mathematics

MDPI

Article

Simultaneous Feature Selection and Classification for Data-Adaptive Kernel-Penalized SVM

Xin Liu [1], Bangxin Zhao [2] and Wenqing He [2,*]

[1] School of Statistics and Management, Shanghai University of Finance and Economics, Shanghai 200433, China; liu.xin@mail.shufe.edu.cn

[2] Department of Statistical and Actuarial Sciences, University of Western Ontario, London, ON N6A 5B7, Canada; bzhao49@uwo.ca

* Correspondence: whe@stats.uwo.ca

Received: 20 September 2020; Accepted: 15 October 2020; Published: 20 October 2020

Abstract: Simultaneous feature selection and classification have been explored in the literature to extend the support vector machine (SVM) techniques by adding penalty terms to the loss function directly. However, it is the kernel function that controls the performance of the SVM, and an imbalance in the data will deteriorate the performance of an SVM. In this paper, we examine a new method of simultaneous feature selection and binary classification. Instead of incorporating the standard loss function of the SVM, a penalty is added to the data-adaptive kernel function directly to control the performance of the SVM, by firstly conformally transforming the kernel functions of the SVM, and then re-conducting an SVM classifier based on the sparse features selected. Both convex and non-convex penalties, such as least absolute shrinkage and selection (LASSO), moothly clipped absolute deviation (SCAD) and minimax concave penalty (MCP) are explored, and the oracle property of the estimator is established accordingly. An iterative optimization procedure is applied as there is no analytic form of the estimated coefficients available. Numerical comparisons show that the proposed method outperforms the competitors considered when data are imbalanced, and it performs similarly to the competitors when data are balanced. The method can be easily applied in medical images from different platforms.

Keywords: classification; data-adaptive kernel; feature selection; penalty; predictive model; simultaneous classification; support vector machine

1. Introduction

As one of the most critical tasks in data mining with high dimensions, the performance of a classification model relies on selecting the most appropriately relevant features while removing irrelevant ones. This offers advantages, including a lower risk of overfitting, less model complexity (and hence the improvement of the generalization capability) and less computational cost [1]. In scientific research, classification models have served as useful tools of artificial intelligence in various areas such as financial credit risk assessment [2], signal processing and pattern recognition [3]. In these tasks, the fundamental goal is the accuracy of prediction in various situations.

The support vector machine (SVM) offers a method of classification that has adequate generalizing ability, fewer local minima with limited dependence on only a few parameters [4], and has achieved success in applications as a powerful classifier of high accuracy with flexibility; see, e.g., [5]. However, the method described in the standard formulation settings cannot decide the importance from different features [6], while its performance may be severely deteriorated when redundant variables are used in determining the decision rule, even those as poor as random guessing due to the accumulation of random noise, especially in a high dimensional space [7,8]. Consequently, the development of

several approaches for selecting features with SVMs has been motivated, e.g., in [8–11], which provide various ways of feature ranking or selection. One of the main directions, known as the filter method, filters out features with poor information based on statistical properties of features, usually done before applying any classification models, such as in [12]. Another framework, called the wrapper method, scores the whole set of features based on their predictive powers, and then selects a subset of variables with the highest scores. The wrapper method shows more accuracy than the filter method. The most popular wrapper method for SVMs may be the Recursive Feature Elimination SVM, proposed in [13], which attempted to find the best subset of r features among m variables $(r < m)$, on the basis of a sequential backward selection technique. However, they all have the drawback of not taking into consideration the combination of features that optimize the performance of the classifier simultaneously.

Correspondingly, the embedded methods are created so that the selection of features can be performed during the model construction. A typical way of achieving this goal is to add some extra term that penalizes the cardinality of the selected subset of features to the standard cost function, named as the hinge loss, of the support vector machines, generally with an appropriate sparsity penalty proposed by [14]. This framework is a unified method that achieves variable selection and prediction simultaneously. The standard SVM is well known to fit in the regularization framework of loss plus penalty with the hinge loss and L_2 norm penalty and, to generalize its usefulness, quite a few attempts have been employed to select features for the SVM by using other forms of penalty. For example, L_1 norm penalty is applied in [15–17]; [18,19] proposed the elastic net penalty for the SVM, and the adaptive LASSO penalty form was proposed to penalize the SVM; [20] suggested a F_∞ norm SVM so that groups of predictors could be selected simultaneously. In recent research, [21] studied the smoothly clipped absolute deviation (SCAD) proposed by [14] and proved the oracle property of the SVM with a fixed number of predictors penalized by SCAD.

The aforementioned penalized feature selection methods for SVMs are all based on the predictors in the original input space. However, there are possibilities that those features which have been penalized and eliminated in the input space with the above methods might be useful in the projected feature space generated by the kernel function in solving the SVM, and hence the classifier will lose some useful information accordingly. Actually, when the SVM projects the original input space into a higher dimensional feature space, the performance of an SVM will depend directly on the so-called kernel function, as is pointed out in, for example, [22–24]; thus, a natural idea is to penalize the kernel function directly, so that the features that are useful in the feature space can be selected and the classification can be achieved simultaneously.

Another issue is the imbalance in data that cannot be simply ignored in real classification applications. When the sizes of different classes are incomparable, the performance of the standard SVM or other popular classifiers tend to be unstable (see details in, e.g., [22–24]). To deal with the problem, SVMs with data-adaptive kernels can be applied. One specific idea is to adopt a two-stage approach of constructing data-adaptive kernels is proposed. The method locally adapts the kernel function to the data locations based on the skewness of the class boundary, and hence enlarges the magnification effect directly on the Riemannian manifold in the feature space. Even when the data are extremely imbalanced, the performance of the SVM constructed accordingly is satisfactory.

In this paper, we propose a new method of simultaneous feature selection and classification by penalizing data-adaptive kernels in SVMs. Instead of penalizing the standard cost function of SVMs, the penalty will be directly added to the data-adaptive kernel function that controls the performance of an SVM, by first transforming the kernel functions of the SVM and then re-conducting the SVM formulation optimization, then, finally, getting the classification result with sparse features selected. Different penalty terms such as SCAD, minimax concave penalty (MCP) and L_1 norm penalties will be compared. The oracle property of the estimated classifier is proposed. An iterative optimization process will be applied, as no analytic form of the estimated coefficients can be obtained. Numerical comparisons show that our proposed classifier outperforms with the imbalanced data and performs as

well as others when the data are balanced. Our contribution in this paper is mainly two-fold. Firstly, the proposed method can select relevant predictors in the feature space, with properties of selection consistency established theoretically. None of the papers in the literature have touched or studied the theoretical property of the selection procedure under the framework of an SVM. We also employ non-convex penalty functions including SCAD and MCP, which are much more difficult to deal with when programming due to a non-convex objective function. On the other hand, we take the data imbalance issue into consideration during feature selection with a data-adaptive procedure, and this appears not easy during feature selection procedure, especially when the input space is divergingly large. The imbalance issue that may severely deteriorate the performance of a classifier has not been accommodated in the previous literature during feature selection procedures.

The methodology is partially motivated by an ongoing prostate caner study from London, ON, Canada. The goal is to construct a classifier to predict the cancerous areas with imaging intensity measures that come from different platforms such as MRI and CT. Several issues need to be considered. One is that redundant measures might deteriorate the performance of the classifier, so that a feature selection technique is necessary. Another problem is the imbalance in the data. The cancerous proportion in the prostate only takes 8% on average, indicating an extreme imbalance. Hence, how to perform accurate classification accommodating these issues needs to be addressed.

The rest of the paper organizes as follows. In Section 2, the framework of SVMs and the penalized SVM is introduced. In Section 3, our model is constructed. Not only is the oracle property proposed, but an algorithm to achieve the goal is also introduced for implementation purposes as well. Section 4 shows the experiment results of comparing numerical performances with different models under different scenarios, and the model is applied on a real data set as well. Remarks conclude the paper and technical proofs are given in Appendix A.

2. Notation and Framework

Consider a binary classification problem. Given a random sample $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where $\mathbf{x}_i$ is a vector of features in the input space $I = \mathbb{R}^p$, y_i represents the class index which takes values $+1$ or -1, and p, the dimension of the input space, indicates the number of features available. The goal is to determine a rule so that future observations with only the features available can be labeled into the corresponding class. The Support Vector Machine (SVM) is a technique to obtain the rule. The SVM finds a linear boundary to separate the two classes by maximizing the smallest distance from the observations of each class to the boundary if the samples are linearly separable. When the samples are not linearly separable, the method finds a nonlinear boundary by mapping the input data $\mathbf{x}$ into a high-dimensional feature space $F = \mathbb{R}^l$ using a nonlinear mapping function $\mathbf{s} : \mathbb{R}^p \to \mathbb{R}^l$, and searching a linear discriminant function or a hyperplane

$$\boldsymbol{\beta}^T \mathbf{s}(\mathbf{x}) + b = 0 \tag{1}$$

in the feature space F, where $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_l)$ is an $l-$dimensional vector of parameters, $\mathbf{s}(\mathbf{x}) = (\mathrm{s}_1(\mathbf{x}), \ldots, \mathrm{s}_l(\mathbf{x}))^T$ is the $l-$dimensional column vector, and b is a scalar bias term. Hence, an individual point with observation $\mathbf{x}$ can be classified by the sign of $D(\mathbf{x}) = \boldsymbol{\beta}^T \mathbf{s}(\mathbf{x}) + b$ as long as the parameters $\boldsymbol{\beta}$ and b are determined, and the boundary $D(\mathbf{x}) = 0$ is nonlinear in the input space. Theoretically, the solution to the SVM can be obtained by maximizing the aggregated margin between the separating boundaries [25]. In the mean time, the features that are used to construct the rule should be limited or even sparse so that the rule is easy to implement in practice.

Mathematically, the SVM boundary is the solution of minimizing

$$Q(\boldsymbol{\beta}, b, \xi) \;=\; \frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\sum_{i=1}^{n} \xi_i \tag{2}$$

with respect to $\boldsymbol{\beta}$, and b, subject to the constraints

$$y_i \left(\boldsymbol{\beta}^T \mathbf{s}(\mathbf{x}_i) + b \right) \geq 1 - \xi_i \qquad \text{for } i = 1, \ldots, n,$$

where C is the so-called soft margin parameter that determines the trade-off between the optimal combinatorial choice of the margin and the classification error, and $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_n)^T$ are non-negative slack variables. Equivalently, this optimization problem can be represented in the Lagrangian dual function with the form

$$\underset{\boldsymbol{\alpha}}{\text{Max}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j < \mathbf{s}(\mathbf{x}_i), \mathbf{s}(\mathbf{x}_j) > .$$

subject to the constraints

$$\begin{aligned} \sum_{i=1}^{n} \alpha_i y_i &= 0, \\ 0 \leq \alpha_i &\leq C \end{aligned}$$

for $i = 1, 2, \ldots, n$, where α_i's are the dual variables, and $< \cdot , \cdot >$ is the inner product operator. Generally, a scalar function $K(\cdot, \cdot)$, which is called a kernel function, is adopted to replace the inner product of the two vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ in the dual function

$$\underset{\boldsymbol{\alpha}}{\text{Max}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j). \tag{3}$$

If we denote SV as the set $\{j \mid \alpha_j > 0 \text{ for } j = 1, 2, \ldots, n\}$ with all the observations, and $\mathbf{x}_i, i \in SV$ as the support vectors, correspondingly, the kernel form of the SVM boundary can be written as

$$\sum_{i \in SV} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b = 0, \tag{4}$$

and, consequently, the estimated bias term b_j obtained by using the jth support vector $\mathbf{x}_j$ is defined as

$$b_j = y_j - \sum_{i \in SV} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_j).$$

The bias term b_j is proved to be identical for all j in the set SV [7]. Thus, in practice, with the estimated coefficients of $\boldsymbol{\alpha}_i$, we can take the average of all the estimated b_js with all support vectors as the estimate of b.

Although the kernel form of the SVM is developed through the projection of input space to higher dimensional space, in practice, we may specify the kernel function instead of finding the projection mapping. A number of commonly used kernel functions are available, for example, the radial kernel function

$$K(\mathbf{x}, \mathbf{z}) = h(-\|\mathbf{x} - \mathbf{z}\|^2), \tag{5}$$

where $h(\cdot)$ is a probability density function. When $h(\cdot)$ comes from a Gaussian distribution with variance σ^2, the kernel function is then

$$K(\mathbf{x}, \mathbf{z}) = \exp(-\|\mathbf{x} - \mathbf{z}\|^2 / 2\sigma^2). \tag{6}$$

2.1. Geometric Interpretation of SVM Kernels

Geometrically speaking, when the input space I is the Euclidean space, the Riemannian metric is induced in the feature space F. Let $\mathbf{f}$ be the mapped result of $\mathbf{x} \in R^p$ in F, i.e., $\mathbf{f} = s(\mathbf{x}) \in R^l$, then a small change in $\mathbf{x}$ in the input space, $\mathrm{d}\mathbf{x}$, will be mapped into the vector $\mathrm{d}\mathbf{f}$ in the feature space so that

$$\mathrm{d}\mathbf{f} \;=\; \nabla \mathbf{s} \cdot \mathrm{d}\mathbf{x} = \sum_j \frac{\partial}{\partial x_j} \mathbf{s}(\mathbf{x})\, \mathrm{d}x_j,$$

where

$$\nabla \mathbf{s} \;=\; \left(\frac{\partial\, (s(\mathbf{x})}{\partial\, \mathbf{x}} \right) = \begin{pmatrix} \frac{\partial\, s_1(\mathbf{x})}{\partial\, x_1} & \cdots & \frac{\partial\, s_1(\mathbf{x})}{\partial\, x_p} \\ \vdots & \vdots & \vdots \\ \frac{\partial\, s_l(\mathbf{x})}{\partial\, x_1} & \cdots & \frac{\partial\, s_l(\mathbf{x})}{\partial\, x_p} \end{pmatrix}. \tag{7}$$

Thus, the squared length of $\mathrm{d}\mathbf{f}$ can be written in the quadratic form as

$$\|\mathrm{d}\mathbf{f}\|^2 \;=\; (\mathrm{d}\mathbf{f})^T \cdot \mathrm{d}\mathbf{f} = \Big(\sum_i \frac{\partial}{\partial x_i} \mathbf{s}(\mathbf{x})\, \mathrm{d}x_i \Big)^T \cdot \Big(\sum_j \frac{\partial}{\partial x_j} \mathbf{s}(\mathbf{x})\, \mathrm{d}x_j \Big) = \sum_{ij} s_{ij}(\mathbf{x}) \mathrm{d}x_i \mathrm{d}x_j,$$

where

$$\mathrm{s}_{ij}(\mathbf{x}) = \left(\frac{\partial}{\partial x_i} \mathbf{s}(\mathbf{x}) \right)^T \cdot \left(\frac{\partial}{\partial x_j} \mathbf{s}(\mathbf{x}) \right) = \left(\frac{\partial \mathrm{s}_1(\mathbf{x})}{\partial x_i}, \ldots, \frac{\partial \mathrm{s}_l(\mathbf{x})}{\partial x_i} \right) \cdot \left(\frac{\partial \mathrm{s}_1(\mathbf{x})}{\partial x_j}, \ldots, \frac{\partial \mathrm{s}_l(\mathbf{x})}{\partial x_j} \right)^T. \tag{8}$$

Consequently, the $l \times l$ matrix $S(\mathbf{x}) = [s_{ij}(\mathbf{x})]$ is defined on the Riemannian metric, which can be derived from the kernel K, and $S(\mathbf{x})$ is positive definite [26]. More straightforwardly, the following lemma demonstrates the connection between a kernel function K and a mapping $\mathbf{s}$:

Lemma 1 ([26]). *Suppose $K(\boldsymbol{x}, \boldsymbol{z})$ is a reproducing kernel function, and **s(x)** is the corresponding mapping in the support vector machine. Then, (9) holds that*

$$s_{ij}(\boldsymbol{x}) \;=\; \frac{\partial}{\partial x_i} \frac{\partial}{\partial z_j} K(\boldsymbol{x}, \boldsymbol{z})|_{z=x}. \tag{9}$$

To increase the separability between two categories, the spatial resolution around the boundary surface in F needs to be enlarged. This motivates us to increase the factor $\sqrt{v(\mathbf{x})}$ around the boundary of $D(\mathbf{x}) = 0$. Therefore, the mapping $\mathbf{s}$ or, equivalently, the related kernel K, is to be examined so that $\mathrm{s}_{ij}(\mathbf{x})$ can be enlarged around the boundary. This knowledge is especially useful when dealing with imbalanced data, since it has been known that an imbalance in the data can severely affect the performance of an SVM, where a data-adaptive kernel function is constructed to solve the problem based on the assumed form of the original kernel function.

2.2. Penalized SVM

When there are a large number of features available, not all the features will contribute to the construction of the classifier. Redundant features and extra noise in the available input features may deteriorate the accuracy of the classifier while leading to the complexity of the classifier if they are all included in the model. The number of features may be controlled in the SVM framework. Under the standard prediction risk framework of loss plus penalty form, the potential misclassification cost can be specified by a universal weight c for each of the sample points from the two classes, namely, $Q_i = c$ if $y_i = 1$ and $Q_i = 1 - c$ if $y_i = -1$ for some $0 < c < 1$, and the classification boundary can be estimated by a linear weighted SVM [7,27] by solving

$$\min_{\boldsymbol{\beta},\, b} Loss(\boldsymbol{\beta}) \quad = \quad \min_{\boldsymbol{\beta},\, b} n^{-1} \sum_{i=1}^{n} Q_i \big(1 - y_i(\mathbf{x}_i^T \boldsymbol{\beta} + b)\big)_+ \; + \; \lambda \boldsymbol{\beta}^T \boldsymbol{\beta},$$

where $(1-t)_+ = \max\{1-t, 0\}$ denotes the hinge loss, $\boldsymbol{\beta}$ are the coefficients of the features, b is the intercept and λ is a positive regularization parameter. When the weight $c = 0.5$, the linear weighted SVM goes back to the standard SVM [27]. When the hinge loss is considered as $E[Q(1-(y\mathbf{X}^T\boldsymbol{\beta}+b))_+]$, an analytic form of the estimators of $\boldsymbol{\beta}$ is given by

$$\hat{\boldsymbol{\beta}}_{true} = \arg\min_{b,\boldsymbol{\beta}} \quad n^{-1} \sum_{i=1}^{n} Q_i \big(1 - y_i(\mathbf{X}_i^T \boldsymbol{\beta} + b)\big)_+ \tag{10}$$

Furthermore, in terms of selecting variables from the input space, suppose the true model has sparse features or, equivalently, $\boldsymbol{\beta}^T = (\boldsymbol{\beta}_{true}^T, \mathbf{0}^T)$, where $\boldsymbol{\beta}_{true}^T = (\beta_1, \beta_2, \ldots, \beta_k)$. Denote $\mathbf{x}_i^T = (\mathbf{z}_i^T, \mathbf{u}_i^T)$, where the $k \times 1$ vector $\mathbf{z}$ is the feature vector corresponding to the non-zero coefficients and the $(p-k) \times 1$ covariate vector $\mathbf{u}$ corresponds to the redundant information. To select the vector $\mathbf{z}$, [8] proposed a general form of penalty terms to be added directly to the loss function as

$$Loss(\boldsymbol{\beta}) = n^{-1} \sum_{i=1}^{n} Q_i (1 - y_i(\mathbf{x}_i^T \boldsymbol{\beta} + \beta_0))_+ \; + \; \sum_{j=1}^{p} p_\lambda(\|\beta_j\|), \tag{11}$$

where $p_\lambda(\cdot)$ is a symmetric, non-convex penalty function with a tuning parameter λ. Oracle properties were developed under some regulatory conditions, and some common penalty functions such as the smoothly clipped absolute deviance (SCAD) [14] penalty and the minimax concave penalty (MCP) [28] were explored.

However, such a feature selection process screens features in the input space. As shown in Lemma 1 in the Appendix A, when the classes are not linearly separable in the input space, the SVM will map the input space into projection space and the linear boundary is obtained in the projected space. As the kernel function controls the classifier's performance, a straightforward idea is then to select features and enhance the performance of the SVM by directly introducing penalty terms to the kernel function. Based on this exploration, we propose a method of simultaneous feature selection and classification by penalizing the kernel function in SVM through a data-adaptive kernel procedure.

3. Methodology of Data-Adaptive Kernel-Penalized SVM

In this section, a data-adaptive kernel-penalized SVM is proposed. The method can simultaneously select features and conduct classifications with a data-adaptive kernel function. Instead of adding a penalty to the standard hinge loss function, we propose to introduce the penalty term directly into the SVM under the kernel formulation, so that the number of predictors are controlled. To accommodate the common imbalance issue in real applications, a data-adaptive kernel will be employed. Thus, the oracle properties of the estimator of the true parameters under the proposed setting are developed.

3.1. Kernel-Based Parameters

We focus on the Gaussian radial basis function kernel (RBF kernel) as in (6) to develop the proposed method, where the parameter σ is originally assumed to be universal for all components of the input vectors $\mathbf{x}$. Actually, the parameter σ can be extended to be component-specific as

$$K(\mathbf{x}, \mathbf{z}) = \exp\big(-\sum_{j=1}^{p} (x_j - z_j)^2 / 2\sigma_j^2\big), \tag{12}$$

where p is the dimension of the input space I [6]. Consequently, the contributions of the corresponding predictors can be determined by the parameters $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \ldots, \sigma_p)$. For instance, if σ_j is very large,

the j-th predictor tends to contribute very little to the kernel function as the corresponding component in the exponent will be close to zero. Contrarily, if σ_j is small, the contribution of the j-th predictor will be large and its importance increases consequently. Thus, by controlling the j-th component in the parameter vector $\boldsymbol{\sigma}$, the importance of the j-th predictor can be determined. This provides a method of feature selection by directly estimating the parameters in the kernel function. Accordingly, we propose the following modification of the kernel function as

$$K(\mathbf{x}, \mathbf{z}; \mathbf{w}) = \exp\{-\|\mathbf{w} \otimes (\mathbf{x} - \mathbf{z})\|^2\}, \tag{13}$$

where $\mathbf{w} = (w_1, w_2, \ldots, w_p) = (1/\sigma_1, 1/\sigma_2, \ldots, 1/\sigma_p)$ and $\otimes$ represents the component-wise product. That is, $\mathbf{w}$ assigns weights for the contribution of each component to the kernel. When w_j is large, the contribution of the j-th feature will be large and hence its importance increases. Contrarily, when w_j is small, the j-th predictor tends to contribute little to the kernel function, and might not be included during the construction of an SVM. However, even if the absolute value of w_j is small (not zero), its influence in the kernel function still exists. Including too many active features in the classifier may dramatically complicate the model and result in extra noisy information. Forcing the effect of some features to be exactly zero may therefore solve such an issue. This can be achieved by introducing a penalty to penalize the weights $\mathbf{w}$ under the assumption that the number of active features are sparse.

3.2. Data-Adaptive Kernel Functions

To deal with the imbalance of the data and enhance the performance of the SVM, we employ the data-adaptive kernel function when constructing the SVM. The data-adaptive kernel SVM is a two stage procedure, where the SVM is applied in the first stage to identify a temporary boundary, and the kernel function is modified adaptively in the second stage based on the boundary and support vectors identified in the first stage. It is proven to have the capability of increasing the separability between two classes by enlarging the spatial resolution around the boundary surface. This is especially important when the data are imbalanced. It has been demonstrated that the imbalance of classes can severely affect the performance of an SVM [29]. To illustrate, let $\mathbf{f}$ be the mapped result of $\mathbf{x} \in \mathbb{R}^p$ in F, i.e., $\mathbf{f} = \mathbf{s(x)} \in \mathbb{R}^l$. A small change in $\mathbf{x}$ in the input space, d$\mathbf{x}$, will be mapped into the vector d$\mathbf{f}$ in the feature space such that

$$\mathrm{d}\mathbf{f} \;=\; \nabla \mathbf{s} \cdot \mathrm{d}\mathbf{x} = \sum_j \frac{\partial}{\partial x_j} \mathbf{s}(\mathbf{x})\, \mathrm{d}x_j,$$

where

$$\nabla \mathbf{s} \;=\; \left(\frac{\partial\, \mathbf{s(x)}}{\partial\, \mathbf{x}}\right) = \begin{pmatrix} \frac{\partial\, s_1(\mathbf{x})}{\partial\, x_1} & \cdots & \frac{\partial\, s_1(\mathbf{x})}{\partial\, x_p} \\ \vdots & \vdots & \vdots \\ \frac{\partial\, s_l(\mathbf{x})}{\partial\, x_1} & \cdots & \frac{\partial\, s_l(\mathbf{x})}{\partial\, x_p} \end{pmatrix}.$$

Thus, the squared length of d$\mathbf{f}$ can be written in the quadratic form as

$$\|\mathrm{d}\mathbf{f}\|^2 = (\sum_i \frac{\partial}{\partial x_i} \mathbf{s}(\mathbf{x}) \mathrm{d}x_i)^T \cdot (\sum_j \frac{\partial}{\partial x_j} \mathbf{s}(\mathbf{x}) \mathrm{d}x_j) = \sum_{ij} s_{ij}(\mathbf{x}) \mathrm{d}x_i \mathrm{d}x_j. \tag{14}$$

where $s_{ij}(\mathrm{x})$ can be regarded as a local magnification factor [26]. To enlarge the spatial separation around the boundary, the kernel function will be adapted based on the data. Let $C(\mathbf{x}, \mathbf{x}')$ be a positive scalar function such that

$$C(\mathbf{x}, \mathbf{x}') \;=\; c(\mathbf{x}) c(\mathbf{x}'),$$

where $\mathbf{x}$ and $\mathbf{x}'$ are vectors of features in the input space, and $c(\mathbf{x})$ is a positive univariate scalar function. Then, the kernel function K is updated as

$$\tilde{K}(\mathbf{x}, \mathbf{x}') \;=\; C(\mathbf{x}, \mathbf{x}') K(\mathbf{x}, \mathbf{x}') \;=\; c(\mathbf{x}) K(\mathbf{x}, \mathbf{x}') c(\mathbf{x}'), \tag{15}$$

where $K(\mathbf{x}, \mathbf{x}')$ is the kernel function in the first stage and $\tilde{K}(\mathbf{x}, \mathbf{x}')$ in the updated kernel in the second stage. It can be viewed as a modification of the original mapping **s(x)** to a new mapping function $\tilde{\mathbf{s}}(\mathbf{x})$, satisfying

$$\tilde{s}_{ij}(\mathbf{x}) \;=\; c_{ij}(\mathbf{x}) s_{ij}(\mathbf{x}).$$

where $s_{ij}(\mathbf{x})$ is defined in (14) and $c_{ij}(\mathbf{x}) = \frac{\partial}{\partial x_i}\frac{\partial}{\partial z_j} C(\mathbf{x}, \mathbf{z})|_{\mathbf{z}=\mathbf{x}}$. This process is referred to as adaptive scaling, and $\tilde{K}$ can be easily shown to satisfy the Mercer positivity condition, which is the sufficient condition for a real function to be a kernel function [22]. When $\tilde{s}_{ij}(\mathbf{x})$ has larger values at the support vectors than other data points, the updated mapping $\tilde{\mathbf{s}}$ can increase the separation when a positive function $c(\mathbf{x})$ is properly chosen. In particular, when the kernel function is Gaussian, we have the following result derived from Amari and Wu [26].

Theorem 1. *When a Gaussian radial basis kernel in* (6) *is used, the modified magnification factor is*

$$\tilde{s}_{ij}(\boldsymbol{x}) \;=\; c_i(\boldsymbol{x}) c_j(\boldsymbol{x}) \;+\; c^2(\boldsymbol{x}) s_{ij}(\boldsymbol{x}) \;=\; c_i(\boldsymbol{x}) c_j(\boldsymbol{x}) \;+\; \frac{c^2(\boldsymbol{x})}{\sigma^2} I(i=j),$$

where $c_i(\boldsymbol{x}) = \partial c(\boldsymbol{x}) / \partial x_i$, *and* $I(\cdot)$ *is the indicator function.*

Thus, to make $\tilde{\mathrm{s}}$ bigger, we need to make the positive scalar $c(\mathbf{x})$ and its first-order derivative relatively large. The authors propose to adaptively scale the primary kernel function K by constructing $c(\mathbf{x})$ with the L_1 norm radial basis function

$$c(\mathbf{x}) \;=\; e^{-|D(\mathbf{x})| \cdot k_M(\mathbf{x})} \tag{16}$$

and

$$k_M(\mathbf{x}) \;=\; \frac{1}{|N_M(\mathbf{x})|} \sum_{i \in N_M(\mathbf{x})} (\|\mathbf{s}(\mathbf{x}_i) - \mathbf{s}(\mathbf{x})\|^2), \tag{17}$$

where $D(\mathbf{x})$ is the (1) obtained from the first stage, $N_M(\mathbf{x}) = \{j : \|\mathbf{s}(\mathbf{x}_j) - \mathbf{s}(\mathbf{x})\|^2 < M,\ y_j \neq y\}$, $|A|$ is the cardinality of the set A, y is the class label associated with $\mathbf{x}$, and M can be regarded as the distance between the nearest and the farthest support vectors from $\mathbf{s}(\mathbf{x})$. This process is important when the data are imbalanced, since by incorporating $k_M(\mathbf{x})$ into $c(\mathbf{x})$, the adaptive scaling process updates the spatial information and balances the data locally by considering only the support vectors , which determine the location of the decision hyperplane, from the opposite class near the boundary. This method has been proved to have greater separability even when the data are imbalanced. The magnification effect is roughly the largest near the initial separating boundary, and decreases robustly with a slow and steady rate from the separating boundary to faraway locations. We will incorporate this data-adaptive kernel procedure to accommodate imbalance classes.

3.3. Data-Adaptive Kernel-Penalized SVM

To control the number of features in the classifier, a penalty term for the weights $p_\lambda(\|\mathbf{w}\|)$ will be introduced to select the features through the kernel function. We propose to add the penalty term directly to the dual maximization problem for the SVM which contains the kernel function. Specifically, the data-adaptive kernel-penalized SVM is initially proposed as the solution to

$$\underset{\boldsymbol{\alpha},\, \mathbf{w}}{\mathrm{Max}} \; \Big\{ \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \tilde{K}(\mathbf{x}_i, \mathbf{x}_j; \mathbf{w}) - \sum_{j=1}^{p} p_\lambda(|\mathrm{w}_j|) \Big\}, \tag{18}$$

such that

$$\begin{aligned}
\sum_{i=1}^{n} \alpha_i y_i &= 0, \\
0 \leq \alpha_i &\leq B, \qquad i = 1, 2, \ldots, l \\
\mathrm{w}_j &\geq 0, \qquad j = 1, 2, \ldots, p,
\end{aligned}$$

where $\tilde{K}(\mathbf{x}, \mathbf{z})$ is the data-adaptive kernel function from (15), $c(\mathbf{x})$ in $\tilde{K}(\mathbf{x}, \mathbf{z})$ is from (16) and the primary kernel function is from (13). When the estimate of $\widehat{\mathbf{w}}$ is obtained, the predictors with non-zero coefficients are considered to be the truly active predictors that will affect the decision boundary. The boundary will be estimated by

$$\widehat{D}(\mathbf{x}) = \sum_{i \in SV} \alpha_i y_i \tilde{K}(\mathbf{x}_i, \mathbf{x}_j; \widehat{\mathbf{w}}) + \widehat{b} \tag{19}$$

and the intercept b can be estimated

$$\widehat{b} = \frac{1}{|SV|} \sum_{j \in SV} \{ y_j - \sum_{i \in SV} \alpha_i y_i \tilde{K}(\mathbf{x}_i, \mathbf{x}_j; \widehat{\mathbf{w}}) \}. \tag{20}$$

with the decision rule in (19), a test observation $\mathbf{x}$ can be assigned to the class by the sign of $\widehat{D}(\mathbf{x})$.

There are several options for the specific forms of the penalty. In general, non-convex penalties satisfying the following assumptions A1 and A2 can be used.

A1. The penalty function $p_{\lambda_n}(x)$ is symmetric, non-decreasing and concave for $x \in [0, \infty)$, with a continuous first-order derivative $p'_{\lambda_n}(x)$ on R^+ and $p'_{\lambda_n}(0) = 0$.

A2. There exists $a > 1$, such that $\lim_{x \to 0+} p'_{\lambda_n}(x) = \lambda_n$, $p'_{\lambda_n}(x) \geq \lambda_n - x/a$ for $0 < x < a\lambda$ and $p'_{\lambda_n}(x) = 0$ for $x \geq a\lambda$.

Such a non-convex penalty term is motivated by the fact that the L_1 LASSO penalty does not have the oracle property due to the over-penalization of large weights, and hence the LASSO penalty is not a proper choice when high dimensional features are involved in classification [8]. Several popularly used non-convex penalties satisfy assumptions A1 and A2:

1. SCAD: smoothly clipped absolute deviation [14]

$$\begin{aligned}
p_\lambda(|\mathbf{w}|) &= \lambda |\mathbf{w}| I(0 \leq |\mathbf{w}| < \lambda) + \frac{a\lambda |\mathbf{w}| - (\mathbf{w}^2 + \lambda^2)/2}{a - 1} I(\lambda \leq |\mathbf{w}| \leq a\lambda) \\
&+ \frac{(a+1)\lambda^2}{2} I(|\mathbf{w}| > a\lambda) \text{ for some a} > 2.
\end{aligned}$$

2. MCP: minimax concave penalty [28]

$$p_\lambda(|\mathbf{w}|) = \lambda(|\mathbf{w}| - \frac{\mathbf{w}^2}{2a\lambda}) I(0 \leq |\mathbf{w}| < a\lambda) + \frac{a\lambda^2}{2} I(|\mathbf{w}| \geq a\lambda) \quad \text{for some a} > 1.$$

3. L_0 norm smooth approximation: $\|\mathbf{w}\|_0 = |\{i : \mathrm{w}_i \neq 0\}|$ by [11]. Unlike L_p norm with $p > 0$, L_0 norm is not precisely a norm because the triangle inequality does not hold and, consequently, it is not smooth. Thus, the approximation by a concave function is applied on the L_0 norm so that a penalty function is

$$p_\lambda(|\mathbf{w}|) = \mathbf{1}^T(\mathbf{1} - \exp(\lambda |\mathbf{w}|)) \approx \|\mathbf{w}\|_0,$$

where λ is an approximation parameter.

Remark 1. *A penalty is usually added to the loss function in the literature; however, the standard loss function does not contain the kernel function. When the data are imbalanced, the performance of a standard SVM will be affected. Consequently, the features selected without considering the imbalance of classes may be unreliable in the imbalance application. Contrarily, data-adaptive kernel-penalized SVM can fulfil the feature selection process while taking the imbalance of classes into account.*

Remark 2. *Although other types of kernels such as the polynomial kernel $K(\boldsymbol{x}, \boldsymbol{z}) = (1 + \sum_{j=1}^{p} x_j z_j)^d$ are also available to describe the mapping by kernels, not all the kernels are feasible for simultaneous feature selection process classification because of technical difficulties. For example, polynomial kernels are determined only by the order parameter d, while it is not obvious how feature selection can be conducted during the classification process. However, the proposed method is still very attractive in applications, since the Gaussian RBF kernel adopted here is the most popular kernel.*

Remark 3. *The constraints in the dual function contain the non-negativity of the parameters $\boldsymbol{w}$—they correspond to the positive scale parameter in the Gaussian kernels. This constraint can be removed by using a quadratic form of the parameters in the penalized kernels.*

3.4. An Algorithm to Solve Data-Adaptive Kernel-Penalized SVM

To solve the data-adaptive kernel-penalized SVM in (18), a two-stage algorithm is proposed. In the first stage, a standard SVM is obtained so that the location information of the support vectors and the temporary decision boundary are available. The primary kernel function is then updated adaptively by (15) in the second stage, and the optimization with both the updated kernel and the penalty is then solved to obtain the final boundary as well as the selected features.

Since the objective function in (15) is non-convex, an iterative procedure is adopted [11]. To be specific, in the t-th round iteration, $t = 1, 2, \ldots, T$, a standard dual optimization problem for an SVM with the $(t-1)$-th estimated kernel parameter vector $\widehat{\mathbf{w}}^{(t-1)}$ is to be solved as

$$\underset{\boldsymbol{\alpha}}{\text{Max}}\ L_1(\boldsymbol{\alpha}) \quad = \quad \underset{\boldsymbol{\alpha}}{\text{Max}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j; \widehat{\mathbf{w}}^{(t-1)}) \tag{21}$$

such that

$$\begin{aligned} \sum_{i=1}^{n} \alpha_i y_i &= 0, \\ 0 \le \alpha_i &\le B, \qquad i = 1, 2, \ldots, n, \end{aligned}$$

and the result is denoted as $\boldsymbol{\alpha}^{(t)}$. During this stage, the support vectors are obtained by those non-zero α_is, and $c(\mathbf{x})$ can be constructed through (16) so that the data-adaptive kernel function $\tilde{K}(\mathbf{x}, \mathbf{z})$ can be constructed by (15).

Finally a non-linear formulation with a fixed $\boldsymbol{\alpha}^{(t)}$ is solved

$$\widehat{\mathbf{w}}^{(t)} = \arg\min_{\mathbf{w}} \ \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i^{(t)} \alpha_j^{(t)} y_i y_j \tilde{K}(\mathbf{x}_i, \mathbf{x}_j; \mathbf{w}) + \sum_{j=1}^{p} p_\lambda(|\mathrm{w}_j|) \tag{22}$$

such that

$$\mathrm{w}_j \ge 0, \qquad j = 1, 2, \ldots, p.$$

The process will stop when $\|\mathbf{w}^{(t)} - \mathbf{w}^{(t-1)}\|$ is sufficiently small.

3.5. The Oracle Property

In this subsection, we develop the oracle property of the estimator. We show that, under some regularity conditions, the distance between the estimates and the true values of the parameters goes to zero with probability 1 when the sample size is sufficient large. Here, we only need to consider the optimization process in the second stage in (22), since all the unknown information regarding the parameters $\mathbf{w}$ is included in this stage (note that $\boldsymbol{\alpha}$ is considered as a fixed constant vector in the second stage). Define the estimator

$$\hat{\mathbf{w}} = \arg\min \{\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i,\mathbf{x}_j;\mathbf{w}) + \sum_{j=1}^{p} p_\lambda(|\mathrm{w}_j|)\}$$

and the following regularity conditions:

C1. The densities of $\mathbf{Z}$ given $Y = 1$ and -1 are continuous with common support in R^q, where $\mathbf{Z}$ are truly relevant predictors.
C2. $E(Z_j^2) < \infty$ for $1 \le j \le q$, i.e., the second order moments of all active predictors are finite.
C3. The true parameter $\boldsymbol{\beta}_0$ is a non-zero and unique vector.
C4. $q = O(n^c)$ for some $0 \le c < 1/2$, namely, $\lim_{n\to\infty} q/n^c < \infty$.
C5. Eigenvalues of $n^{-1}[\mathbf{X}^{\odot 2}]^T\mathbf{X}^{\odot 2}$ are finite, where $\mathbf{X}$ is the input matrix, and $(\cdot)^{\odot 2}$ is the component-wise square.

Conditions C1–3 are the assumptions to ensure that the oracle estimator constructed in our proposed method is consistent and that the optimal classification decision rule is not constant. Condition C4 is a common requirement in high-dimensional inference, indicating that the the number of the truly active predictors cannot diverge with a rate faster than $\sqrt{n}$. Condition C5 gives the upper boundary of the largest eigenvalues of the squared design matrix, which is necessary in our proposed method due to the quadratic form in the radial kernel functions. With these conditions, the following oracle property holds:

Theorem 2. *Assume that Conditions C1-5 and Assumptions 1-2 for the penalty are satisfied. If* $\max\{|p''_\lambda(w_j)| : w_j \neq 0\} \to 0$, *then there exists a local minimizer* $\hat{\boldsymbol{w}}$ *of* $L_2(\boldsymbol{w}) = \{\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j K(\boldsymbol{x}_i,\boldsymbol{x}_j;\boldsymbol{w}) + \sum_{j=1}^{p} p_\lambda(|w_j|)\}$ *such that* $\|\hat{\boldsymbol{w}} - \boldsymbol{w}_{true}\| = O_p\{\sqrt{q/n}\}$, *where* $\boldsymbol{w}_{true}$ *is the true value of* $\boldsymbol{w}$.

Detailed proof is provided in Appendix A. Theorem 2 guarantees that the estimate of the parameter in the proposed method acts as if the true values of the parameters were known. When the sample size is sufficiently large, the distance between the estimates and the true values of the parameters will be small enough. Consequently, the estimated decision rule in (19) can be obtained as if the true decision boundary were known, and it can then be employed to classify new observations.

Though various approaches for SVM-based feature selection procedures are available in literature, the proposed method is different in that it directly obtains a minimal subset of features and simultaneously classifies objects by penalizing the kernel function, eliminating noisy features without ranking the features. The process of the proposed method is more time-efficient compared to the methods in the literature, and the proposed method improves the classification performance, especially when the data are imbalanced.

4. Numerical Studies

In this section, simulation studies are carried out to assess the performance of the data-adaptive kernel-penalized SVM, and to compare the proposed method with some other penalized SVMs in the literature. In the data-adaptive (DA) kernel-penalized SVM, the SCAD (DA-SCAD-SVM), MCP (DA-MCP-SVM) penalties and L_0 norm approximation (DA-L0-SVM) are used. For other penalized SVMs, we use the penalties of SCAD (SCAD-SVM, [8]), MCP (MCP-SVM, [8]), L_1 norm

(L_1-SVM, [30]), adaptively weighted L_1 norm with a weight parameter $c = 0.5$ (adapt L_1-SVM, [10]) and L_0 norm approximation (L_0-SVM, [11]). The comparisons are made under various levels of imbalance in the data. The abilities of identifying the relevant features and controlling the test error are compared when the data are both balanced and imbalanced.

4.1. Simulation Study

We consider the data generation process of a standard discriminant analysis following the settings from [21] and [8]. The model is described as $Pr(Y = 1) = c$ while $Pr(Y = -1) = 1 - c$, where c will control the imbalance level. The input features $X|(Y = 1) \sim MVN(\boldsymbol{\mu}, \Sigma)$ and $X|(Y = -1) \sim MVN(-\boldsymbol{\mu}, \Sigma)$, $\boldsymbol{\mu} = (0.1,\ 0.2,\ 0.3,\ 0.4,\ 0.5,\ 0, \ldots,\ 0)^T \in \mathbf{R}^p$, $\Sigma = (\sigma_{ij})$ with diagonal elements $\sigma_{ii} = 1$ for $i = 1, 2, \ldots, p$ and $\sigma_{ij} = \rho = -0.2$ for $1 \leq i \neq j \leq K$, and K is set as 5. The true label is determined by the Bayes rule boundary as $sgn(1.5X_1 + 2.3X_2 + 2.8X_3 + 3.3X_4 + 3.8X_5)$ with a Bayes error of 6.1%.

In terms of tuning the regularization parameters for all of the approaches considered, we adopt a procedure similar to [31]. The prediction error is estimated using a five-fold cross-validation method. The initial value of $\mathbf{w}$ is set as $\mathbf{1}^T$. During the second stage of solving the data-adaptive kernel-penalized SVM, the gradient descent procedure is adopted for the non-linear optimization problem. The iterative algorithm will stop if the change in the estimates of $\mathbf{w}$ in two consecutive rounds, namely $\|\mathbf{w}^{(t+1)} - \mathbf{w}^{(t)}\|$, is smaller than a given threshold ϵ, which is set as 10^{-4} for fast convergence.

For the tuning parameter λ in the penalty term, we use the SVM-extended Bayesian information criterion (SVMIC) proposed in [8] as

$$SVMIC_\gamma(S) = \sum_{i=1}^{n} 2c\xi_i + \log n|S| + 2\gamma \binom{p}{|S|}, \tag{23}$$

where ξ_i, $i = 1, 2, \ldots, n$, are the optimal slack predictors and, correspondingly, S is a subset of $\{1, 2, \ldots, p\}$, $|S|$ is the cardinality of S, and $\binom{\cdot}{\cdot}$ represents the combination operator. This idea is motivated by the standard Bayesian information criterion and is extended by [32]. The range of λ is set as $\{2^{-6}, 2^{-5}, \ldots, 2^3\}$, and γ is set as 0.5 in the tuning procedure without a loss of generality [32]. The value of λ will be set as the one that maximizes (23). Note that the values of the slack variables ξ_i in (23) are not available directly, but they can be calculated by $\xi_i = [1 - y_i \widehat{D}(\mathbf{x}_i)]_+$ for $i = 1, \ldots, n$, where $[t]_+ = \max\{0, t\}$, and $\widehat{D}(\mathbf{x}_i)$ can be obtained by (19) [33].

As suggested in [8], for $SCAD$ and MCP penalties, the constant a values will be set as 3.7 and 3, respectively.

Tables 1 and 2 summarize the performances with different combinations of imbalance levels and numbers of predictors, based on a replication of 100 times. The sample sizes n are fixed as 100 and 400, respectively. The 'Relevant' and 'Irrelevant' columns show the information of the mean values of the truly active and inactive predictors selected by the model, respectively. Column 'True' gives the percentage when the true model, containing exactly those five active predictors, is correctly selected during the 100 replications. Values in parentheses are the corresponding empirical standard errors.

Table 1. Simulation study outcome where the sample size $n = 100$. Margins are provided in brackets.

Method	Proportion	p	Relevant	Irrelevant	True%	Test Error%
DA-SCAD-SVM	c = 0.50	50	5.00(0.00)	0.88(0.16)	96	8.16(0.2)
		100	5.00(0.00)	0.91(0.14)	96	8.72(0.2)
	c = 0.75	50	4.96(0.01)	0.92(0.23)	94	9.23(0.3)
		100	4.95(0.01)	0.95(0.27)	94	9.85(0.3)
	c = 0.90	100	4.91(0.03)	1.10(0.39)	91	10.55(0.4)
		100	4.90(0.03)	1.09(0.41)	91	10.93(0.4)

Table 1. *Cont.*

Method	Proportion	p	Relevant	Irrelevant	True%	Test Error%
DA-MCP-SVM	c = 0.50	50	5.00(0.00)	0.12(0.01)	98	7.20(0.2)
		100	5.00(0.00)	0.13(0.01)	98	7.38(0.2)
	c = 0.75	50	4.98(0.01)	0.26(0.03)	96	8.44(0.2)
		100	4.98(0.01)	0.28(0.03)	96	8.90(0.2)
	c = 0.90	100	4.95(0.02)	0.42(0.04)	92	9.20(0.3)
		100	4.94(0.02)	0.45(0.04)	92	9.65(0.3)
DA-L0-SVM	c = 0.50	50	5.00(0.00)	0.36(0.02)	97	7.81(0.2)
		100	5.00(0.00)	0.39(0.02)	97	7.86(0.2)
	c = 0.75	50	4.97(0.01)	0.47(0.03)	95	8.02(0.2)
		100	4.96(0.01)	0.51(0.03)	95	8.10(0.2)
	c = 0.90	100	4.92(0.02)	0.68(0.04)	91	9.70(0.3)
		100	4.92(0.02)	0.65(0.04)	90	9.82(0.3)
SCAD-SVM	c = 0.50	50	4.92(0.02)	1.92(0.18)	96	8.23(0.2)
		100	4.91(0.02)	1.99(0.17)	96	8.66(0.2)
	c = 0.75	50	4.83(0.03)	2.01(0.31)	91	10.19(0.4)
		100	4.78(0.04)	2.13(0.36)	91	10.87(0.4)
	c = 0.90	100	4.76(0.04)	3.35(0.41)	88	12.15(0.5)
		100	4.74(0.04)	3.40(0.43)	87	12.36(0.5)
MCP-SVM	c = 0.50	50	5.00(0.00)	0.27(0.02)	98	7.32(0.2)
		100	5.00(0.00)	0.29(0.02)	98	7.41(0.2)
	c = 0.75	50	4.92(0.01)	0.43(0.03)	93	8.96(0.2)
		100	4.91(0.01)	0.47(0.03)	93	9.29(0.3)
	c = 0.90	100	4.85(0.03)	0.88(0.05)	89	10.63(0.4)
		100	4.84(0.03)	0.91(0.05)	89	11.79(0.4)
L_1-SVM	c = 0.50	50	4.86(0.05)	31.08(1.52)	10	16.67(0.5)
		100	4.71(0.06)	42.98(2.13)	4	19.33(0.6)
	c = 0.75	50	4.62(0.07)	35.71(1.67)	3	19.18(0.6)
		100	4.45(0.08)	46.29(2.20)	0	22.00(0.8)
	c = 0.90	50	4.33(0.10)	39.53(2.02)	1	22.61(0.8)
		100	4.02(0.10)	59.01(2.54)	0	25.98(1.0)
Adapt L_1-SVM	c = 0.50	50	4.38(0.07)	13.62(0.90)	23	16.28(0.5)
		100	4.01(0.10)	13.10(0.86)	5	20.23(0.5)
	c = 0.75	50	4.13(0.09)	15.18(1.05)	8	18.71(0.5)
		100	3.91(0.10)	14.92(1.03)	0	22.33(0.6)
	c = 0.90	50	3.87(0.10)	16.99(1.22)	2	20.02(0.6)
		100	3.81(0.13)	16.87(1.21)	0	25.01(0.7)
L_0-SVM	c = 0.50	50	4.85(0.02)	2.87(0.66)	62	12.16(0.5)
		100	4.78(0.04)	2.93(0.49)	54	14.16(0.4)
	c = 0.75	50	4.61(0.04)	4.11(0.23)	55	13.88(0.4)
		100	4.37(0.08)	4.23(0.56)	43	15.73(0.4)
	c = 0.90	50	4.33(0.07)	6.28(0.77)	41	16.68(0.5)
		100	4.03(0.10)	6.79(0.78)	25	17.02(0.5)

Table 2. Simulation study outcome where the sample size $n = 400$. Margins are provided in brackets.

Method	Proportion	p	Relevant	Irrelevant	True%	Test Error%
DA-SCAD-SVM	c = 0.50	200	5.00(0.00)	0.58(0.11)	98	7.76(0.2)
		400	5.00(0.00)	0.72(0.13)	98	8.13(0.2)
	c = 0.75	200	4.98(0.01)	0.67(0.12)	96	8.76(0.3)
		400	4.98(0.01)	0.71(0.13)	96	9.12(0.3)
	c = 0.90	200	4.95(0.02)	0.81(0.17)	93	9.14(0.3)
		400	4.94(0.02)	0.77(0.16)	93	9.93(0.3)
DA-MCP-SVM	c = 0.50	200	5.00(0.00)	0.05(0.01)	98	6.28(0.2)
		400	5.00(0.00)	0.06(0.01)	98	6.91(0.2)
	c = 0.75	200	4.98(0.01)	0.12(0.04)	97	7.45(0.2)
		400	4.98(0.01)	0.11(0.04)	97	7.93(0.2)
	c = 0.90	200	4.95(0.02)	0.18(0.05)	94	8.60(0.2)
		400	4.94(0.02)	0.19(0.05)	94	9.11(0.3)
DA-L0-SVM	c = 0.50	200	5.00(0.00)	0.26(0.01)	98	7.02(0.2)
		400	5.00(0.00)	0.28(0.01)	98	7.12(0.2)
	c = 0.75	200	4.98(0.01)	0.33(0.08)	96	7.88(0.2)
		400	4.98(0.01)	0.36(0.08)	97	8.02(0.2)
	c = 0.90	200	4.95(0.02)	0.44(0.10)	93	9.15(0.2)
		400	4.94(0.02)	0.49(0.10)	93	9.54(0.3)
SCAD-SVM	c = 0.50	200	4.96(0.01)	1.52(0.15)	96	8.01(0.2)
		400	4.96(0.01)	1.76(0.16)	96	8.36(0.2)
	c = 0.75	200	4.88(0.03)	1.77(0.16)	92	9.59(0.3)
		400	4.82(0.04)	1.98(0.18)	92	10.27(0.4)
	c = 0.90	200	4.82(0.04)	2.89(0.36)	90	11.32(0.5)
		400	4.77(0.04)	3.11(0.40)	89	11.87(0.4)
MCP-SVM	c = 0.50	200	5.00(0.00)	0.27(0.02)	98	7.32(0.2)
		400	5.00(0.00)	0.29(0.02)	98	7.41(0.2)
	c = 0.75	200	4.92(0.01)	0.43(0.03)	93	8.96(0.2)
		400	4.91(0.01)	0.47(0.03)	93	9.29(0.3)
	c = 0.90	200	4.85(0.03)	0.88(0.05)	89	10.63(0.4)
		400	4.84(0.03)	0.91(0.05)	89	11.79(0.4)
L_1-SVM	c = 0.50	200	4.88(0.04)	25.08(1.22)	15	14.91(0.4)
		400	4.79(0.06)	28.66(1.56)	8	17.76(0.5)
	c = 0.75	200	4.65(0.07)	28.12(1.54)	5	16.53(0.5)
		400	4.45(0.08)	31.67(1.53)	1	20.35(0.7)
	c = 0.90	200	4.43(0.09)	33.53(1.61)	0	19.53(0.6)
		400	4.11(0.09)	40.27(2.08)	0	23.16(0.9)
Adapt L_1-SVM	c = 0.50	200	4.49(0.08)	11.28(0.90)	35	13.28(0.5)
		400	4.25(0.9)	13.10(0.86)	16	16.55(0.6)
	c = 0.75	200	4.25(0.09)	13.65(1.05)	17	15.97(0.5)
		400	4.12(0.09)	14.16(1.03)	6	18.46(0.6)
	c = 0.90	200	3.87(0.10)	14.85(1.22)	5	18.98(0.6)
		400	4.01(0.10)	15.26(1.21)	1	21.98(0.7)
L_0-SVM	c = 0.50	200	4.88(0.02)	2.42(0.66)	77	11.42(0.5)
		400	4.82(0.02)	2.65(0.23)	60	12.91(0.5)
	c = 0.75	200	4.73(0.04)	3.69(0.30)	65	12.51(0.5)
		400	4.49(0.06)	3.82(0.23)	48	13.80(0.5)
	c = 0.90	200	4.46(0.06)	5.52(0.63)	47	15.23(0.5)
		400	4.33(0.07)	6.18(0.76)	29	16.45(0.6)

In general, the SVMs with the non-convex penalized data-adaptive kernels show a much greater probability of correctly selecting the true model as n increases, which is consistent with the asymptotic oracle property. According to the numbers in the Relevant column, the SVMs with penalties of $SCAD$ and MCP find the most relevant predictors compared with other methods. The SVM with L_0 norm approximation can find some relevant predictors, while the SVMs with an L_1 norm penalty tend to fail in selecting the correct predictors, with or without adaptive weights. According to the Irrelevant column, the two data-adaptive kernel-penalized methods exclude most irrelevant predictors and hence eliminate the noisy predictors. The missing relevant predictor, if there is any, is mostly from X_1 due to the fact that setting X_1 has the weakest effect.

On the other hand, when the imbalance level of the data is increasing, the prediction error tends to increase. However, given a specific level of imbalance in data, test prediction errors from data-adaptive kernel-penalized SVMs are universally smaller than those obtained from other approaches, because these two methods give the fewest noisy predictors so that the prediction error is minimized. More importantly, when the imbalance level increases, our data-adaptive kernel-penalized SVMs outperform among all methods, which agrees with the fact that the data-adaptive kernel can improve the classification performance. This adaptive scaling process on the kernel is only applicable to our setting and not to any other method due to the lack of kernel functions in the model structures (penalized SVMs have penalty terms directly on the loss function, which is not described in the kernel form). In the mean time, the feature selection performance changes little, especially in the non-convex penalized data-adaptive kernel SVMs.

It is worth noting that the combination (n, p) shows that, even when the number of potential predictors is proportional to the sample size or larger, our method still performs well. This gives us some clue that the method may still work in big data or ultra-high dimensional settings. Indeed, the oracle property in our proposed method indicates that the true predictors can still be selected even when the dimension of the input space grows proportional to the sample size, which is the high-dimensional setting.

4.2. A Real Data Example

A publicly available Wisconsin Breast Cancer (WBC) data set from the UCI Machine Learning Repository [34] provides an illustration of the proposed method. The data set can be found and downloaded via https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic). The WBC data set contains 569 observations (212 malignant and 357 benign tumors). Thirty continuous features, such as the radius (mean of distances from center to points on the perimeter) of the cancer, as well as the texture (standard deviation of gray-scale values), smoothness (local variation in radius lengths) and area of the cancer, are expected to be used to classify the two classes of malignant and benign tumors. These features are measured by a digitized image of a Fine Needle Aspirate (FNA) of a breast mass, which can describe the cell nuclei shown in the images. We refer readers to a full description of the data set in [35]. As a pre-process step, the features were first standardized.

Different methods are applied to the data set, both with and without penalties. For classifiers without penalties, the Gaussian kernel will be adopted with all the input features being used to estimate the decision boundary. For those with penalties, we will use data-adaptive kernel-penalized SVMs with SCAD and MCP penalties, as well as the penalized SVMs with SCAD and MCP penalties to the hinge loss.

The numbers of selected features and the test errors from all the considered methods will be reported. For those approaches that require a two-stage optimization process, the solutions for the first stage optimization are used as the initial values for the second stage optimization if needed. For SCAD and MCP penalties, the constant a values are still fixed as 3.7 and 3, respectively, the same as the values

used in the simulation process. A five-fold cross validation will be conducted to obtain the tuning parameters, which will be chosen from the following sets

$$B \in \{0.1, 0.5, 1, 5, 10, 20, 50, 80, 100, 200, 500\}$$
$$\sigma \in \{0.1, 0.5, 1, 2, 3, 4, 5, 10, 50, 100\}.$$

The tuning parameter λ is selected by the grid search from $\{2^{-14}, 2^{-9}, \ldots, 2^{5}\}$ in 100 repetitions.

Table 3 summarizes the classification outcome of the mean and the standard deviation (in parentheses) of the prediction error and the number of predictors selected with different approaches. It is clear that the data-adaptive kernel-penalized SVMs perform the best among all approaches, with a significantly lower prediction error and number of predictors selected than any other method. Compared with penalized SVM with SCAD and MCP penalties, data-adaptive kernel-penalized SVMs with the corresponding penalties still outperform, even though the penalties are the same. MCP seems to be a better choice for the penalty term, since the number of the predictor is the smallest, and the standard deviation is smaller. Adaptively weighted L_1 norm SVM and L_1 norm SVM are fair. Clearly, the numerical results have confirmed that data-adaptive kernel-penalized SVMs with SCAD or MCP penalties are both promising classifiers with low prediction errors and excellent feature selection abilities.

Table 3. Classification outcome on the Wisconsin Breast Cancer data set. Margins are provided in brackets.

Methods	**# of Features**	**Prediction Error(%)**
DA-SCAD-SVM	6(0.8)	9.6(0.3)
DA-MCP-SVM	5(0.2)	9.4(0.2)
DA-L0-SVM	5(0.4)	9.6(0.2)
SCAD-SVM	7(0.8)	10.9(0.3)
MCP-SVM	6(0.2)	13.2(0.2)
L_0-norm Approximation SVM	12(1.3)	15.2(0.2)
Adapt L_1-norm SVM	14.50(2.4)	17(1.5)

5. Concluding Remarks

In this paper, we propose a data-adaptive kernel-penalized SVM, a new method that simultaneously achieves feature selection and classification, especially when the data is imbalanced. Instead of penalizing the loss function of SVMs, as has been done in the literature, a non-convex penalty is proposed to be added directly to the kernel form of the SVM. The benefit is that the features are selected more correctly in the feature space instead of the original input space. This is because it is the kernel function that mainly determines the classification process. Moreover, the data-adaptive kernel is applicable to SVM so that, even when the data is imbalanced, the performance of the SVM is still excellent, while—in this setting—other penalized SVM cannot work well due to the lack of flexibility in SVM. Along with the oracle properties, if the true sparsity in the feature space is already known, our proposed method works well in both the simulation study and the real data example, possibly even when the ultra-dimensional setting exists.

The method proposed in this paper is actually an embedded approach, as mentioned in the introduction part, and the forms of penalty terms are not limited to those applied in the methodology above. In terms of the multi-category classification problem, the methodology can be extended to fit in the direct method, though the data-adaptive kernels need to be modified. Another issue is the choice of the primary kernel function. The methodology proposed is base on the Gaussian RBF kernel due to its natural link with the contribution of the predictors. Extensions will be considered in future works.

Furthermore, in terms of the cancer image dataset, the patients included in the study probably have more diseases in the prostate other than cancer, and this requires techniques for multi-category classifiers. Moreover, measurement errors will probably exist due to the co-registration of the measures

from different platforms, and this may affect the accuracy of the classifier. Future works will continue this study, taking all of these issues into consideration.

Author Contributions: Conceptualization, X.L. and W.H.; Data curation, B.Z.; Formal analysis, X.L.; Funding acquisition, W.H.; Methodology, X.L.; Project administration, W.H.; Software, B.Z.; Supervision, W.H.; Validation, X.L. and B.Z.; Writing of original draft, X.L.; Writing of review & editing, B.Z. and W.H. All authors have read and agreed to the published version of the manuscript.

Funding: Liu's research is funded by the Fundamental Research Funds for the Central Universities (number 2018110185). He's research is funded by a grant from the Natural Sciences and Engineering Research Council of Canada and a team grant from the Canadian Institute of Health Research (CIHR).

Acknowledgments: The authors thank the review team for their helpful comments on the initial submission. The authors thank the CIHR Team in Image-Guided Prostate Cancer Management at the University of Western Ontario.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Proof of Lemmas and Theorems

Appendix A.1. Proof of Lemma 1

Proof. By the definition of a reproducing kernel function $K(\mathbf{x}, \mathbf{z})$ with its values λ_k and the corresponding scalar eigenfunctions $\mathrm{g}_k(\mathbf{x})$, we have

$$\int K(\mathbf{x}, \mathbf{z}) \cdot \mathrm{g}_k(\mathbf{z})\, \mathrm{d}\mathbf{z} \;=\; \lambda_k \cdot \mathrm{g}_k(\mathbf{x})$$

where $k = 1, 2, \ldots, l$. Then, the kernel is represented as

$$K(\mathbf{x}, \mathbf{z}) \;=\; \sum_k \lambda_k \cdot \mathrm{g}_k(\mathbf{x}) \cdot \mathrm{g}_k(\mathbf{z}).$$

By rescaling the function $g_k(\cdot)$ as $s_k(\mathbf{x}) = \sqrt{\lambda_k} g_k(\mathbf{x})$, the kernel function can be further presented as

$$K(\mathbf{x}, \mathbf{z}) \;=\; \sum_k \mathrm{s}_k(\mathbf{x}) \cdot \mathrm{s}_k(\mathbf{z}) \;=\; [\mathbf{s}(\mathbf{x})]^T \cdot [\mathbf{s}(\mathbf{z})]$$

where $[\mathbf{s}(\mathbf{x})]^T = (\mathrm{s}_1(\mathbf{x}), \mathrm{s}_2(\mathbf{x}), \ldots, \mathrm{s}_l(\mathbf{x}))$ and $[\cdot]^T$ is the transpose operator. Thus, if we further define

$$\nabla \mathbf{s} \;=\; \left(\frac{\partial\, s(\mathbf{x})}{\partial\, \mathbf{x}}\right) = \begin{pmatrix} \frac{\partial\, s_1(\mathbf{x})}{\partial\, x_1} & \cdots & \frac{\partial\, s_1(\mathbf{x})}{\partial\, x_p} \\ \vdots & \vdots & \vdots \\ \frac{\partial\, s_l(\mathbf{x})}{\partial\, x_1} & \cdots & \frac{\partial\, s_l(\mathbf{x})}{\partial\, x_p} \end{pmatrix}$$

and

$$\begin{aligned} \mathrm{s}_{ij}(\mathbf{x}) \;&=\; \left(\frac{\partial}{\partial x_i}\mathbf{s}(\mathbf{x})\right)^T \cdot \left(\frac{\partial}{\partial x_j}\mathbf{s}(\mathbf{x})\right) \\ &=\; \left(\frac{\partial \mathrm{s}_1(\mathbf{x})}{\partial x_i}, \ldots, \frac{\partial \mathrm{s}_l(\mathbf{x})}{\partial x_i}\right) \cdot \left(\frac{\partial \mathrm{s}_1(\mathbf{x})}{\partial x_j}, \ldots, \frac{\partial \mathrm{s}_l(\mathbf{x})}{\partial x_j}\right)^T, \end{aligned}$$

as in (7) and (8), it follows that

$$\frac{\partial}{\partial x_i}\frac{\partial}{\partial z_j} K(\mathbf{x}, \mathbf{z})|_{\mathbf{z}=\mathbf{x}} = [\nabla \mathbf{s}(\mathbf{x})]^T \cdot \nabla \mathbf{s}(\mathbf{z}) = \left(\frac{\partial}{\partial x_i}\mathbf{s}(\mathbf{x})\right)^T \cdot \left(\frac{\partial}{\partial x_j}\mathbf{s}(\mathbf{x})\right) = \mathrm{s}_{ij}(\mathbf{x}).\ \sharp$$

□

The lemma shows how mapping **s** is associated with the corresponding kernel function K. Thus, given a specific form of a kernel function and an adaptive scaling function $c(\mathbf{x})$, we have Theorems 1 and 2.

Appendix A.2. Proof of Theorem 1

Proof. When we apply, in Theorem 1, the Gaussian RBF kernel as in (12), it is found that

$$K_{i\cdot}(\mathbf{x},\mathbf{x}) = K_{\cdot j}(\mathbf{x},\mathbf{x}) = 0$$

and

$$K(\mathbf{x},\mathbf{x}) = 1$$

for any i and j, so the third term in the result of Theorem 1 is zero, and the second term is changed into $c_i(\mathbf{x}) \cdot c_j(\mathbf{x})$. Furthermore, when $i \neq j$,

$$\mathrm{s}_{ij}(\mathbf{x}) = \frac{\partial}{\partial x_i}\frac{\partial}{\partial z_j}K(\mathbf{x},\mathbf{z})|_{\mathbf{z}=\mathbf{x}} = \frac{1}{\sigma^2}(x_i - z_i)\cdot K(\mathbf{x},\mathbf{x})\cdot(x_j - z_j)\Big|_{\mathbf{z}=\mathbf{x}} = 0,$$

while, when $i = j$,

$$\mathrm{s}_{ii}(\mathbf{x}) = \frac{1}{\sigma^2}\Big((x_i - z_i)\cdot K(\mathbf{x},\mathbf{z})\cdot(x_i - z_i) + K(\mathbf{x},\mathbf{z})\Big)\Big|_{\mathbf{z}=\mathbf{x}} = \frac{1}{\sigma^2};$$

thus, the first term becomes

$$\frac{c^2(\mathbf{x})}{\sigma^2}\cdot(i=j).$$

Combining all the above results, Theorem 1 is proved. □

Appendix A.3. Proof of Theorem 2: The Oracle Properties in Data-Adaptive Kernel-Penalized SVM

Proof. Define

$$L(\boldsymbol{\beta}) = \sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i,\mathbf{x}_j;\boldsymbol{\beta}) + \sum_{j=1}^{p} p_{\lambda_n}(|\beta_j|) \;=\; L_1(\boldsymbol{\beta}) + \sum_{j=1}^{p} p_{\lambda_n}(|\beta_j|), \tag{A1}$$

which comes from the second part of the optimization problem in (22). We shall show that, for $\forall\,\epsilon > 0$, there is a constant Δ, such that, when n is sufficiently large,

$$\mathbf{Pr}[\inf_{\|\mathbf{u}\|=\Delta} L(\boldsymbol{\beta}_{true} + \sqrt{q/n}\cdot\mathbf{u}) > L(\boldsymbol{\beta}_{true})] \;\geq\; 1-\epsilon. \tag{A2}$$

In the following proof, $\boldsymbol{\beta}_{true}$ will be replaced by $\boldsymbol{\beta}$ for short, without misleading the proof. Note that $\sum_{i=1}\alpha_i y_i = 0$ from the constraints of (21).

$$\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j = \sum_{i=1}^{l}\alpha_i y_i \cdot \sum_{j=1}^{l}\alpha_j y_j = 0, \tag{A3}$$

and, furthermore,

$$0 = \sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j = \sum_{i,\,j,\,y_i y_j=1}\alpha_i\alpha_j - \sum_{i,\,j,\,y_i y_j=-1}\alpha_i\alpha_j. \tag{A4}$$

This immediately leads to

$$L_1(\boldsymbol{\beta}) \;=\; \sum_{i,\,j,\,y_i y_j=1}\alpha_i\alpha_j K(\mathbf{x}_i,\mathbf{x}_j;\boldsymbol{\beta}) \;-\; \sum_{i,\,j,\,y_i y_j=-1}\alpha_i\alpha_j K(\mathbf{x}_i,\mathbf{x}_j;\boldsymbol{\beta}) \tag{A5}$$

Since $y_i \in \{1,-1\}$, then $y_i y_j \in \{1,-1\}$ for all (i,j), with a probability of $\pi_+^2 + \pi_-^2$ for 1 and $2\pi_+\pi_1$ for -1, assuming independence between y_i and y_j, where $\pi_+ = \Pr(y_i = 1)$ and $\pi_- = \Pr(y_i = 1)$; furthermore, it is easy to check

$$0 \le E(y_i y_j) = \pi_+^2 + \pi_-^2 - 2\pi_+\pi_1 = (\pi_+ - \pi_-)^2 \le 1$$

and thus

$$E(L_1(\boldsymbol{\beta})) \;=\; (\pi_+ - \pi_-)^2 \sum_{i=1}^{l}\sum_{j=1}^{l} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j; \boldsymbol{\beta}) \ge 0. \tag{A6}$$

Now, let

$$\begin{aligned}
\Lambda_n(\mathbf{u}) =\; & nq^{-1} \cdot [L_1(\boldsymbol{\beta} + \sqrt{q/n} \cdot \mathbf{u}) - L_1(\boldsymbol{\beta})] \\
=\; & nq^{-1} \sum_{i=1}^{l}\sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j; \boldsymbol{\beta}) \cdot \exp\{-\frac{1}{2} q/n \cdot [(\mathbf{x}_i - \mathbf{x}_j)^{\odot 2}]^T \mathbf{u} - 1\} \\
=\; & nq^{-1} \cdot \sum_{i,\, j,\, y_i \cdot y_j = 1}^{l} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j; \boldsymbol{\beta}) \cdot \exp\{-\frac{1}{2} q/n \cdot [(\mathbf{x}_i - \mathbf{x}_j)^{\odot 2}]^T \mathbf{u} - 1\} \\
& - nq^{-1} \cdot \sum_{i,\, j,\, y_i \cdot y_j = -1}^{l} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j; \boldsymbol{\beta}) \cdot \exp\{-\frac{1}{2} \sqrt{q/n} \cdot [(\mathbf{x}_i - \mathbf{x}_j)^{\odot 2}]^T \mathbf{u} - 1\}
\end{aligned} \tag{A7}$$

where $()^{\odot 2}$ is the component-wise square. Since $\exp(x) > x + 1$ for all x and $\alpha_i \ge 0$ for all i, then the first item in (A7) is

$$\begin{aligned}
& \ge nq^{-1} \sum_{i,\, j,\, y_i \cdot y_j = 1}^{l} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j; \boldsymbol{\beta}) \cdot [-\frac{1}{2} \sqrt{q/n} \cdot [(\mathbf{x}_i - \mathbf{x}_j)^{\odot 2}]^T \mathbf{u} - 1 + 1] \\
& = \sqrt{nq^{-1}} \sum_{i,\, j,\, y_i \cdot y_j = 1}^{l} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j; \boldsymbol{\beta}) \cdot \{-\frac{1}{2} \cdot [(\mathbf{x}_i - \mathbf{x}_j)^{\odot 2}]^T \mathbf{u}\}.
\end{aligned} \tag{A8}$$

Taking the standard augmentation of the Taylor expansion with respect to $\mathbf{u}$,

$$\begin{aligned}
\exp\{-\frac{1}{2} \cdot [(\mathbf{x}_i - \mathbf{x}_j)^{\odot 2}]^T \mathbf{u} - 1\} = & -\frac{1}{2} \sqrt{q/n} \cdot [(\mathbf{x}_i - \mathbf{x}_j)^{\odot 2}]^T \mathbf{u} \\
& + \frac{1}{4} \cdot \frac{q}{n} \cdot \mathbf{u}^T [(\mathbf{x}_i - \mathbf{x}_j)^{\odot 2}][(\mathbf{x}_i - \mathbf{x}_j)^{\odot 2}]^T \mathbf{u} + o_p(n^{-1}).
\end{aligned} \tag{A9}$$

Then it is easy to find that the second item in (A7) is

$$\begin{aligned}
\le nq^{-1} \cdot \sum_{i,\, j,\, y_i \cdot y_j = -1}^{l} & \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j; \boldsymbol{\beta}) \cdot (-\frac{1}{2} \sqrt{q/n} \cdot [(\mathbf{x}_i - \mathbf{x}_j)^{\odot 2}]^T \mathbf{u} \\
& + \frac{1}{4} \cdot \frac{q}{n} \cdot \mathbf{u}^T [(\mathbf{x}_i - \mathbf{x}_j)^{\odot 2}][(\mathbf{x}_i - \mathbf{x}_j)^{\odot 2}]^T \mathbf{u} + o_p(1)
\end{aligned} \tag{A10}$$

Now, by combining (A8) and (A10), we have

$$\begin{aligned}\Lambda_n(\mathbf{u}) \geq & \sqrt{nq^{-1}} \cdot [\sum_{i,\, j,\, y_i \cdot y_j = 1}^{l} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j; \boldsymbol{\beta}) \cdot \{-\frac{1}{2} \cdot [(\mathbf{x}_i - \mathbf{x}_j)^{\odot 2}]^T \mathbf{u} \\ & - \sum_{i,\, j,\, y_i \cdot y_j = -1}^{l} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j; \boldsymbol{\beta}) \cdot \{-\frac{1}{2} \cdot [(\mathbf{x}_i - \mathbf{x}_j)^{\odot 2}]^T \mathbf{u}\}] \\ & + \frac{1}{4} \cdot \mathbf{u}^T [(\mathbf{x}_i - \mathbf{x}_j)^{\odot 2}][(\mathbf{x}_i - \mathbf{x}_j)^{\odot 2}]^T \mathbf{u} + o_p(1) \\ = & \sqrt{nq^{-1}} \cdot \sum_{i,j}^{l} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j; \boldsymbol{\beta}) \cdot \{-\frac{1}{2} \cdot [(\mathbf{x}_i - \mathbf{x}_j)^{\odot 2}]^T\} \cdot \mathbf{u} \\ & + \frac{1}{4} \sum_{i,\, j,\, y_i \cdot y_j = -1}^{l} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j; \boldsymbol{\beta}) \cdot \mathbf{u}^T [(\mathbf{x}_i - \mathbf{x}_j)^{\odot 2}][(\mathbf{x}_i - \mathbf{x}_j)^{\odot 2}]^T \mathbf{u} + o_p(1)\end{aligned} \tag{A11}$$

Note that the first part in (A11) is equivalent to $\frac{\partial}{\partial \boldsymbol{\beta}} L_1'(\boldsymbol{\beta}) = 0$ due to the necessary condition that $\boldsymbol{\beta} = \arg\min L_1(\boldsymbol{\beta})$, and the second term, which is obviously non-negative, will dominate (A11). In terms of the penalty term, it is obvious that

$$\begin{aligned}P_n(\boldsymbol{\beta}) &= nq^{-1} \sum_{j=1}^{p} [p_{\lambda_n}(|\beta_j + \sqrt{q/n} \cdot u_j|) - p_{\lambda_n}(|\beta_j|)]; \quad \text{using } p_{\lambda_n}(0) = 0 \text{ and } p_{\lambda_n}(\cdot) \geq 0 \\ &\geq \sum_{j=1}^{q} nq^{-1} \cdot [p_{\lambda_n}(|\beta_j + \sqrt{q/n} \cdot u_j|) - p_{\lambda_n}(|\beta_j|)]; \quad \text{using Taylor Expansion} \\ &= \sum_{j=1}^{q} \cdot [q^{-1/2} p'_{\lambda_n}(|\beta_j|) + p''_{\lambda_n}(|\beta_j|) u_j^2 \{1 + o_p(1)\}],\end{aligned} \tag{A12}$$

which is bounded by $q^{-1/2}\|\mathbf{u}\| + \max\{|p''_{\lambda_n}(\beta_j)| : \beta_j \neq 0\}\|\mathbf{u}|$. Thus, by choosing a sufficiently large Δ, $P_n(\boldsymbol{\beta})$ is dominated by the second item in (A11) as well. Thus, $L(\boldsymbol{\beta}) = \Lambda_n(\mathbf{u}) + P_n(\boldsymbol{\beta})$ is dominated by a non-negative item with probability 1 within a ball. This indicates that with a probability of at least $1 - \epsilon$, there exists a local minimum in the ball $\{\boldsymbol{\beta} + \sqrt{q/n} \cdot \mathbf{u} : \|\mathbf{u}\| \leq \Delta\}$, and hence there exists a local minimizer, such that $\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\| = O_p\{\sqrt{q/n}\}$. Note that when the kernel function K is updated by $\tilde{K}$, nothing is changed except that the kernel is multiplied by two finite constants constructed from the first stage of SVM, and hence the theorem still holds. This completes the proof. □

References

1. Blum, A.L.; Langley, P. Selection of relevant features and examples in machine learning. *Artif. Intell.* **1997**, *97*, 245–271. [CrossRef]
2. Zhang, L.; Hu, H.; Zhang, D. A credit risk assessment model based on SVM for small and medium enterprises in supply chain finance. *Financ. Innov.* **2015**, *1*, 14. [CrossRef]
3. Khokhar, S.; Zin, A.A.B.M.; Mokhtar, A.S.B.; Pesaran, M. A comprehensive overview on signal processing and artificial intelligence techniques applications in classification of power quality disturbances. *Renew. Sustain. Energy Rev.* **2015**, *51*, 1650–1663. [CrossRef]
4. Vapnik, V.N.; Vapnik, V. *Statistical Learning Theory*; Wiley: New York, NY, USA, 1998; Volume 1.
5. Rodger, J.A. Discovery of medical Big Data analytics: Improving the prediction of traumatic brain injury survival rates by data mining Patient Informatics Processing Software Hybrid Hadoop Hive. *Inform. Med. Unlocked* **2015**, *1*, 17–26. [CrossRef]
6. Maldonado, S.; Weber, R. A wrapper method for feature selection using support vector machines. *Inf. Sci.* **2009**, *179*, 2208–2217. [CrossRef]

7. Friedman, J.; Hastie, T.; Tibshirani, R. *The Elements of Statistical Learning*; Springer: Berlin, Germany, 2001; Volume 1.
8. Zhang, X.; Wu, Y.; Wang, L.; Li, R. Variable selection for support vector machines in moderately high dimensions. *J. R. Stat. Soc. Ser. B (Stat. Methodol.)* **2016**, *78*, 53–76. [CrossRef]
9. Guyon, I.; Weston, J.; Barnhill, S.; Vapnik, V. Gene selection for cancer classification using support vector machines. *Mach. Learn.* **2002**, *46*, 389–422. [CrossRef]
10. Zou, H. An Improved 1-norm SVM for Simultaneous Classification and Variable Selection. *AISTATS* **2007**, *2*, 675–681.
11. Maldonado, S.; Weber, R.; Basak, J. Simultaneous feature selection and classification using kernel-penalized support vector machines. *Inf. Sci.* **2011**, *181*, 115–128. [CrossRef]
12. Pehro, D.; Stork, D. *Pattern Classification*; Wiley Interscience Publication: Hoboken, NJ, USA, 2001.
13. Guyon, I.; Elisseeff, A. An introduction to variable and feature selection. *J. Mach. Learn. Res.* **2003**, *3*, 1157–1182.
14. Fan, J.; Li, R. Variable selection via nonconcave penalized likelihood and its oracle properties. *J. Am. Stat. Assoc.* **2001**, *96*, 1348–1360. [CrossRef]
15. Bradley, P.S.; Mangasarian, O.L. Feature selection via concave minimization and support vector machines. *ICML* **1998**, *98*, 82–90.
16. Fumera, G.; Roli, F. Support vector machines with embedded reject option. In *Pattern Recognition with Support Vector Machines*; Springer: New York, NY, USA, 2002; pp. 68–82.
17. Zhu, J.; Rosset, S.; Hastie, T.; Tibshirani, R. 1-norm Support Vector Machines. *NIPS* **2003**, *15*, 49–56.
18. Wang, L.; Zhu, J.; Zou, H. The doubly regularized support vector machine. *Stat. Sin.* **2006**, *12*, 589–615.
19. Wang, L.; Zhu, J.; Zou, H. Hybrid huberized support vector machines for microarray classification and gene selection. *Bioinformatics* **2008**, *24*, 412–419. [CrossRef] [PubMed]
20. Zou, H.; Yuan, M. The $F_\infty-$norm support vector machine. *Stat. Sin.* **2008**, *18*, 379–398.
21. Park, C.; Kim, K.R.; Myung, R.; Koo, J.Y. Oracle properties of scad-penalized support vector machine. *J. Stat. Plan. Inference* **2012**, *142*, 2257–2270. [CrossRef]
22. Wu, G.; Chang, E.Y. Adaptive feature-space conformal transformation for imbalanced-data learning. In Proceedings of the 20th International Conference on Machine Learning (ICML-03), Washington, DC, USA, 21–24 August 2003; pp. 816–823.
23. Williams, P.; Li, S.; Feng, J.; Wu, S. Scaling the kernel function to improve performance of the support vector machine. In *Advances in Neural Networks–ISNN 2005*; Springer: Cham, Switzerland, 2005; pp. 831–836.
24. Maratea, A.; Petrosino, A.; Manzo, M. Adjusted F-measure and kernel scaling for imbalanced data learning. *Inf. Sci.* **2014**, *257*, 331–341. [CrossRef]
25. Boser, B.E.; Guyon, I.M.; Vapnik, V.N. A training algorithm for optimal margin classifiers. In Proceedings of the Fifth Annual Workshop on Computational Learning Theory, Pittsburgh, PA, USA, 27–29 July 1992; pp. 144–152.
26. Amari, S.i.; Wu, S. Improving support vector machine classifiers by modifying kernel functions. *Neural Netw.* **1999**, *12*, 783–789. [CrossRef]
27. Lin, Y. Support vector machines and the Bayes rule in classification. *Data Min. Knowl. Discov.* **2002**, *6*, 259–275. [CrossRef]
28. Zhang, C.H. Nearly unbiased variable selection under minimax concave penalty. *Ann. Stat.* **2010**, *38*, 894–942. [CrossRef]
29. Wu, S.; Amari, S.I. Conformal transformation of kernel functions: A data-dependent way to improve support vector machine classifiers. *Neural Process. Lett.* **2002**, *15*, 59–67. [CrossRef]
30. Zhu, J.; Rosset, S.; Tibshirani, R.; Hastie, T.J. 1-norm support vector machines. In *Advances in Neural Information Processing Systems*; The MIT Press: New York, NY, USA, 2004; pp. 49–56.
31. Mazumder, R.; Friedman, J.H.; Hastie, T. Sparsenet: Coordinate descent with nonconvex penalties. *J. Am. Stat. Assoc.* **2011**, *106*, 1125–1138. [CrossRef] [PubMed]
32. Chen, J.; Chen, Z. Extended Bayesian information criteria for model selection with large model spaces. *Biometrika*, **2008**, *95*, pp. 759–771. [CrossRef]
33. Claeskens, G.; Croux, C.; Kerckhoven, J.V. An information criterion for variable selection in support vector machines. *J. Mach. Learn. Res.* **2008**, *9*, 541–558.

[CrossRef]

34. Blake, C.L.; Merz, C.J. *UCI Repository of Machine Learning Databases*; Department information Computer Science, University of California: Irvine, CA, USA, 1998; Volume 55.
35. Mangasarian, O.L.; Street, W.N.; Wolberg, W.H. Breast cancer diagnosis and prognosis via linear programming. *Oper. Res.* **1995**, *43*, 570–577. [CrossRef]