NOVEL WRAPPER APPROACH FOR VARIABLE SELECTION IN SUPPORT VECTOR MACHINE

Koda, Satoru
Graduate School of Mathematics, Kyushu University

Nishii, Ryuei
Institute of Mathematics for Industry, Kyushu University

Fukasawa, Yuta
Graduate School of Mathematical Science, The University of Tokyo

https://doi.org/10.5109/2203031

出版情報：Bulletin of informatics and cybernetics. 48, pp.105-117, 2016-12. 統計科学研究会
バージョン：
権利関係：

Bulletin of Informatics and Cybernetics, Vol. 48, 2016

NOVEL WRAPPER APPROACH FOR VARIABLE SELECTION IN SUPPORT VECTOR MACHINE

By

Satoru Koda[*], **Ryuei Nishii**[†] and **Yuta Fukasawa**[‡]

Abstract

Support vector machine (SVM) is an efficient machine learning method for classification. In this paper, we propose two variable selection criteria for SVM that use wrapper methods. The criteria measure the contribution of each variable for a target function. The variable importance is quantified on the basis of the measured amount. The methods have high computational efficiency because they evaluate the importance of all variables without recursive calculations. They were applied to several artificial and real-world data sets, and their results were superior to those of existing methods.

Key Words and Phrases: Binary classification, Feature selection, Kernel functions

1. Introduction

Support vector machine (SVM) is one of the most powerful classifiers, and many researchers have proposed modifications, developed the mathematical behind them, and applied them to problems in various fields [Wang (2005)], including medical science [Furey et al. (2000)] and economics [Shin et al. (2005)]. In the context of classification, SVM maps samples to a high dimensional space, called a feature space, in which linear classifications are conducted. Thanks to this mapping operation, it can handle nonlinear classifications in the sample space. SVM has other good properties, in particular, efficient parameter estimation and the capability of using the kernel trick, which is a way of replacing inner products in feature space with kernel functions.

Variable selection is an important issue in pattern recognition in the following points. First, it helps to reduce the risk of overfitting and improves prediction accuracy. Second, it can be a helpful way of clarifying causal relationships between input variables and class labels in real-world data analysis. Third, it reduces computational costs by eliminating unnecessary variables.

Variable selection methods in SVM are categorized into filter, embedded, and wrapper [George et al. (2001)], which will be briefly described in section 2. In this paper, we will deal with the wrapper method and propose new variable selection criteria for it. First, we define a target function related to the classification ability of the model. Then, we evaluate the effect that each variable exerts on the function in two ways. Finally, we quantify the importance of the variables based on the evaluated effects. The proposed

[*] Graduate School of Mathematics, Kyushu University, Japan. ma214020@math.kyushu-u.ac.jp

[†] Institute of Mathematics for Industry, Kyushu University, Japan. nishii@imi.kyushu-u.ac.jp

[‡] Graduate School of Mathematical Science, The University of Tokyo, Japan. fukasawa@ms.u-tokyo.ac.jp

methods can evaluate a variable's importance by using only one optimization process. Therefore, they speed up the variable selection procedure. They were applied to datasets from the MLC++ [Kohavi et al. (1994)], UCI [Lichman (2013)] and LIBSVM [Chang et al. (2001)] databases to ascertain their ability. It turned out that they outperformed other methods in almost all cases.

The remainder of this paper is organized as follows. In section 2, we briefly explain the algorithm of SVM classification analysis and review the conventional variable selection strategies. The new variable selection methods are described in section 3. Section 4 shows the results of applying them to benchmark datasets for classification. Section 5 summarizes the paper.

2. Support Vector Machine

In this section, we briefly explain the SVM algorithm and the notation used in the paper. We also review the variable selection methods in the literature.

2.1. Brief Review of SVM

Suppose we want to classify a sample $\boldsymbol{x} = (x_1, \ldots, x_d)^{\mathrm{T}} \in \mathbb{R}^d$ into one of two classes C_1 or C_{-1}. The classification function used in SVM takes the following linear form:

$$f(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}) + b \tag{1}$$

where $\boldsymbol{\phi}(\boldsymbol{x}) \in \mathbb{R}^p$ denotes a fixed feature-space transformation called a feature map. The parameters $\boldsymbol{w}$ and b are to be optimized. The sample $\boldsymbol{x}$ is assigned to one of the classes according to the sign of $f(\boldsymbol{x})$. Here, a kernel function is defined by

$$K(\boldsymbol{x}, \widetilde{\boldsymbol{x}}) = \boldsymbol{\phi}(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{\phi}(\widetilde{\boldsymbol{x}}), \quad \boldsymbol{x},\ \widetilde{\boldsymbol{x}} \in \mathbb{R}^d \tag{2}$$

which calculates an inner product in the feature space. The following kernel functions are frequently used in SVM.

- Linear: $K_L(\boldsymbol{x}, \widetilde{\boldsymbol{x}}) = \boldsymbol{x}^{\mathrm{T}}\widetilde{\boldsymbol{x}}$
- Polynomial: $K_P(\boldsymbol{x}, \widetilde{\boldsymbol{x}}) = (\boldsymbol{x}^{\mathrm{T}}\widetilde{\boldsymbol{x}} + r)^k$, $r > 0,\ k \in \mathbb{N}$
- Gaussian: $K_G(\boldsymbol{x}, \widetilde{\boldsymbol{x}}) = \exp(-\gamma\|\boldsymbol{x} - \widetilde{\boldsymbol{x}}\|^2)$, $\gamma > 0$
- Sigmoid: $K_S(\boldsymbol{x}, \widetilde{\boldsymbol{x}}) = \tanh(\gamma\boldsymbol{x}^{\mathrm{T}}\widetilde{\boldsymbol{x}} + r)$, $\gamma > 0,\ r > 0$.

Now, let us suppose that training data $(\boldsymbol{x}_i, y_i)$ in $\mathbb{R}^d \times \{\pm 1\}$ are given for $i = 1, \ldots, n$, where y_i denotes the class label. The decision boundary which separates the sample space is determined by maximizing a geometric margin in feature space under constraints. The margin means the distance between the boundary hyperplane and any of the samples. The optimization formula is given as

$$\min_{\boldsymbol{w}, \boldsymbol{\xi}} \left\{ \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n} \xi_i \right\}, \quad \boldsymbol{\xi} = (\xi_1, \ldots, \xi_n)^{\mathrm{T}} \tag{3}$$

subject to

$$y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_i) + b) \geq 1 - \xi_i \quad \text{with} \quad \xi_i \geq 0 \quad \text{for} \quad i = 1, \ldots, n.$$

Here, $\|\boldsymbol{w}\|^2$ is the inverse margin, and ξ_i is a slack variable giving a penalty to missclassification. A positive constant C, called a cost parameter, represents the tradeoff between classification accuracy and the complexity of the decision boundary. This optimization problem can be solved using the Lagrange multiplier method. In solving its dual problem, the optimal parameter $\boldsymbol{w}$ is given by

$$\boldsymbol{w} = \sum_{i=1}^{n} a_i y_i \boldsymbol{\phi}(\boldsymbol{x}_i), \tag{4}$$

where a_i $(0 \leq a_i \leq C)$ is the Lagrange multiplier obtained in the optimization process. Consequently, the classification function is derived as

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} a_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b \tag{5}$$

where the intercept b is estimated using support vectors with $a_i > 0$.

2.2. Three Groups of Variable Selection Methods

The classification methods of SVM are divided into three types: filter, embedded, and wrapper [George et al. (2001)]. We will describe them below.

Filter methods measure each variable importance individually. They utilize univariate metrics like correlation coefficients [Hall (1999)], F score [Polat et al. (2009)], mutual information [Ding et al. (2005)] and so on. This enables them to be used for not only classification analysis but also regression. The main drawback is that their variable selection ability is less accurate compared with the other two types. Hence, they are typically used in a preliminary variable screening stage.

Embedded methods simultaneously conduct variable selection and parameter optimization. Methods with the l_1 norm penalty [Bradley et al. (1998)] may be the most well-known methods of this type. Some elements of the parameter vector $\boldsymbol{w}$ would be optimized to be exactly zero, which implies that the corresponding variables are not important for classification.

Typical wrapper methods use a target function derived from the SVM training such as $||\boldsymbol{w}||^2$. The target functions do not normally have any clear aspects for prediction, but there are methods like the theoretical leave-one-out cross validation error bound [Vapnik et al. (2000)] that keep these aspects. The main concern of wrapper methods is to find a target function that selects the best variable subset for prediction. The genetic algorithm is an example of such a search method [Yan et al. (1998)]; it generates candidate variable subsets as comprehensively as possible; then it chooses the best model from the candidates according to criteria. This method is, however, computationally very expensive.

A representative wrapper type method, the SVM-RFE (recursive feature elimination) algorithm [Guyon et al. (2002)], makes use of the margin $||\boldsymbol{w}||^2$ as the target function and searches for the best variable subset by using backward elimination. To illustrate it, suppose that $\boldsymbol{x}_i^{(-k)}$ denotes a sample vector eliminating the k-th variable x_{ik}, $\boldsymbol{w}^{(-k)}$ denotes an estimated SVM parameter based on a training set $\left\{\left(\boldsymbol{x}_i^{(-k)}, y_i\right) \mid i = 1, ..., n\right\}$, and a_i^* is the optimized Lagrange multiplier for the training samples in $\mathbb{R}^{d-1}$.

squared norms of the weight vectors $\boldsymbol{w}$ and $\boldsymbol{w}^{(-k)}$ are derived as

$$\begin{aligned} \|\boldsymbol{w}\|^2 &= \sum_{i,j} a_i a_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j), && (6) \\ \|\boldsymbol{w}^{(-k)}\|^2 &= \sum_{i,j} a_i^* a_j^* y_i y_j K\Big(\boldsymbol{x}_i^{(-k)}, \boldsymbol{x}_j^{(-k)}\Big), && (7) \end{aligned}$$

where $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $K\Big(\boldsymbol{x}_i^{(-k)}, \boldsymbol{x}_j^{(-k)}\Big)$ are kernel functions on $\mathbb{R}^d \times \mathbb{R}^d$ and $\mathbb{R}^{d-1} \times \mathbb{R}^{d-1}$. We denote them using the same notation. The SVM-RFE algorithm uses the margin as follows:

$$\big|\ \|\boldsymbol{w}\|^2 - \|\boldsymbol{w}^{(-k)}\|^2\ \big|,\ \ k = 1, \ldots, d. \tag{8}$$

If Eq. 8 takes a small value, the variable x_k should be removed. In actual analysis, a_i^* is substituted with a_i to reduce computational costs. Another wrapper method [Byvatov et al. (2004)] quantifies the variation caused by variable x_k through differentiation:

$$\sum_{\boldsymbol{x}_j \in SV} \left(\sum_{\boldsymbol{x}_i \in SV} a_i y_i \frac{\partial K(\boldsymbol{x}, \boldsymbol{x}_j)}{\partial x_k}\Big|_{\boldsymbol{x}=\boldsymbol{x}_i} + b \right) \Bigg/ \left(\sum_{\substack{\boldsymbol{x}_i \in SV \\ 1 \le l \le d}} a_i y_i \frac{\partial K(\boldsymbol{x}, \boldsymbol{x}_j)}{\partial x_l}\Big|_{\boldsymbol{x}=\boldsymbol{x}_i} + b \right), \tag{9}$$

where $SV = \{\boldsymbol{x}_i \mid a_i > 0, i = 1, \ldots, n\}$ is the set of support vectors. Variables with small values of Eq. 9 are removed from the model. This method regards the classifier itself as the target function and differentiates it w.r.t each variable.

The RFE-type algorithms select variables based on recursive feature elimination processes. On the other hand, the proposed method evaluates every variable importance with the estimated parameters obtained at the first training phase. The method without iterative steps makes variable selection simpler.

3. Proposed Classifiers

We propose two variable selection criteria for SVM classification. In what follows, we use data which have already been normalized in terms of their mean and standard deviation.

First, we define the target function. Since samples are classified into two classes according to the sign of the function $f(\boldsymbol{x})$ of Eq. 1, correctly classified samples satisfy the inequality $y_i f(\boldsymbol{x}_i) > 0$. Thus, the sum given by

$$\Delta(f) = \sum_{\boldsymbol{x}_i \in SV} y_i f(\boldsymbol{x}_i) = \sum_{\boldsymbol{x}_i \in SV} y_i \left\{ \sum_{\boldsymbol{x}_i \in SV} a_j y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) \right\} \tag{10}$$

increases when the decision boundary divides the sample space appropriately. Therefore, Eq. 10 is a target function to maximize. The next step is to identify the effective variables for the target function. Such variables can be regarded as informative because sensitive variables significantly contribute to the classification. We evaluate the variable sensitivity by using the following two criteria.

Variable elimination criterion: The sensitivity of variable x_k is evaluated by eliminating the variable from the model. Specifically, the criterion is

$$\Delta^{(-k)}(f) = \sum_{\boldsymbol{x}_i \in SV} y_i \left\{ \sum_{\boldsymbol{x}_j \in SV} a_j y_j K\Big(\boldsymbol{x}_i^{(-k)}, \boldsymbol{x}_j^{(-k)}\Big) + b \right\} \quad \text{for} \quad k = 1, ..., d. \tag{11}$$

If $\Delta^{(-k)}(f)$ becomes small compared with $\Delta(f)$, we can consider that classification accuracy deteriorates as a result of removing x_k from the model. Thereby, we can regard x_k as important.

Variable differentiation criterion: This criterion is based on the derivative:

$$D_k\Delta(f) = \sum_{\boldsymbol{x}_i \in SV} y_i \frac{\partial f(\boldsymbol{x})}{\partial x_k}\Big|_{\boldsymbol{x}=\boldsymbol{x}_i} = \sum_{\boldsymbol{x}_i \in SV} y_i \sum_{\boldsymbol{x}_j \in SV} a_j y_j \frac{\partial K(\boldsymbol{x}, \boldsymbol{x}_j)}{\partial x_k}\Big|_{\boldsymbol{x}=\boldsymbol{x}_i}. \tag{12}$$

Variables with large absolute values of $D_k\Delta(f)$ must sensitively affect the target function. Therefore, we can regard such variables as informative. Differentiability of the kernel function is hardly a matter because typical kernels are differentiable.

We derive the features of the two criteria defined above. Moreover, unlike genetic algorithm based methods, they require only one optimization procedure for the SVM parameters $\boldsymbol{w}$ and b. In addition, our criteria assess the variation for the target function instead of a margin like Eq. 8. Also, the method using Eq. 9 differentiates the optimized classification function itself, while our differentiation method takes the derivative of a reasonable target function.

There is a practical concern as to judging how many ranked variables have the best classification ability. In the elimination method, it is clear that the variables are noisy if the values $\Delta(f) - \Delta^{(-k)}(f)$ are negative. Hence, we can simply remove the variables with negative sensitivities. The judgment for the differentiation method is not clear, but we recommend defining a threshold θ and removing the variables whose sensitivities are under the threshold. For instance, θ is set to $0.1 \times |D_{k^*}\Delta(f)|$, where $k^* = \text{argmax}_k |D_k\Delta(f)|$. We will examine these viewpoints in our numerical experiments.

4. Numerical Experiments

We conducted numerical experiments on the proposed methods as well as other wrapper methods, i.e., SVM-RFE using Eq. 8 and differential based criterion (D-SVM) using Eq. 9. F score [Chen et al. (2006)] (a filter method) and L1-regularized SVM [Fan et al. (2008)] (an embedded method) were also compared. Since the L1 SVM explicitly uses a linear kernel, its results describe degrees of nonlinearity for every piece of data. They were applied to benchmark datasets from the MLC++ [Kohavi et al. (1994)], UCI [Lichman (2013)] and LIBSVM [Chang et al. (2001)] data repositories. We used the 'kernlab' package [Karatzoglou et al. (2004)] in R software. The Gaussian kernel was used because it has good properties for constructing a flexible boundary and a simple structure with one parameter γ [Hsu et al. (2003)]. In this case, the proposed elimination criterion function is given by

$$\Delta^{(-k)}(f) = \sum_{\boldsymbol{x}_i \in SV} y_i \left\{ \sum_{\boldsymbol{x}_j \in SV} a_j y_j \exp\Big(-\gamma \|\boldsymbol{x}_i^{(-k)} - \boldsymbol{x}_j^{(-k)}\|^2 \Big) + b \right\}, \tag{13}$$

and the differentiation criterion function is derived as

$$D_k\Delta(f) = -2\gamma \sum_{\boldsymbol{x}_i \in SV} y_i \sum_{\boldsymbol{x}_j \in SV} a_j y_j (x_{ik} - x_{jk}) \exp(-\gamma \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2). \tag{14}$$

To evaluate the classification results, we used the balanced error rate (BER) [Guyon et al. (2004)] given by

$$\text{BER} = \frac{1}{2}(\text{false negative rate} + \text{false positive rate}). \tag{15}$$

The false negative rate means the ratio of incorrectly classified positive samples, and the false positive rate is similarly defined. The average BER is effective if the training data have a large difference in the numbers of positive and negative samples.

The criterion functions were evaluated as follows.

(1) Let each variable be normalized to have zero mean and unit variance. (2) The cost parameter C was determined by 10-fold cross validation. The search space was $\{2^{-4}, 2^{-3}, \ldots, 2^{15}\}$. (3) The scale parameter γ in the Gaussian kernel was determined by 10-fold cross validation. The search space was $\{2^{-15}, 2^{-14}, \ldots, 2^{4}\}$. (4) After training the SVM classifier, the elimination criterion $\Delta^{(-k)}(f)$ defined by Eq. 13 and the differentiation criterion $D_k\Delta(f)$ defined by Eq. 14, as well as the other criteria in Eqs. 8 and 9, were calculated. (5) The variables were sorted in order of their importance.

4.1. Numerical Experiments on Artificial Data

We evaluated the proposed methods on artificial benchmark datasets from the MLC++ database [Kohavi et al. (1994)]. It is known whether each variable of the dataset is informative for classification or not. Therefore, our primary interest was how well the methods select informative variables.

4.1.1. Specifications of Artificial Data for Binary Classification

The proposed classifiers were applied to the following four datasets.

- **Monk1 data set**: The set consists of 124 samples with six categorical variables. Class labels were generated in accordance with the rule: $(x_1 = x_2)$ or $(x_5 = 1)$
- **Monk3 data set**: The set consists of 122 samples with six categorical variables. Class labels were generated in accordance with the rule: $(x_5 = 3$ and $x_4 = 1)$ or $(x_5 \neq 4$ and $x_2 \neq 3)$. Moreover 5% of the labels were reversed randomly.
- **Corral data set**: The set consists of 64 samples with six binary variables. Class labels were generated in accordance with the rule: $(x_1$ and $x_2)$ or $(x_3$ and $x_4)$. The variable x_5 was irrelevant to the class labels, and x_6 was highly correlated with the class labels, but with a 25% error rate.
- **Parity 5+5 data set**: The set consists of 1024 samples with ten binary variables. Class labels were binary sums of x_2, x_3, x_4, x_6 and x_8.

Table 1 outlines the specifications of the datasets. The last column means informative variables for classification.

Table 1: Specifications of Artificial Data Sets

Name	# of variables	Training sample size	Valid variables
Corral	6	64	$\mathbf{1, 2, 3, 4}$
Monk1	6	124	$\mathbf{1, 2, 5}$
Monk3	6	122	$\mathbf{2, 4, 5}$
Parity 5+5	10	1024	$\mathbf{2, 3, 4, 6, 8}$

Table 2: Results for the Artificial Data Sets: Variables Selected with the Proposed and Other Methods

Methods	F score	L1 SVM	D-SVM	SVM-RFE	Elimination	Differentiation
Corral	6, **1**, **2**, **3**, **4**, 5	**1**, **2**, **3**, **4**, 6	**2**, **3**, **1**, **4**, 5, 6	**1**, **3**, **2**, **4**, 5, 6	**1**, **2**, **3**, **4**, 6, 5	**2**, **3**, **4**, **1**, 6, 5
Monk1	**5**, **1**, 4, 3, **2**, 6	**1**, **5**	**5**, 3, 4, 6, **2**, **1**	**1**, **2**, **5**, 6, 4, 3	**5**, **1**, **2**, 6, 3, 4	**5**, **2**, **1**, 3, 4, 6
Monk3	**2**, **5**, 6, 1, 3, **4**	1, **2**, 3, **4**, **5**, 6	6, 3, 1, **4**, **5**, **2**	**2**, **5**, 6, 3, 1, **4**	**2**, **5**, **4**, 1, 3, 6	**5**, **2**, **4**, 3, 1, 6
Parity 5+5	Same F scores	▯	**8**, **4**, **2**, **3**, **6**, 7, 10, 9, 5, 1	**4**, **6**, **3**, **2**, **8**, 9, 10, 7, 5, 1	**2**, **3**, **6**, **4**, **8**, 9, 5, 10, 7, 1	**2**, **3**, **8**, **6**, 10, 9, 5, 1, 7, **4**

The bold numbers are informative variables for class labels.

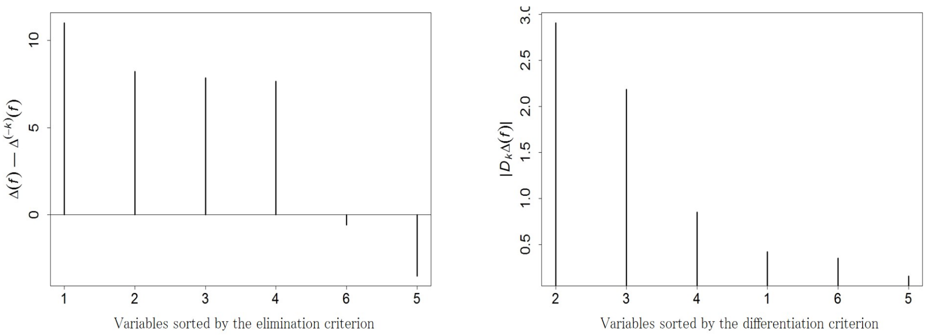

Figure 1: Variable Sensitivity of Corral Data Set: The left figure illustrates the variable sensitivity determined by the elimination criterion. For the sake of simplicity, we calculated $\Delta(f) - \Delta^{(-k)}(f)$ for each variable. The right figure illustrates the variable sensitivity determined by the differentiation criterion. The solid lines mean a boundary between the valid variables and the others.

4.1.2. Evaluation of Classification Results

Table 2 summarizes the classified results of the four datasets. The variables in each cell are ordered by the importance determined from each criterion. The bold numbers are the valid variables. For example, F score gave a preference order of $6 > \mathbf{1} > \mathbf{2} > \mathbf{3} > \mathbf{4} > 5$. Unfortunately, the top-rated 6-th variable is actually invalid. Hence, F score failed to find the valid variables, whereas the remaining classifiers successfully placed all valid variables among the top four. The F score classifier gave the same values for all variables of Parity 5+5. Moreover, L1-SVM failed to find an appropriate subset of Parity 5+5 variables.

Table 2 implies that the proposed elimination method completely succeeded in choosing the valid variables. The proposed differentiation criterion selected all the valid variables for the Corral, Monk1 and Monk3 data, and the four valid variables out of five for the Parity 5+5 data.

Figure 1 shows the variable sensitivity of the proposed methods for the Corral dataset. Compared with the differentiation criterion (right), the sensitivity of the elimination criterion (left) showed a clear difference between the valid variables and the others. Especially in the left figure, the negative sensitivity of variable x_5 and x_6 indicates that they are obstructions interfering with proper classification. This fact suggests that the sensitivities in the elimination method are useful because we can simply remove the variables with negative values of $\Delta(f) - \Delta^{(-k)}(f)$. Actually, the other experiments also showed this simple judgement works well for variable selection as well.

4.2. Numerical Experiments on Real-World Data

The proposed methods were also applied to real-world benchmark data sets from UCI [Lichman (2013)] and LIBSVM [Chang et al. (2001)]. We divided each dataset into three subsets; two subsets were used for training, the other for testing. We devised Algorithm 1 for careful evaluation of the variable importance.

Algorithm 1 Numerical Evaluation on Real-World Data

```
Sort variables according to the importance determined by the proposed methods.
for $k = 1$ to $d$ do
    Pick up the top $k$ variables of the data.
    Tune SVM parameters $\gamma$ and $C$ for the data constituted by the $k$ variables.
    Train an SVM classifier.
    Calculate BER for the test data.
end for
Find the minimum BER.
```

In order to reduce computation costs, we utilized the method [Wu et al. (2009)] for optimizing γ. This method optimizes γ by maximizing the distance between clusters in feature space.

4.2.1. Specifications of Real-World Data

We used the following seven real-world datasets, whose specifications are outlined in Table 3.

- **Australian dataset** [Lichman (2013)]: The set concerns credit card applications. It consists of 690 samples with 14 variables including six numerical and eight categorical variables. The missing values have been already replaced with the medians.
- **Diabetes dataset** [Lichman (2013)]: The set concerns diabetes diagnoses of the Pima Indians who have the highest rate of diabetes in the world. The task is to predict whether patients have diabetes or not from eight variables for 768 patients.
- **QSAR biodegradation dataset** [Lichman (2013)]: The set concerns biodegration, which is the biological decomposition of materials. The purpose is to classify 1055 chemicals into two classes, readily or not readily biodegradable, with 41 molecular descriptor variables.

Table 3: Specifications of Real-World Data Sets

Name	# of variables	Training sample size	Test sample size
Australian	14	460	230
Diabetes	8	512	256
QSAR	41	703	339
Sonar	60	138	70
Splice	60	1000	2175
WDBC	30	379	190
w1a	300	2477	47272

Table 4: Results for Real-World Datasets: BER (%) and Number of Selected Variables

Methods	F score		L1 SVM		D-SVM		SVM-RFE		Elimination		Differentiation	
	BER	#	BER	#	BER	#	BER	#	BER	#	BER	#
Australian	11.70	8	22.67	12	11.87	10	11.55	13	**10.76**	**7**	11.55	13
Diabetes	**27.76**	**3**	30.71	8	28.51	4	**27.76**	**3**	**27.76**	**3**	34.88	8
QSAR	10.79	21	15.93	35	12.05	40	11.43	34	**9.97**	37	11.86	28
Sonar	7.25	58	8.11	52	5.73	47	5.73	52	5.73	39	**4.22**	43
Splice	6.01	14	11.00	40	10.30	60	5.28	8	**4.94**	**8**	8.90	34
WDBC	2.41	23	3.73	10	2.41	23	**1.97**	22	**1.97**	11	**1.97**	12
w1a	18.95	295	**17.34**	265	18.62	295	18.96	295	17.56	224	18.96	300

#: *The number of variables with the minimum BER*

- **Sonar dataset** [Lichman (2013)]: The set concerns sonar signals. The task is to classify whether the signals were bounced off a metal cylinder or a roughly cylindrical rock. The set consists of 208 samples and 60 variables.

- **Splice dataset** [Lichman (2013)]: The set concerns splice junctions in a DNA sequence. Here, we classified whether the boundary is exon or intron using 1000 samples with 60 variables. We also used 2175 test samples to evaluate the prediction accuracy of the trained classier.

- **WDBC dataset** [Lichman (2013)]: The set concerns classification of malignant and benign breast cancers. The data was on 569 patients, and the variables were obtained from digitized images of fine needle aspirates of breast masses.

- **w1a dataset** [Chang et al. (2001)]: The set concerns web page classification [Platt (1999)]. The data set has 300 keyword variables.

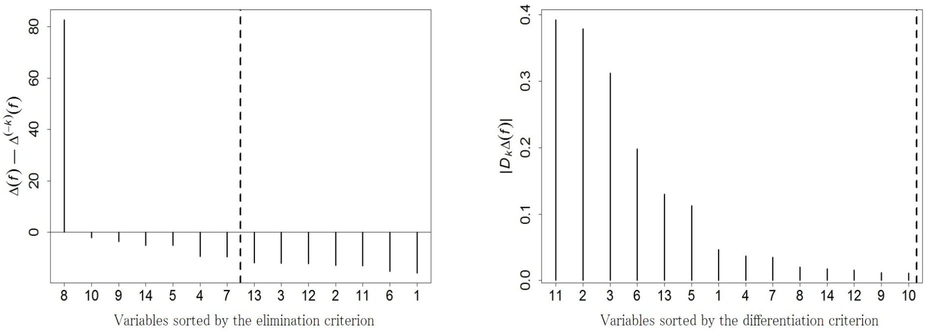

Figure 2: Variable Sensitivity of Australian Data Set: The left figure illustrates the variable sensitivity determined by the elimination criterion. For the sake of simplicity, we calculated $\Delta(f) - \Delta^{(-k)}(f)$ for each variable. The right figure illustrates the variable sensitivity determined by the differentiation criterion. The variables on the left side of the solid lines of both figures are those used when the classifiers recorded the minimum BER.

4.2.2. Results of the Experiments

Table 4 illustrates the results of the experiments by tabulating the minimum BER values (%) and the number of variables. The minimum BER of each data set among the six classifiers is written in boldface text. If the number of variables is also the smallest, the numbers are also in boldface.

The BERs of L1 SVM were worse than the other methods, except for w1a data, which implies that the other data sets have a nonlinear structure. For the Australian, Diabetes, and Splice data, the variable elimination criterion gave outstanding results. For each of them, the variables selected by the criterion gave the minimum BER with the smallest number of variables. Figure 2 shows the variable sensitivity of the proposed methods for the Australian dataset. In the left figure with the elimination criterion, variable x_8 was regarded as quite important. This was the desired result because the signs of the variable correspond to class labels with a ratio of around 85%. As we mentioned in Section 4.1, the sensitivity of the elimination method was also good. Unfortunately, the BER of the test data with the single variable x_8 was 12.58 %, which was slightly worse than that of the best model, 10.76%, with 7 variables. In the future, we will try to improve the criterion in the sense of the prediction error.

For the Sonar data, the variable differentiation criterion gave the minimum BER (4.22%) with a relatively small variable subset (#:43). Moreover, the elimination criterion gave the same BER as the other two wrapper methods while using fewer variables. For the WDBC data, the elimination and differentiation criteria improved the classification accuracy with one or two additional variables compared with L1-SVM. Additionally, compared with the other methods, they gave the minimum BER (1.97%) with fewer variables (#:11 or 12). For the QSAR data, the elimination criterion gave the minimum BER. For the w1a data, it gave the second smallest BER with the smallest number of variables. Overall, our variable elimination criteria found the best subset of variables.

5. Conclusion

In classification analysis, variable selection helps to prevent models from overfitting and reduces computational costs. In this paper, we focused on classification analysis with SVM and proposed variable selection strategies for the wrapper approach. Each method evaluates the variation that variables exert on the target function, which increases as the classifier's accuracy. The variable elimination criterion evaluates the variation by removing variables, whereas the variable differentiation criterion evaluates it by differentiating the target function. Note that these methods can calculate the importance of all variables by making only one optimization for the SVM parameter. Therefore, they are computationally efficient.

To validate the proposed methods, we applied them to several datasets. The results showed that they are more appropriate than other methods [Chen et al. (2006)] [Fan et al. (2008)] [Byvatov et al. (2004)] [Guyon et al. (2002)]. The elimination criterion succeeded in selecting all the valid variables in the artificial datasets from the MLC++ database [Kohavi et al. (1994)]. The differentiation criterion also gave good results. Moreover, the elimination criterion showed its validity on real-world datasets from the UCI [Lichman (2013)] and LIBSVM [Chang et al. (2001)] databases. For the Australian, Diabetes and Splice datasets, it gave the minimum BER value for the test data with the smallest number of variables.

Acknowledgements

The datasets used in this paper were provided by the MLC++, UCI and LIBSVM databases. The authors would like to express their appreciation to the reviewer for the constructive comments. This study was motivated by an issue raised by Nikon Corporation at Study Group Workshop 2014 hosted by the University of Tokyo. This research was supported by the Leading PhD Program in Mathematics for Key Technologies at Kyushu University and JSPS KAKENHI Grant (Number 23300106).

References

P.S. Bradley and O.L. Mangasarian (1998). Feature selection via concave minimization and support vector machines. Proceedings of the Fifteenth International Conference on Machine Learning, pp:82-90.

E. Byvatov and G. Schneider (2004). SVM-based feature selection for characterization of focused compound collections. *Journal of Chemical Information and Modeling*, 44(3):993-999.

C.C. Chang and C.J. Lin (2001). LIBSVM: A library for support vector machines. Available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`.

Y.W. Chen and C.J. Lin (2006). Combining SVMs with Various Feature Selection Strategies. *Feature Extraction Studies in Fuzziness and Soft Computing*, 207:315-324.

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

R.E. Fan, K.W. Chang, C.J. Hsieh, X.R. Wang and C.J. Lin (2008). LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871-1874.

T.S. Furey, N. Cristianini, N. Duffy, D.W. Bednarski, M. Schummer and D. Haussler (2000). Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16(10):906-914.

G. George and V.C. Raj (2001). Review on feature selection techniques and the impact of SVM for cancer classification using gene expression profile. *International Journal of Computer Science and Engineering Survey*, 2(3):16-26.

I. Guyon, J. Weston, S. Barnhill and V. Vapnik (2002). Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1):389-422.

I. Guyon, S. Gunn and A.B. Hur (2004). Result analysis of the nips 2003 feature selection challenge. *Advances in Neural Information Processing Systems*, 17:545-552.

M.A. Hall (1999). *Correlation-based Feature Selection for Machine Learning.* PhD thesis, The University of Waikato.

C.W. Hsu, C.C. Chang and C.J. Lin (2003). A practical guide to support vector classification. Available at `http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf`.

A. Karatzoglou, A. Smola, K. Hornik and A. Zeileis (2004). kernlab - an S4 package for kernel methods in R. *Journal of Statistical Software*, 11(9).

R. Kohavi, G. John, R. Long, D. Manley and K. Pfleger (1994). MLC++: A machine learning library in C++. Available at `http://www.sgi.com/tech/mlc/`.

M. Lichman (2013). UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science. Available at `http://archive.ics.uci.edu/ml`.

J. Platt (1999). Fast training of support vector machines using sequential minimal optimization. *Advances in kernel methods ? Support Vector Learning*, MIT Press Cambridge, MA, USA.

K. Polat and S. Gunes (2009). A new feature selection method on classification of medical datasets: Kernel F-score feature selection. *Expert Systems with Applications*, 36(7):10367-10373.

K.S. Shin, T.S. Lee and H. Kim (2005). An application of support vector machines in bankruptcy prediction model. *Expert Systems with Applications*, 28(1):127-135.

V. Vapnik (1998). *Statistical Learning Theory*, Wiley Interscience, New York.

V. Vapnik and O. Chapelle (2000). Bounds on error expectation for support vector machines. *Neural Computation*, 12(9):2013-2036.

L. Wang (2005). *Support Vector Machines: Theory and Applications*, Springer, Berlin, Heidelberg.

K.P. Wu and S.D. Wang (2009). Choosing the kernel parameters for support vector machines by the inter-cluster distance in the feature space. *Pattern Recognition*, 42(5):710-717.

J. Yan and V. Honavar (1998). Feature subset selection using a genetic algorithm. *Fea-*

ture Extraction, Construction and Selection, 453:117-136.

Received March 15, 2016
Revised October 4, 2016