second-order polynomial kernels. Figure 1 presents example feature paths for both radial and polynomial kernels. It can be observed that while the polynomial kernel gives much smoother feature paths, the radial kernel estimates the true features for a much larger range of the regularization parameter. Moving from large values of the penalty parameter to smaller values, note that the weights for the radial kernel shift when additional features are added to the model. This occurs as the feature weights also behave as automatic scaling factors, a point also noted by Grandvalet and Canu (2002).

4.1.2 Nonlinear Classification. Here, kernel SVMs to assess KNIFE's performance on a nonlinear classification simulation are used based on the skin of the orange example taken from Hastie, Tibshirani, and Friedman (2009). In this example, the first class has four

Table 1. Simulation results for sinusoidal, nonlinear regression. The data have 10 features, five of which are noise features. Mean squared error on training and test sets are given with the standard errors in parentheses. The average percentage of correct features and of noise features selected by the methods is also given. The best performing method is in bold

Method	Training error	Test error	% Features	% Nonfeatures
Ridge	4.8337 (0.1428)	6.2589 (0.0771)	–	–
COSSO	2.2851 (0.7584)	7.0126 (0.6998)	94.0 (1.74)	31.7 (4.32)
Kernel ridge (KR)—radial	0.0874 (0.0035)	6.1239 (0.0750)	–	–
SIS/KR—radial	0.9592 (0.0862)	3.8119 (0.2752)	91.6 (1.63)	8.4 (1.63)
RFE/KR—radial	0.7848 (0.0526)	5.4386 (0.2585)	83.6 (1.70)	36.4 (1.70)
KNIFE/KR—radial	2.1187 (0.0410)	**3.5498** (0.0587)	**98.8** (3.39)	**2.4** (4.67)
KNIFE/KR—polynomial	5.2376 (0.1280)	6.8591 (0.1315)	94.8 (7.42)	0.0 (0.00)

standard normal features, $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$, and $\mathbf{x}_4$, and the second class has the same conditioned on $9 \leq \sum_{j=1}^{4} \mathbf{x}_j^2 \leq 16$. Thus, the model has one class that is spherical, with the second class surrounding the sphere like the skin of the orange. The two classes are not completely separable, however, giving a Bayes' error of 0.0611. KNIFE with SVMs is compared with a standard SVM, SIS and RFE with SVMs, and the adaptive scaling approach of Grandvalet and Canu (2002). All methods use second-order polynomial kernels. Note also that KNIFE was used with the squared error hinge loss approximation to the hinge loss of the SVM (Supplementary Materials).

In Table 2, results for the skin of the orange simulation in terms of test and training misclassification error and the percentage of true and noise features selected are presented. Here, KNIFE outperforms competing methods in terms of test error and selection of true features. Also, note that the adaptive scaling method performs similarly to KNIFE in terms of error rates, but is less selective of features as a large percentage of noise features are selected. Recall that this method employs a similar objective to KNIFE, but with an L_2 penalty instead of an L_1 penalty. Results for this simulation clearly illustrate the advantages of using an $L_!$ penalty in this context for feature selection.

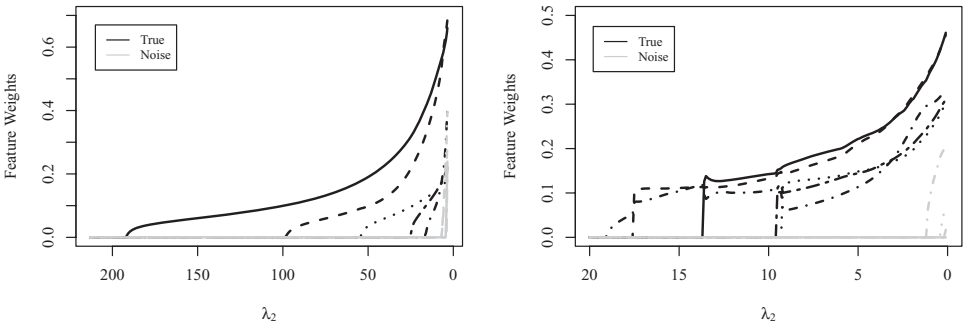

Figure 1. Example feature paths for KNIFE with second-order polynomial kernels (left) and radial kernels (right) with a squared error loss function. Data of dimension 100×10 were simulated from the sinusoidal simulation with five true features and five noise features. KNIFE for both kernel types gives nonzero weights to the five true features for much of the feature paths.

Table 2. Average misclassification errors for the skin of the orange simulation with four true features and six noise features. All methods use SVMs with second-order polynomials. Additionally, the average percentage of correct features and of noise features selected by the methods is given. Standard errors are in parentheses with the best performing method in bold

Method	Training error	Test error	% Features	% Nonfeatures
SVM	0 (0)	0.1919 (0.0042)	–	–
SIS/SVM	0 (0)	0.2147 (0.0082)	92.5 (1.63)	88.3 (1.09)
RFE/SVM	0.0872 (0.0090)	0.2188 (0.0097)	78.0 (2.44)	31.3 (1.65)
Adaptive scaling/SVM	0.0374 (0.0028)	0.1093 (0.0040)	100.0 (0)	88.7 (1.81)
KNIFE/SVM	0.0366 (0.0025)	**0.0986** (0.0043)	**100.0** (0)	**16.7** (1.98)

To explore the behavior of KNIFE when the number of noise features in the model increases, the study begins with the skin of the orange model with four true features and adds noise features. Results compared with the standard SVM are shown in Figure 2. It is observed that the misclassification errors for KNIFE remain stable as the number of noise features in the model increases, whereas those of the standard SVM increase dramatically.

Also, here, an interesting characteristic of KNIFE for SVMs is noted. The SVM is sparse in the observation space, meaning that only a subset of the observations are chosen as "support vectors." With KNIFE/SVM, one obtains sparsity in the feature space also, leaving one with an important submatrix of observations and features that can limit computational storage for prediction purposes.

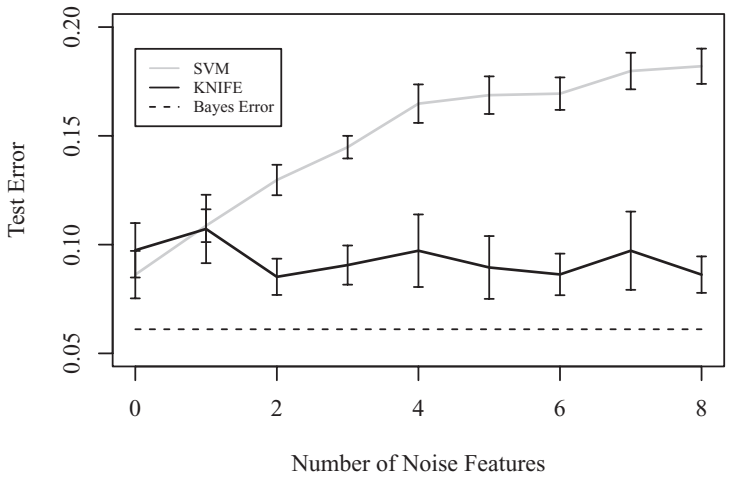

Figure 2. Mean test misclassification error with standard errors when noise features are added to the skin of the orange simulation. SVMs and KNIFE for SVMs with second-order polynomial kernels are trained on data with 100 observations and tested on data with 1000 observations. The Bayes' error of this simulation is 0.0611. By selecting the true features, the error rates for KNIFE remain constant as the number of noise features in the model increases.

Table 3. Average misclassification rates with standard errors for 10 randomly created test sets trained on a set of equal size for the colon cancer microarray data. All methods use a linear SVM. Best performing methods are in bold

No. of genes	SIS/SVM	RFE/SVM	L_1-norm/SVM	KNIFE/SVM
500	**0.0452** (0.016)	0.1290 (0.016)	0.1097 (0.022)	0.0484 (0.019)
250	0.0581 (0.022)	0.1452 (0.020)	0.1032 (0.023)	**0.0516** (0.018)
100	**0.0548** (0.019)	0.1677 (0.021)	0.1097 (0.022)	0.0710 (0.0054)
50	**0.0677** (0.0047)	0.1774 (0.0065)	0.1097 (0.0074)	0.0806 (0.018)
25	0.0968 (0.017)	0.1742 (0.013)	0.1097 (0.023)	**0.0871** (0.017)
15	0.1161 (0.019)	0.1484 (0.016)	0.1161 (0.022)	**0.1065** (0.019)
10	**0.1194** (0.027)	0.1581 (0.019)	0.1355 (0.021)	**0.1194** (0.019)

4.2 EXAMPLES

In this section, KNIFE is applied to three feature selection applications: gene selection in microarrays, variable selection in vowel recognition data, and variable selection in predicting ozone levels.

4.2.1 High-Dimensional Data: Microarrays. With 30,000 human genes, doctors often need a small subset of genes to test that are predictive of a disease. To demonstrate the performance of KNIFE for gene selection, the method discussed here is applied to microarray data on colon cancer, publicly available at *http://genomics-pubs.princeton.edu/oncology/* (Alon et al. 1999). The dataset consists of 62 samples, 22 of which are normal and 40 of which are from colon cancer tissues. The genes are already prefiltered, consisting of the 2000 genes with the highest variance across samples.

For this analysis, a linear SVM is used for classification, comparing KNIFE with the L_1-norm SVM (Zhu et al. 2003) and gene filtering using SIS and RFE. Eight subsets of previously fixed numbers of genes on the three methods are evaluated. For the gene selection with RFE, begin by eliminating 50 genes, 10 genes, and then one gene at each step as outlined by Guyon et al. (2002). To determine predictive ability, the samples are randomly split into training and test sets of equal sizes. This is repeated 10 times and misclassification rates are averaged with the results given in Table 3.

The results indicate that KNIFE outperforms RFE filtering and the L_1-norm SVM for all subsets of genes, and performs similarly to the SIS filtering method with a small advantage when selecting small subsets of genes. While the subsets of genes determined by SIS may perform similarly in terms of classification, often researchers are interested in a subset of genes that are members of different pathways and are hence less correlated. For SIS filtering, the median pairwise correlation of genes selected is 0.665 and 0.714 for 25 and 10 genes, respectively, compared with 0.590 and 0.590 for KNIFE. The latter is close to the median correlation observed among genes in the dataset as a whole at 0.591. KNIFE, then, selects genes that classify the sample well and are less correlated.

4.2.2 Nonlinear Classification: Vowel Recognition. As an example of KNIFE for nonlinear classification, the method is applied to the vowel recognition dataset available at *http://www-stat.stanford.edu/tibs/ElemStatLearn/* (Hastie, Tibshirani, and Friedman 2009). This dataset consists of 11 classes of vowels broken down into 10 features in which 15

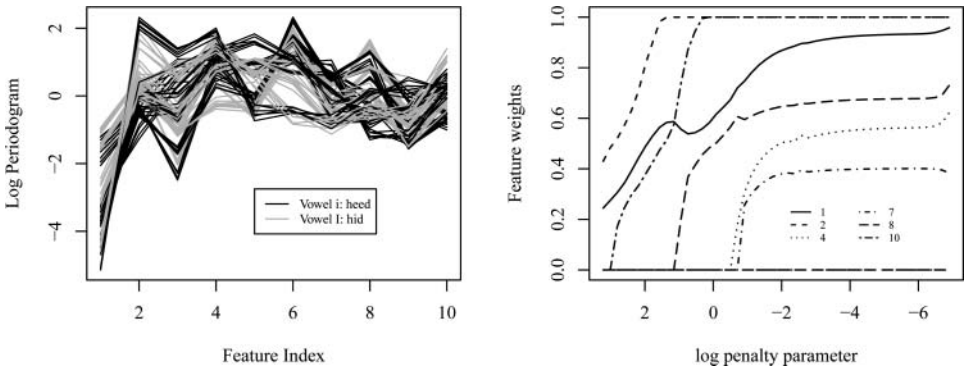

Figure 3. Vowel recognition data for vowels "i" and "I" (left) with KNIFE-estimated feature paths for classifying between the two vowels using an SVM with radial kernels (right). Five-fold cross-validation on the training set chose to include the six features is shown. KNIFE gives a test misclassification error of 8.3%, while a radial kernel SVM has 19.1% test error.

individuals were recorded speaking six times. A radial kernel SVM and KNIFE/SVM are applied to classify between vowel sounds "i" and "I" based on 10 features (related to the log periodogram), as shown on the left in Figure 3. Five-fold cross-validation is used to determine the margin size for the SVM and the λ_2 value for KNIFE. Both methods were trained on a dataset with 48 instances and tested with 42 instances of each vowel. KNIFE gives a test misclassification error of 8.3%, while the SVM gives an error of 19.1%. It can be observed from the example feature paths in Figure 3 that KNIFE selects six features that are indicative of the two vowel types and gives a visual representation of the highly nonlinear relationship between the features and the vowel sounds.

4.2.3 Nonlinear Regression: Ozone Data. The ozone dataset (Breiman and Friedman 1985), available in the COSSO R package (Zhang and Lin 2010), has been commonly used to compare methods for nonlinear regression. The dataset consists of 330 daily readings of ozone levels and eight predictors from Los Angeles in 1976. KNIFE using L_2 loss with radial kernels is compared with the COSSO nonlinear regression method (Lin and Zhang 2007) by randomly splitting the data into 10 test and training sets of equal sizes. Five-fold cross-validation is used to selected penalty parameters for both the methods. KNIFE performs well in terms of mean squared error yielding a training and a test error of 14.27 (0.40) and 16.96 (0.39), respectively (standard errors are in parenthesis), while COSSO yields errors of 15.28 (0.36) and 18.57 (0.36), respectively. In Figure 4, KNIFE also reveals important nonlinear relationships between the predictors and the ozone levels. Temperature is the most important variable in predicting ozone levels followed by inversion height (invHt) and humidity (hum). These variables are predictive of ozone in largely independent manners as observed by the steady slopes of the feature estimates. When the variables' barometric pressure and inversion temperature enter the model, the feature weights for temperature, inversion height, and humidity shift, indicating that there are colinearities, in the nonlinear kernel space, between these two sets of variables. Five-fold cross-validation for KNIFE on the full dataset gives nonzero weight to all but two predictors, that is, wind speed and millibar pressure.

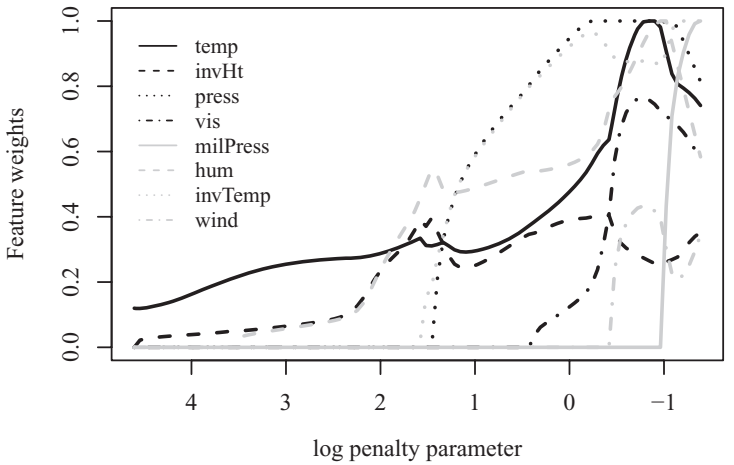

Figure 4. Nonlinear feature relationships estimated by KNIFE with radial kernels to predict ozone levels based upon eight predictors. Five-fold cross-validation selects six variables, all except wind speed and millibar pressure. On 10 random splits of this data into test and training sets, KNIFE yields a training and a test mean squared error of 14.17 (0.40) and 16.96 (0.39), respectively, compared with COSSO at 15.28 (0.36) and 18.57 (0.36), respectively.

5. DISCUSSION

In this article, an algorithm to minimize a feature-regularized loss function has been presented, thus achieving automatic feature selection in nonlinear kernel spaces.

Computationally, the KNIFE algorithm compares favorably with existing kernel feature selection methods with the exception of simple feature filtering methods. The KNIFE algorithm iterates between an optimization problem in n-dimensional feature space and then in p-dimensional feature space. This is similar to the computational burden of the adaptive scaling procedure of Grandvalet and Canu (2002), but is more efficient than the DC programming approach of Argyriou et al. (2006). In addition, one can fun the KNIFE algorithm for a limited number of iterations to further restrict computational costs.

While specific examples with kernel ridge regression and kernel SVMs have been presented, the KNIFE method is applicable to a variety of kernel problems that can be written with a loss function. These include kernel logistic regression, which has a binomial deviance loss, kernel principal component analysis, kernel discriminant analysis, and kernel canonical correlation, which all can be written with a Frobenius norm loss. Thus, the KNIFE method has many potential future uses for feature selection in a variety of kernel methods. Also, the KNIFE problem for general kernel regression or classification problems has been presented. Problem-specific versions of the KNIFE algorithm will be considered in future work.

In conclusion, a coherent criterion for selecting features in kernel problems via the feature-regularized kernel loss function has been presented. The methods discussed in this article are applicable to general kernel problems, are computationally feasible in high-dimensional settings, and give visual representations of the relationships between

variables. Thus, KNIFE provides a valuable tool for feature selection with nonlinear kernel problems.

SUPPLEMENTARY MATERIALS

Appendix: Technical details on the convergence analysis of KNIFE and its theoretical properties.

Computer code: MATLAB functions and scripts implementing the KNIFE algorithm.

ACKNOWLEDGMENTS

The author is grateful to Robert Tibshirani for the helpful suggestions and advice in developing and testing this method. The author thanks Stephen Boyd for suggesting kernel linearization, Rahul Mazumder and Holger Hoefling for discussions on algorithm convergence, and Trevor Hastie for the helpful suggestions. The author also thanks three anonymous reviewers, the editor, and associate editor for suggestions that led to several improvements in this article.

[Received July 2010. Revised December 2011.]

REFERENCES

Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D., and Levine, A. J. (1999), "Broad Patterns of Gene Expression Revealed by Clustering Analysis of Tumor and Normal Colon Tissues Probed by Oligonucleotide Arrays," *Proceedings of the National Academy of Sciences*, 96, 6745–6750. [295]

Argyriou, A., Hauser, R., Micchelli, C. A., and Pontil, M. (2006), "A DC-Programming Algorithm for Kernel Selection," in *ICML '06: Proceedings of the 23rd International Conference on Machine Learning*, pp. 41–48. [285,287,297]

Bach, F. R., Lanckriet, G. R. G., and Jordan, M. I. (2004), "Multiple Kernel Learning, Conic Duality, and the SMO Algorithm," in *ICML '04: Proceedings of the Twenty-First International Conference on Machine Learning*, pp. 41–48. [285,290]

Bertin, K., and Lecué, G. (2008), "Selection of Variables and Dimension Reduction in High-Dimensional Non-Parametric Regression," *Electronic Journal of Statistics*, 2, 1224–1241. [285,286]

Breiman, L. (1995), "Better Subset Regression Using the Nonnegative Garrote," *Technometrics*, 37, 373–384. [290]

Breiman, L., and Friedman, J. (1985), "Estimating Optimal Transformations for Multiple Regression and Correlation," *Journal of the American Statistical Association*, 80, 580–598. [296]

Candes, E., and Plan, Y. (2009), "Near-Ideal Model Selection by ℓ_1 Minimization," *The Annals of Statistics*, 37, 2145–2177. [284]

Cao, B., Shen, D., Sun, J., Yang, Q., and Chen, Z. (2007), "Feature Selection in a Kernel Space," in *ICML '07: Proceedings of the 24th International Conference on Machine Learning*, pp. 121–128. [285]

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004), "Least Angle Regression," *The Annals of Statistics*, 32, 407–499. [291]

Fan, J., and Lv, J. (2008), "Sure Independence Screening for Ultrahigh Dimensional Feature Space," *Journal of the Royal Statistical Society*, Series B, 70, 849–911. [292]

Friedman, J., Hastie, T., and Tibshirani, R. (2010), "Regularization Paths for Generalized Linear Models via Coordinate Descent," *Journal of Statistical Software*, 33, 1–22. [291]

Grandvalet, Y., and Canu, S. (2002), "Adaptive Scaling for Feature Selection in SVMs," in *Advances in Neural Information Processing Systems 15*, pp. 553–560. [285,286,292,297]

Guyon, I., Bitter, H. M., and Ahmed, Z. (2003), "Multivariate Nonlinear Feature Selection With Kernel Multiplicative Updates and Gram-Schmidt Relief," in *Proceedings of the BISC FLINT-CIBI 2003 Workshop*, Berkeley, CA. [285]

Guyon, I., Weston, J., Barnhill, S., and Vapnik, V. (2002), "Gene Selection for Cancer Classification Using Support Vector Machines," *Machine Learning*, 46, 389–422. [285,292,295]

Hastie, T., Rosset, S., Tibshirani, R., and Zhu, J. (2004), "The Entire Regularization Path for the Support Vector Machine," *The Journal of Machine Learning Research*, 5, 1391–1415. [290]

Hastie, T., Tibshirani, R., and Friedman, J. (2009), *Elements of Statistical Learning* (2nd ed.), New York: Springer. [292,295]

Lafferty, J., and Wasserman, L. (2008), "Rodeo: Sparse, Greedy Nonparametric Regression," *The Annals of Statistics*, 36, 28–63. [285,286]

Lanckriet, G., Cristianini, N., Bartlett, P., and Ghaoui, L. E. (2004), "Learning the Kernel Matrix With Semidefinite Programming," *Journal of Machine Learning Research*, 5, 27–72. [285,290]

Li, F., Yang, Y., and Xing, E. (2006), "From Lasso Regression to Feature Vector Machine," in *Advances in Neural Information Processing Systems 18*, pp. 779–786. [285]

Lin, Y., and Zhang, H. H. (2007), "Component Selection and Smoothing in Multivariate Nonparametric Regression," *The Annals of Statistics*, 34, 2272–2297. [285,286,290,292,296]

Navot, A., and Tishby, N. (2004), "Margin Based Feature Selection—Theory and Algorithms," in *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 43–50. [285,286]

Neumann, J., Schnörr, C., and Steidl, G. (2005), "Combined SVM-Based Feature Selection and Classification," *Machine Learning*, 61, 129–150. [285]

Osborne, M., Presnell, B., and Turlach, B. (2000), "A New Approach to Variable Selection in Least Squares Problems," *IMA Journal of Numerical Analysis*, 20, 389–403. [291]

Tibshirani, R. (1996), "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society*, Series B, 58, 267–288. [290]

Wahba, G., Lin, Y., and Zhang, H. (1999), "Generalized Approximate Cross Validation for Support Vector Machines, or, Another Way to Look at Margin-Like Quantities," Technical Report, University of Wisconsin. [286]

Wang, L. (2008), "Feature Selection With Kernel Class Separability," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30, 1534–1546. [285,286]

Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., and Vapnik, V. (2000), "Feature Selection for SVMs," in *Advances in Neural Information Processing Systems 13*, pp. 668–674. [285,286]

Xu, Z., Jin, R., Ye, J., Lyu, M. R., and King, I. (2009), "Non-Monotonic Feature Selection," in *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 1145–1152. [285,290]

Zhang, H. H., and Lin, C.-Y. (2010), *COSSO* (R package version 1.0-1), Raleigh, NC: North Carolina State University. [296]

Zhu, J., Rosset, S., Hastie, T., and Tibshirani, R. (2003), "1-Norm Support Vector Machines," in *Advances in Neural Information Processing Systems*, pp. 1–16. [285,295]

Zou, H. (2006), "The Adaptive Lasso and Its Oracle Properties," *Journal of the American Statistical Association*, 101, 1418–1429. [290]

Zou, H., and Hastie, T. (2005), "Regularization and Variable Selection via the Elastic Net," *Journal of the Royal Statistical Society*, Series B, 67, 301–320. [290]