Embedded variable selection method using signomial classification

KYOUNGMI HWANG,

KYUNGSIK LEE,

CHUNGMOK LEE,

AND SUNGSOO PARK

TECHNICAL REPORT

NOV 21, 2013

DEPARTMENT OF INDUSTRIAL AND SYSTEMS ENGINEERING

KOREA ADVANCED INSTITUTE OF SCIENCE AND TECHNOLOGY (KAIST)

EMBEDDED VARIABLE SELECTION METHOD USING SIGNOMIAL CLASSIFICATION

This article has been submitted to [Information Sciences].

* Contact Address :
Sungsoo Park, Professor,
Department of Industrial and Systems Engineering
Korea Advanced Institute of Science and Technology (KAIST)
291 Daehak-ro, Yuseong-gu, Daejeon, 305-701, Republic of Korea
PHONE : +82-42-350-3121
FAX : +82-42-350-3110
E-mail : sspark@kaist.ac.kr

Embedded variable selection method using signomial classification

Kyoungmi Hwang[a], Kyungsik Lee[b], Chungmok Lee[c], and Sungsoo Park[a*]

[a] *Department of Industrial and Systems Engineering, KAIST*
291 Daehak-ro, Yuseong-gu, Daejeon 305-701, Republic of Korea

[b] *Industrial & Management Engineering, Hankuk University of Foreign Studies*
89 Wangsan-ri, Mohyeon-myon, Yongin-si, Gyeonggi-do 449-791, Republic of Korea

[c] *Optimization, IBM Research - Ireland*
Building 3, Damastown Industrial Park, Mulhuddart, Dublin 15, Ireland

Abstract

We propose two variable selection methods using signomial classification. We attempt to select, among a set of the input variables, the variables that lead to the best performance of the classifier. One method repeatedly removes variables based on backward selection, whereas the second method directly selects a set of variables by solving an optimization problem. The proposed methods conduct variable selection considering nonlinear interactions of variables and obtain a signomial classifier with the selected variables. Computational results show that the proposed methods more effectively selects desirable variables for predicting output and provide the classifiers with better or comparable test error rates, as compared with existing methods.

Keywords: classification problems ; variable selection ; embedded method ; signomial classification ;

1 Introduction

We consider a given set of m examples of observed data $(\mathbf{x}_i, y_i)_{i=1}^m$ consisting of n input variables $x_{i,j}$, $j = 1, \ldots, n$, and one output variable y_i, $i = 1, ..., m$. A variable selection problem involves selecting, among a set of the n input variables, the variables that are desirable for predicting output. Variable selection plays an important role in data-mining applications because it can improve the prediction performance of classifiers, help constructing faster and more cost-effective classifiers, and sometimes provide a better understanding of the underlying process that generated the data (Guyon and Elisseeff, 2003).

There are three distinct approaches for variable selection in the literature: filters, wrappers, and embedded methods (Guyon and Elisseeff, 2003; Guyon et al., 2002; Kohavi and John, 1997; Lal et al., 2006). Filters assess the merits of variables using measures such as information, distance, dependency and consistency, and

[*] Corresponding author. E-mail: sspark@kaist.ac.kr, Phone:82-42-350-3121, Fax:82-42-350-3110

select input variables as a pre-processing step, independently of the chosen learning machine (Dash et al., 2002; Torkkola, 2003; Yu and Liu, 2003). Filters are usually fast but do not account for the effects of a set of variables on the performance of the learning machine.

Wrappers search through the variable space and evaluate subsets of variables using the learning machine itself as a black box (Kohavi and John, 1997; Kohavi and Sommerfield, 1995). Because wrappers require no knowledge about the specific structure of the classification or regression function, they can be combined with any learning machine. Most wrappers are slower than other methods, because, at each iteration, wrappers need to retrain and evaluate the learning machine using accuracy estimation techniques.

In contrast to filters and wrappers, embedded methods incorporate variable selection as part of the training process. In embedded methods, the learning machine structure plays a crucial role; therefore, they are usually specific to given learning machines (Guyon and Elisseeff, 2003; Lal et al., 2006). For example, the embedded method proposed by Weston et al. (2000) is valid for support vector machines (SVMs) only. Embedded methods make better use of the available data than wrappers, because, unlike wrappers, they do not need to split available training data into training and validation sets for variable selection (Guyon and Elisseeff, 2003).

Moreover, embedded methods can be roughly categorized into three groups: forward-backward, scaling factor optimization, and direct optimization methods (Lal et al., 2006). Forward-backward methods iteratively add or remove variables from the problem according to a method-specific selection criterion (Cun et al., 1989; Guyon et al., 2002; Hermes and Buhmann, 2000; Maldonado and Weber, 2009; Perkins et al., 2003; Rakotomamonjy, 2003; Rivals and Personnaz, 2003; Stoppiglia et al., 2003). One of the common methods includes using the objective function as a selection criterion (Cun et al., 1989; Guyon et al., 2002; Hermes and Buhmann, 2000; Perkins et al., 2003; Rivals and Personnaz, 2003; Stoppiglia et al., 2003). For example, the change of the objective function (Guyon et al., 2002; Rivals and Personnaz, 2003; Stoppiglia et al., 2003) or the absolute value of the derivative of the objective function (Perkins et al., 2003) can be used as a selection criterion. Other methods use the sensitivity of the leave-one-out error (Rakotomamonjy, 2003; Rivals and Personnaz, 2003) instead of that of the objective function.

Another method is to assess the variable relevance using scaling factors (scaling factor optimization method). Scaling factors are hyper-parameters adjusted by model selection. Among the methods using SVMs, some convert the scaling factors to parameters of the learning machine (Grandvalet and Canu, 2002; Maldonado et al., 2011), and others optimize the scaling factors to minimize a generalization bound (Chapelle et al., 2002; Weston et al., 2000). Some probabilistic methods translate variable selection problems into estimating the posterior distribution over the weight vector (Jebara and Jaakkola, 2000; Tipping, 2001).

Direct optimization methods use a sparsity term, which measures the number of selected variables. Most of these methods are limited to linear classifiers of the form $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x}$, where $\mathbf{w}$ is weight vector. The ℓ_0-norm of $\mathbf{w}$, the cardinality of nonzero components of $\mathbf{w}$ (Bradley et al., 1998; Weston et al., 2003), and the ℓ_1-norm of $\mathbf{w}$ (Bi et al., 2003; Bradley et al., 1998; Fung and Mangasarian, 2004; Tibshirani, 1996) are often used as sparsity terms. Perkins et al. (2003) add not only the ℓ_0-norm of $\mathbf{w}$ but also the ℓ_1-norm or ℓ_2-norm of $\mathbf{w}$ to an objective function for additional regularization. The method provided by Weston et al. (2003) performs a minimization of the ℓ_0-norm with nonlinear classifiers. However, it finds a subset of features in the high-dimensional space induced by a kernel rather than that of input variables.

In this paper, we develop two variable selection methods for nonlinear classification using the signomial classification (SC) method recently proposed by Lee et al. (2012). One method attempts to repeatedly remove variables based on backward selection, and the other is a direct optimization method that selects a set of variables simultaneously. These methods additionally obtain a classifier with the selected variables without requiring an additional learning process. We compare the proposed methods with the Pearson correlation filter method (Biesiada and Duch, 2007; Yu and Liu, 2003) and three embedded methods (Guyon et al., 2002; Rakotomamonjy, 2003; Weston et al., 2000) that are applicable to input variable selection for nonlinear classifiers.

The remainder of this paper is organized as follows. In Section 2, we introduce SC for binary classification. Section 3 presents the proposed embedded variable selection methods. Computational experiments are reported in Section 4, and conclusions are given in the last section.

2 Signomial classification (SC)

In this section, we briefly describe the SC method for binary classification proposed by Lee et al. (2012).

Let $\mathbf{x} = (x_1, ..., x_n)$ be a vector of real positive numbers, and define a function of $\mathbf{x}$, $g_{\mathbf{d}}(\mathbf{x}) = \prod_{j=1}^{n} x_j^{d_j}$, where $\mathbf{d} = (d_1, ..., d_n)$ is a real vector. Then, a signomial function of $\mathbf{x}$ is defined as follows:

$$f(\mathbf{x}) = \sum_{\mathbf{d} \in D} w_{\mathbf{d}} g_{\mathbf{d}}(\mathbf{x}) + b, \tag{1}$$

where $b \in \mathbb{R}$, $w_{\mathbf{d}} \in \mathbb{R}, \forall \mathbf{d} \in D$, and D is a finite subset of $\mathbb{R}^n$ such that $\mathbf{0} \notin D$. The set D is defined by four parameters, d_{min}, d_{max}, L, and T, as follows:

$$D = \left\{ \mathbf{d} \in \mathbb{R}^n : d_{min} \le d_j \le d_{max}, j = 1, ..., n, \sum_{i=1}^{n} |d_i| \le L, T\mathbf{d} \in \mathbb{Z}^n \right\}. \tag{2}$$

For a given set of m training data $(\mathbf{x}_i, y_i)_{i=1}^m$, where $\mathbf{x}_i \in \mathbb{R}_{++}^n$ consists of n input variables and output $y_i \in \{1, -1\}$. We seek to find a signomial classifier $f(\mathbf{x})$, which takes the form of a signomial function (1). To achieve this, we minimize the following regularized functional as given in Lee et al. (2012):

$$F(f(\mathbf{x}), \mathbf{y}) = \sum_{i=1}^{m} L(f(\mathbf{x}_i), y_i) + \lambda R(\mathbf{w}), \tag{3}$$

where L is the hinge loss function, $L(f(\mathbf{x}), y) = |1 - yf(\mathbf{x})|_+$. The regularization function is $R(\mathbf{w}) = ||\mathbf{w}||_1 = \sum_{\mathbf{d} \in D} |w_{\mathbf{d}}|$, and $\lambda > 0$ is the regularization parameter.

The problem of finding a $f(\mathbf{x})$ that minimizes (3) can be formulated as follows:

[Problem 1]

$$
\begin{aligned}
\min \ & \sum_{i=1}^{m} \varepsilon_i + \lambda \sum_{\mathbf{d}\in D} |w_{\mathbf{d}}| \\
\text{s.t.} \ & 1 - y_i \left\{ \sum_{\mathbf{d}\in D} w_{\mathbf{d}} g_{\mathbf{d}}(\mathbf{x}_i) + b \right\} \le \varepsilon_i, \qquad \forall i = 1, \ldots, m, \\
& \mathbf{w} \in \mathbb{R}^{|D|}, b \in \mathbb{R}, \varepsilon \in \mathbb{R}^m_+,
\end{aligned}
$$

where $\lambda > 0$ is the regularization parameter and D is the set of exponents defined as (2). If we replace $w_{\mathbf{d}}$ with $w^+_{\mathbf{d}} - w^-_{\mathbf{d}}$, where $w^+_{\mathbf{d}} \ge 0$ and $w^-_{\mathbf{d}} \ge 0$, then we can linearize Problem 1 as follows:

[Problem 2]

$$
\begin{aligned}
\min \ & \sum_{i=1}^{m} \varepsilon_i + \lambda \sum_{\mathbf{d}\in D} (w^+_{\mathbf{d}} + w^-_{\mathbf{d}}) \qquad (4) \\
\text{s.t.} \ & 1 - y_i \left\{ \sum_{\mathbf{d}\in D} (w^+_{\mathbf{d}} - w^-_{\mathbf{d}}) g_{\mathbf{d}}(\mathbf{x}_i) + b \right\} \le \varepsilon_i, \qquad \forall i = 1, \ldots, m, \\
& \mathbf{w}^+, \mathbf{w}^- \in \mathbb{R}^{|D|}_+, b \in \mathbb{R}, \varepsilon \in \mathbb{R}^m_+.
\end{aligned}
$$

The size of the exponent set D can be exponentially large, which makes Problem 2 practically intractable. Lee et al. (2012) showed that Problem 2 is NP-hard. However, Lee at al. (2012) proposed an efficient algorithm to solve Problem 2 that is based on the column generation method (Bertsimas and Tsitsiklis, 1997). Their algorithm iteratively generates promising elements (exponent vector) $d \in D$ to form a subset $\hat{D} \subset D$ and finds a near-optimal solution $(\hat{\mathbf{w}}^+, \hat{\mathbf{w}}^-, \hat{b})$. Then, the resulting signomial classifier is as follows:

$$
f(\mathbf{x}) = \sum_{\mathbf{d}\in \hat{D}} (\hat{w}^+_{\mathbf{d}} - \hat{w}^-_{\mathbf{d}}) g_{\mathbf{d}}(\mathbf{x}) + \hat{b},
$$

where $\mathbf{x}$ is classified in class $y = 1$ if $f(\mathbf{x}) > 0$, and $\mathbf{x}$ is classified in class $y = -1$ if $f(\mathbf{x}) < 0$.

3 Variable selection using the signomial classification

In this section, we attempt to find a subset of size k among n variables ($k < n$) which maximizes the performance of the classifier. For this, we propose two variable selection methods using SC. One is a backward-elimination method that repeatedly removes variables based on backward selection until k variables remain, and the other is a direct optimization method that simultaneously selects k variables.

3.1 Backward-elimination variable selection (BE–VSSC) method

Let k be the number of variables to be selected. The BE-VSSC method repeats the elimination process based on backward selection (Kohavi and Sommerfield, 1995) until k variables remain and then generates nested subsets of variables of sizes $n, \ldots, k$. Many backward methods select variables to be removed iteratively by estimating changes in the objective function value incurred by excluding each of the variables (Cun et al., 1989; Guyon et al., 2002; Hermes and Buhmann, 2000; Perkins et al., 2003; Rivals and Personnaz, 2003; Stoppiglia et al., 2003). BE–VSSC derives a selection criterion from the objective function (4).

Let $Z(\mathbf{w}^+, \mathbf{w}^-, \varepsilon)$ be the value of the objective function (4):

$$Z(\mathbf{w}^+, \mathbf{w}^-, \varepsilon) = \sum_{i=1}^{m} \varepsilon_i + \lambda \sum_{\mathbf{d} \in D} (w_{\mathbf{d}}^+ + w_{\mathbf{d}}^-).$$

BE–VSSC removes one variable at a time, whose elimination minimizes the change in the $Z(\mathbf{w}^+,\mathbf{w}^-,\varepsilon)$. Suppose that we have s candidate variables for removal. Let $Z(\mathbf{w}^+, \mathbf{w}^-, \varepsilon)^s$ be the value of the objective function with all the s variables, and let $Z(\mathbf{w}^+, \mathbf{w}^-, \varepsilon)_j^{s-1}$ be the objective value for a variable subset excluding variable j of the s variables. The BE–VSSC method removes variable j with the minimum value of $|Z(\mathbf{w}^+, \mathbf{w}^-, \varepsilon)^s$ $- Z(\mathbf{w}^+, \mathbf{w}^-, \varepsilon)_j^{s-1}|$. The elimination process is repeated until k variables remain. To compute $Z(\mathbf{w}^+,\mathbf{w}^-,$ $\varepsilon)_j^{s-1}$, we should retrain a signomial classifier for every candidate variable. Thus, we need to solve Problem 2 $n(n+1) - k(k-1)/2$ times to select k variables.

To reduce the computational burden, we introduce a simple heuristic approach: We assume no change in the values of $\mathbf{w}^+$, $\mathbf{w}^-$, and ε when we exclude one variable. Let $(\hat{\mathbf{w}}^{+,s}$, $\hat{\mathbf{w}}^{-,s}$, $\hat{\varepsilon}^s)$ be an optimal solution to Problem 2 with those s variables. When the variable j among the s variables is excluded, the change in the objective function value is calculated as follows:

$$\begin{aligned}|Z(\mathbf{w}^+, \mathbf{w}^-, \varepsilon)^s - Z(\mathbf{w}^+, \mathbf{w}^-, \varepsilon)_j^{s-1}| &= \lambda \sum_{\mathbf{d} \in D} (\hat{w}_{\mathbf{d}}^{+,s} + \hat{w}_{\mathbf{d}}^{-,s}) - \lambda \sum_{\substack{\{\mathbf{d} \in D| \\ d_j = 0\}}} (\hat{w}_{\mathbf{d}}^{+,s} + \hat{w}_{\mathbf{d}}^{-,s}) \\ &= \lambda \sum_{\substack{\{\mathbf{d} \in D| \\ d_j \neq 0\}}} (\hat{w}_{\mathbf{d}}^{+,s} + \hat{w}_{\mathbf{d}}^{-,s}).\end{aligned}$$

We eliminate the variable j with the minimum value of $\sum_{\mathbf{d} \in D, d_j \neq 0} (\hat{w}_{\mathbf{d}}^{+,s} + \hat{w}_{\mathbf{d}}^{-,s})$ in the elimination process. We obtain nested subsets of sizes $n, \ldots, k$ by repeating the procedure for $s := n, \ldots, k$. The detailed description of the BE–VSSC method is given in Figure 1.

3.2 k–variable selection (k–VSSC) method

We develop the k–VSSC method, which simultaneously selects k variables using SC. Let $\boldsymbol{\sigma} \in \{0,1\}^n$ be a vector of indicator variables, where σ_j, $j = 1, \ldots, n$ indicates that if $\sigma_j = 1$ then the jth input variable is selected; otherwise it is not. Let $S(\boldsymbol{\sigma})$ be a sparsity function that measures the sparsity of an indicator vector $\boldsymbol{\sigma}$. We define $S(\boldsymbol{\sigma}) := ||\boldsymbol{\sigma}||_0$ representing the cardinality of the set $\{\sigma_j \mid \sigma_j = 1, j = 1, \ldots, n\}$ and the componentwise vector product operator $*$ as $\mathbf{a}*\mathbf{b} = (a_1 b_1, \ldots, a_n b_n)$. We can formulate the optimization problem that directly selects k variables as follows:

$$\min_{\sigma, f(\mathbf{z})} \{F(f(\mathbf{z}), \mathbf{y}) : \mathbf{z}_i = \mathbf{x}_i * \boldsymbol{\sigma}, i = 1, \ldots, m, S(\boldsymbol{\sigma}) = k\}. \tag{5}$$

```
**Initialize**: $\hat{D} := \emptyset$, $V := \{1, \ldots, n\}$.
**Elimination procedure**:
  **repeat**:
      Solve the Problem 2 with variables $j \in V$.
      $\hat{w}^+_{\mathbf{d}}, \hat{w}^-_{\mathbf{d}}$ ← the optimal solution to the Problem 2.
      **for** $j \in V$ **do**
          Compute $\sum_{\mathbf{d} \in D, d_j \neq 0} (\hat{w}^+_{\mathbf{d}} + \hat{w}^-_{\mathbf{d}})$.
      **end-do**
      $r := \operatorname*{argmin}_{j \in V} \sum_{\mathbf{d} \in D, d_j \neq 0} (\hat{w}^+_{\mathbf{d}} + \hat{w}^-_{\mathbf{d}})$.
      $V := V \setminus \{r\}$.
  **until** $|V| = k$.
Solve the Problem 2 with variables $j \in V$.
$\hat{w}^+_{\mathbf{d}}, \hat{w}^-_{\mathbf{d}}, \hat{b}$ ← the optimal solution to the Problem 2.
$\hat{D}$ ← the generated subset of $D$.
$f(\mathbf{x}) := \sum_{\mathbf{d} \in \hat{D}} (\hat{w}^+_{\mathbf{d}} - \hat{w}^-_{\mathbf{d}}) g_{\mathbf{d}}(\mathbf{x}) + \hat{b}$.
Return the variable set $V$ and the classifier $f(\mathbf{x})$.
```

Figure 1: Backward-elimination variable selection (BE–VSSC) method

By modifying Problem 1, we can formulate the problem (5) as follows:

[Problem 3]

$$
\begin{aligned}
&\min \sum_{i=1}^{m} \varepsilon_i + \lambda \sum_{\mathbf{d}\in D} |w_{\mathbf{d}}| \\
&\text{s.t. } 1 - y_i \left\{ \sum_{\mathbf{d}\in D} w_{\mathbf{d}} g_{\mathbf{d}}(\mathbf{x}_i * \boldsymbol{\sigma}) + b \right\} \le \varepsilon_i, \qquad \forall i = 1,\ldots,m, \\
&\sum_{j=1}^{n} \sigma_j = k, \\
&\mathbf{w} \in \mathbb{R}^{|D|}, b \in \mathbb{R}, \boldsymbol{\varepsilon} \in \mathbb{R}_+^m, \boldsymbol{\sigma} \in \{1,0\}^n,
\end{aligned}
$$

where $\lambda > 0$ is the regularization parameter and D is the set of exponents defined as (2). We replace $w_{\mathbf{d}}$ with $w_{\mathbf{d}}^+ - w_{\mathbf{d}}^-$, where $w_{\mathbf{d}}^+ \ge 0$ and $w_{\mathbf{d}}^- \ge 0$, and then obtain the following Problem 4:

[Problem 4]

$$
\begin{aligned}
&\min \sum_{i=1}^{m} \varepsilon_i + \lambda \sum_{\mathbf{d}\in D} (w_{\mathbf{d}}^+ + w_{\mathbf{d}}^-) \\
&\text{s.t. } 1 - y_i \left\{ \sum_{\mathbf{d}\in D} (w_{\mathbf{d}}^+ - w_{\mathbf{d}}^-) g_{\mathbf{d}}(\mathbf{x}_i * \boldsymbol{\sigma}) + b \right\} \le \varepsilon_i, \qquad \forall i = 1,\ldots,m, \\
&\sum_{j=1}^{n} \sigma_j = k, \\
&\mathbf{w}^+, \mathbf{w}^- \in \mathbb{R}_+^{|D|}, b \in \mathbb{R}, \boldsymbol{\varepsilon} \in \mathbb{R}_+^m, \boldsymbol{\sigma} \in \{1,0\}^n.
\end{aligned}
$$

The Problem 4, however, is a nonlinear optimization problem where some of the variables are constrained to have integer or discrete values (a mixed-integer nonlinear programming problem [MINLP]), which is generally NP-hard (Garey and Johnson, 1979; Murty and Kabadi, 1987). Note that Problem 4 with $k = n$ is the same problem as Problem 2. The fact that Problem 2 is NP-hard, which was shown by Lee et al. (2012), implies that Problem 4 is NP-hard as well.

Now, we reformulate Problem 4 as a mixed-integer linear program. Suppose that, for some j, $(w_{\mathbf{d}}^+ - w_{\mathbf{d}}^-) \neq 0$ for any $\mathbf{d} \in D$ with $d_j \neq 0$. This means that the corresponding variable j appears in the resulting classifier, and the variable j has been selected. Thus, if the variable j has not been selected, $(w_{\mathbf{d}}^+ - w_{\mathbf{d}}^-) = 0$ for all $\mathbf{d} \in D$ with $d_j \neq 0$. This can be formulated as $\sum_{\{\mathbf{d}\in D : d_j \neq 0\}} (w_{\mathbf{d}}^+ + w_{\mathbf{d}}^-) \leqslant M\sigma_j$, where M is a large number. If $\sum_{\{\mathbf{d}\in D : d_j \neq 0\}} (w_{\mathbf{d}}^+ + w_{\mathbf{d}}^-) > 0$ then $\sigma_j = 1$. As the contraposition, if $\sigma_j = 0$, then $\sum_{\{\mathbf{d}\in D : d_j \neq 0\}} (w_{\mathbf{d}}^+ + w_{\mathbf{d}}^-) = 0$. Therefore, Problem 4 can be reformulated as the following mixed-integer linear program:

```
Initialize: $\hat{D} := \emptyset$ and $V := \{1,\ldots,n\}$.
Exponent generating procedure:
  Solve the Problem 2 with all input variables.
  $D := \hat{D} \leftarrow$ the generate profitable exponents to the Problem 2 .
Variable selection procedure:
  Solve the Problem 5.
  $(\hat{\mathbf{w}}^+, \hat{\mathbf{w}}^-, \hat{b}, \hat{\boldsymbol{\sigma}}) \leftarrow$ the optimal solution to the Problem 5.
  for $j \in 1,\ldots,n$ do
    if $\hat{\sigma}_j = 0$ then $V := V \setminus \{j\}$.
  end-do
  $f(\mathbf{x}) := \sum_{\mathbf{d}\in\hat{D}} (\hat{w}^+_{\mathbf{d}} - \hat{w}^-_{\mathbf{d}}) g_{\mathbf{d}}(\mathbf{x}) + \hat{b}$.
Return the variable subset $V$ and the classifier $f(\mathbf{x})$.
```

Figure 2: k–variable selection (k–VSSC) method

[Problem 5]

$$
\begin{aligned}
\min\ & \sum_{i=1}^{m} \varepsilon_i + \lambda \sum_{\mathbf{d}\in D} (w^+_{\mathbf{d}} + w^-_{\mathbf{d}}) & \\
\text{s.t. }\ & 1 - y_i \left\{ \sum_{\mathbf{d}\in D} (w^+_{\mathbf{d}} - w^-_{\mathbf{d}}) g_{\mathbf{d}}(\mathbf{x}_i) + b \right\} \le \varepsilon_i, & \forall i = 1,\ldots,m, \\
& \sum_{\{\mathbf{d}\in D : d_j \neq 0\}} (w^+_{\mathbf{d}} + w^-_{\mathbf{d}}) \leqslant M\sigma_j, & \forall j = 1,\ldots,n, \\
& \sum_{j=1}^{n} \sigma_j = k, & \\
& \mathbf{w}^+, \mathbf{w}^- \in \mathbb{R}^{|D|}_+, b \in \mathbb{R}, \boldsymbol{\varepsilon} \in \mathbb{R}^m_+, \boldsymbol{\sigma} \in \{1,0\}^n, &
\end{aligned}
$$

where M is a large number.

To solve Problem 5, we first form a subset $\hat{D}$ of D by solving Problem 2, instead of enumerating all the elements of D defined as (2). To generate as many promising exponents as possible, we solve the Problem 2 by using the algorithm proposed by Lee et al. (2012) with a small value of λ (e.g. 10^{-3}). Then, we set $D := \hat{D}$ and solve Problem 5 using a branch-and-bound algorithm. Branch and bound is a general algorithm that finds optimal solutions to discrete and combinatorial problems (Lawler and Wood, 1966). The detailed description of the k–VSSC method is given in Figure 2.

Table 1: Data sets used for experiments

Data set	# of Instances		# of Variables
	Positive	Negative	
BLD	200	145	6
PID	268	500	8
BC	444	239	9
HD	137	160	13
WDBC	212	357	30

4 Computational experiments

4.1 *Computational setting*

Computational experiments were conducted on real-world data sets to evaluate the performance of the proposed methods: BE–VSSC and k–VSSC. The proposed methods were tested on five medical data sets: Bupa Liver Disorders (BLD), Pima Indians Diabetes (PID), Breast Cancer (BC), Heart Disease (HD), and Wisconsin Diagnostic Breast Cancer (WDBC) data sets. We obtained the data sets from the University of California, Irvine (UCI) repository of machine learning databases (Bache and Lichman, 2013). Table 1 presents a description of the data sets.

We compared the performance of the proposed methods with those of the Pearson correlation filter method (Biesiada and Duch, 2007; Yu and Liu, 2003) and three embedded methods: R^2W^2 (Weston et al., 2000), BE–R^2W^2 (Rakotomamonjy, 2003), and Recursive Feature Elimination (RFE) (Guyon et al., 2002). The embedded methods were included for the comparison because they are applicable to select input variables for nonlinear classifiers and are embedded in a SVM. The correlation filter selects input variables independently of the classification method. Therefore, the performance of the correlation filter was evaluated with the classifiers obtained by other classification methods: k-nearest neighbor (kNN) (Bay, 1998; Cover and Hart, 1967), classification and regression tree (CART) (Breiman et al., 1984), and logistic regression (LR) (Hosmer and Lemeshow, 2005).

We conducted variable selection for all the possible numbers of variables to be selected, $k = 1, ..., n$. The BE–VSSC and k–VSSC methods were implemented with the Xpress Mosel language (Xpress, 2012). For the other methods, we used well-known implementations for Matlab (MATLAB, 2010): the Spider Toolbox (Weston et al., 2006) for R^2W^2 and the SVM-KM Toolbox (Canu et al., 2005) for the correlation filter, BE–R^2W^2, and RFE.

For parameter setting and performance testing, each data set was divided into three disjoint subsets: training, validation, and test sets. We randomly selected the subsets in the ratio of 5:3:2 100 times. For various parameter settings, we trained and evaluated classifiers using 100 pairs of training and validation sets, respectively. We selected model parameters that achieved the best average of classification error rates on the validation sets. With the selected model parameters, we performed variable selection and trained classifiers using the training sets. Note that, for the LR, we used training and validation data sets together without model selection. We predicted the test set using the resulting classifier trained with the selected variables. As a performance criterion, we used the average test error rates.

For model selection, the following set of values for the parameters (regularization parameter λ, penalty parameter C, Gaussian kernel width γ, degree of the polynomial function d, the number of nearest neighbors k, and pruning level p) were used:

$$
\begin{aligned}
\lambda \in & \{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3\}, \\
C \in & \{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3\}, \\
\gamma \in & \{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3\}, \\
d \in & \{2, 3, 4, 5, 6, 7, 8\}, \\
k \in & \{1, 2^1, 2^2, \ldots\}, \\
p \in & \{0, 1, 2, 3, 4, 5, 6\}.
\end{aligned}
$$

We used a Gaussian RBF kernel $K(x_i, x_j) := exp(-\gamma \|x_i - x_j\|^2)$ and polynomial kernel $K(x_i, x_j) := (x_i \cdot x_j)^d$ for the BE–R^2W^2 and RFE methods and a polynomial kernel $K(x_i, x_j)$ for the R^2W^2 method.

We used 7 × 7 combinations of penalty parameters C and kernel parameters d for the R^2W^2 method and 2 × 7 × 7 combinations of types of kernels (RBF or polynomial kernels), penalty parameters C, and kernel parameters γ or d for the BE–R^2W^2 and RFE methods. The kNN was tested using seven different numbers of k-nearest neighbors, and the CART was tested using seven different pruning levels p. We tested the proposed methods with the set $D := \{\mathbf{d} \in \mathbb{R}^n : -1 \leq d_j \leq 1, j = 1, ..., n, \sum_{i=1}^n |d_i| \leq 1, 10\mathbf{d} \in \mathbb{Z}^n\}$ and seven regularization parameter values of λ.

For the proposed methods, we had to translate the original data into the range $[1, \infty)$ because the definition of the signomial function (1). On the other hand, for the R^2W^2, BE–R^2W^2, and RFE methods, data scaling is an important preprocessing step to using a SVM (Hsu et al., 2003). We linearly scaled input data into the range $[-1, 1]$ and translated the scaled data by adding 2 for the proposed methods.

4.2 Computational results

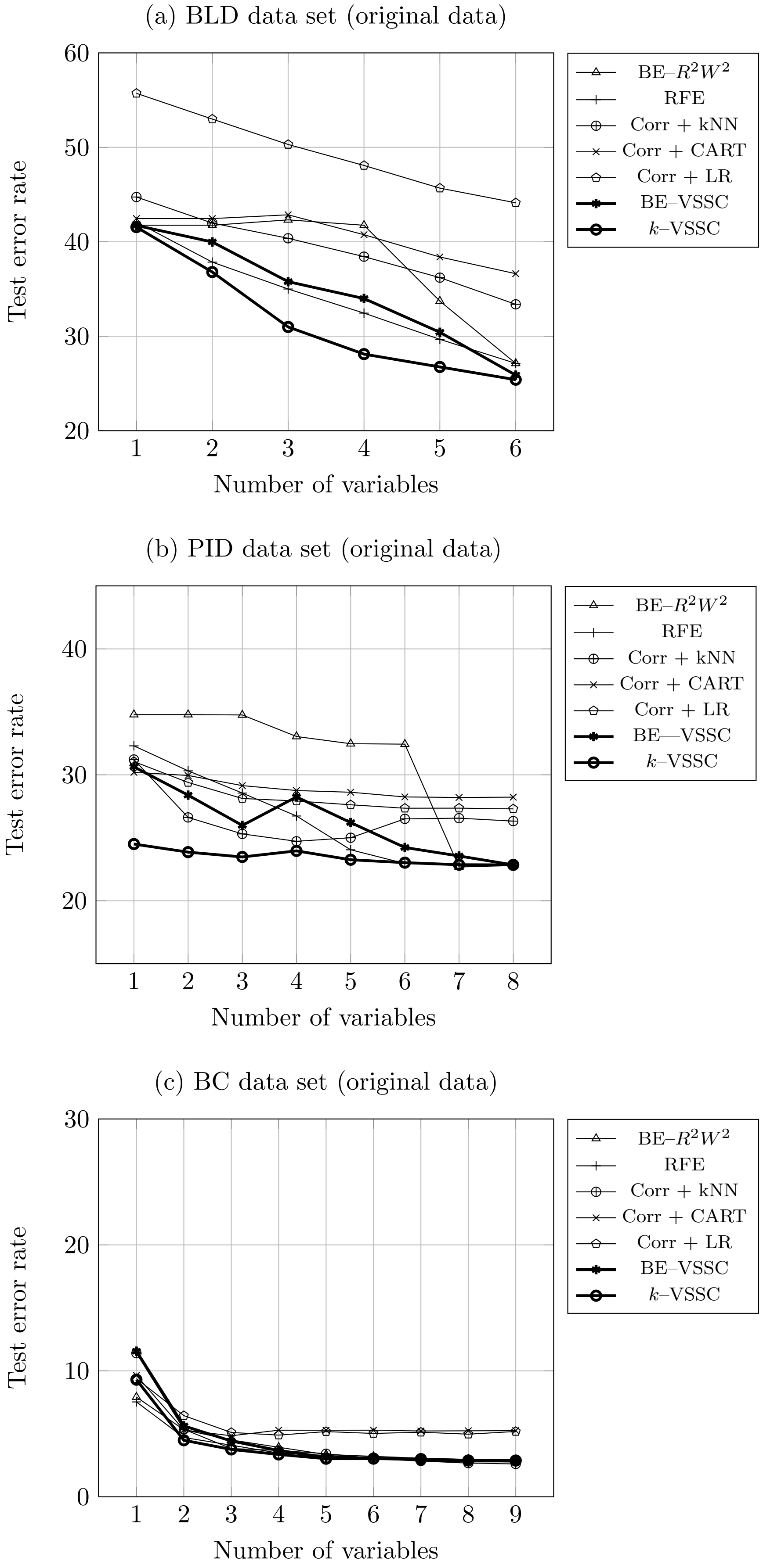
The test results are presented in Figures 3 through 6 and Tables 2 and 3. The '+' denotes the combination of the correlation filter and classification methods: kNN, CART, and LR. The average test error rates for the number of the selected variables are plotted in Figures 3 through 6 for each of the data sets. The horizontal axis denotes the number of the selected variables, and the vertical axis denotes the average test error rate. The standard deviation of the average test error rate has not been plotted for the sake of clarity. The average test error rates on the original data are reported in Figures 3 and 4, and those on the scaled data are reported in Figures 5 and 6. Tables 2 and 3 show the overall experimental results for the original data and scaled data, respectively. In Tables 2 and 3, we reported the average test error rate and standard deviation for each of the data sets.

Because the polynomial kernel provided considerably worse test error rates than the Gaussian RBF kernel in the experiments on the original data, we selected the Gaussian RBF kernel for the BE–R^2W^2 and RFE methods and did not report the experimental results of R^2W^2, which used a polynomial kernel. In the experiments on the scaled data, for the R^2W^2, BE–R^2W^2, and RFE methods, we selected a polynomial kernel for the BLD, PID, and WDBC data sets and a Gaussian RBF kernel for the HD and BC data sets.

In Figures 3 through 6, the closer a line is to the bottom, the better the average test error rate. Thus, the method with the line near the bottom more effectively selects desirable variables for predicting output, as compared with the other methods. For all data sets, the lines for k–VSSC are located below the lines of other methods through the original and scaled data. Comparatively, the lines for BE–VSSC are located near the bottom. The proposed methods can select variables better than or comparably to the other methods.

Table 2 displays experimental results for the original data. k–VSSC performed the best in terms of providing average test error rates for each of the data sets. BE–VSSC performed second-best for the PID, HD, and WDBC data sets and third-best for the BLD data set. Overall, for the original data, the proposed methods exhibited the best variable selection performances. Among the other methods, RFE achieved the best average test error rate.

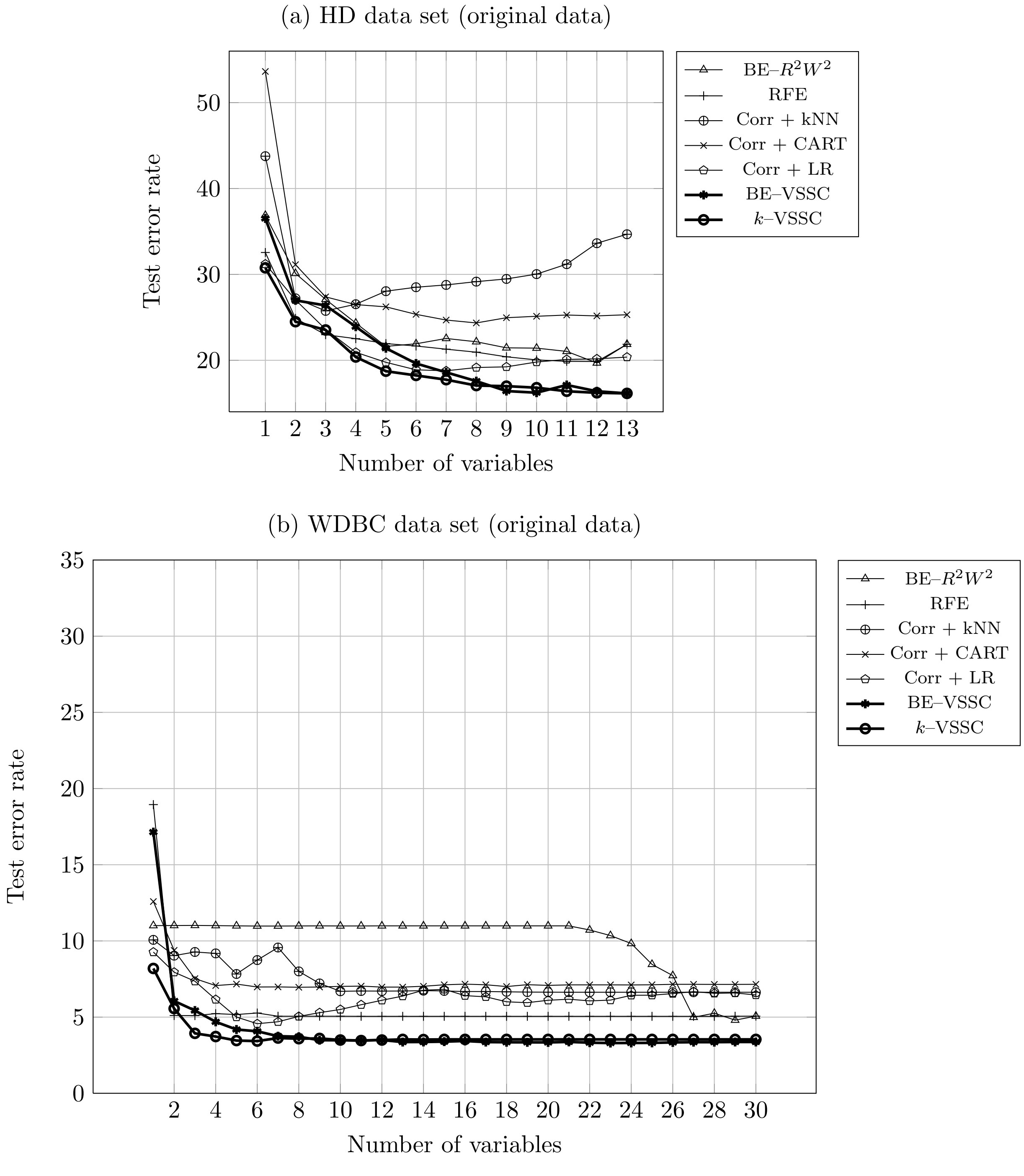
Table 3 represents experimental results for the scaled data. k–VSSC performed the best in terms of providing average test error rates for each of the data sets. R^2W^2 performed second-best for the BLD and PID data sets, whereas for the WDBC data set, RFE performed second-best. BE–VSSC performed third-best for the BLD, PID, and WDBC data sets. Overall, also for the scaled data, the proposed methods exhibited the best variable selection performance, and among the other methods, RFE provided the best average test error rate.


(a) BLD data set (original data)

(b) PID data set (original data)

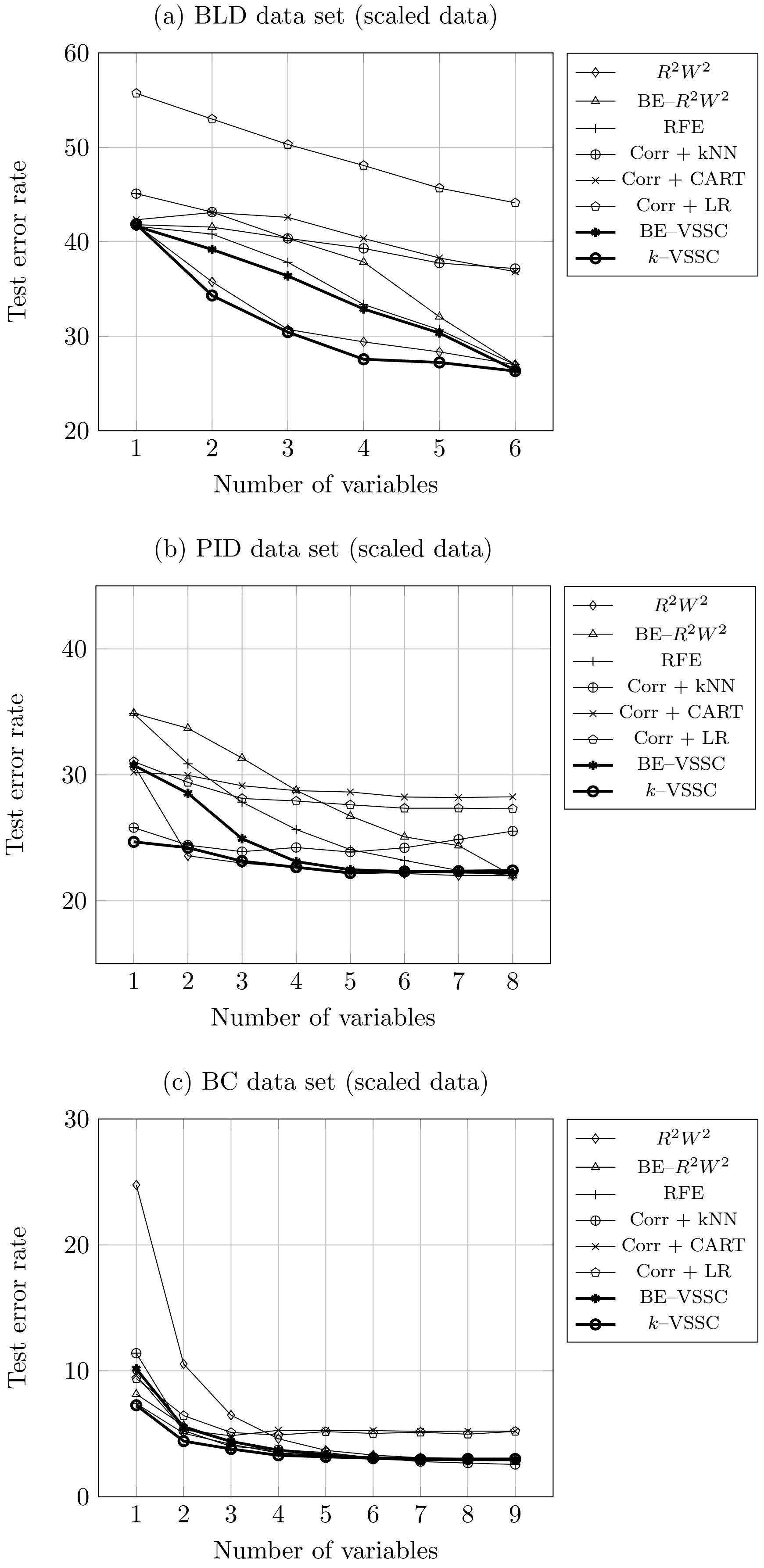
(c) BC data set (original data)

Figure 3: Average test error rate (%) vs. the number of selected variables for BLD, PID, and BC data sets (original data).


(a) HD data set (original data)

(b) WDBC data set (original data)

Figure 4: Average test error rate (%) vs. the number of selected variables for HD and WDBC data sets (original data).

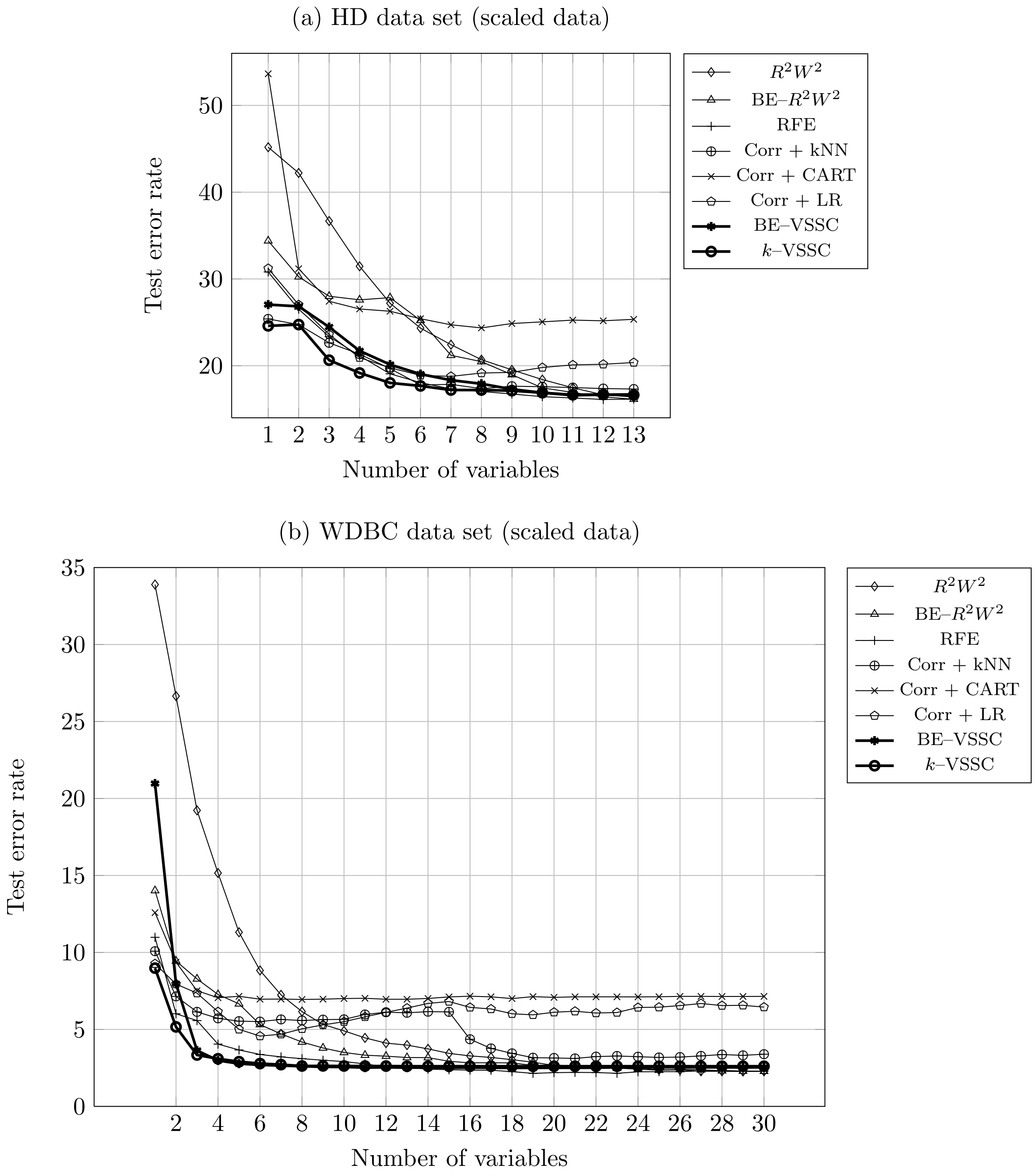


(a) BLD data set (scaled data)

(b) PID data set (scaled data)

(c) BC data set (scaled data)

Figure 5: Average test error rate (%) vs. the number of selected variables for BLD, PID, and BC data sets (scaled data).


(a) HD data set (scaled data)

(b) WDBC data set (scaled data)

Figure 6: Average test error rate (%) vs. the number of selected variables for HD and WDBC data sets (scaled data).

Table 2: Performance results (%): Average test error rate and standard deviation for all the numbers of selected variables. For each data set, the two best performance results are printed in bold, as compared with the other methods. All the methods were tested on the original data.

Data set	BE–R^2W^2	RFE	Corr + kNN	Corr + CART	Corr + LR	BE–VSSC	k–VSSC
BLD	38.07 ± 2.12	**34.03 ± 4.53**	39.18 ± 4.89	40.71 ± 4.66	49.48 ± 3.82	34.63 ± 4.05	**31.60 ± 3.45**
PID	30.97 ± 1.32	26.31 ± 3.63	26.53 ± 2.39	28.92 ± 2.86	28.26 ± 2.50	**25.12 ± 3.36**	**23.32 ± 2.20**
BC	4.12 ± 1.20	**3.86 ± 1.13**	4.32 ± 1.39	5.71 ± 1.85	5.70 ± 1.65	4.46 ± 1.40	**3.97 ± 1.26**
HD	24.01 ± 4.59	22.37 ± 4.14	30.52 ± 4.53	28.08 ± 5.29	21.46 ± 4.05	**20.95 ± 4.13**	**19.51 ± 3.40**
WDBC	9.93 ± 1.77	5.54 ± 1.99	7.29 ± 1.77	7.35 ± 1.85	6.25 ± 2.01	**4.11 ± 1.80**	**3.77 ± 1.47**
Average	21.42 ± 2.20	18.42 ± 3.08	21.57 ± 2.99	22.15 ± 3.30	22.23 ± 2.80	**17.85 ± 2.95**	**16.44 ± 2.35**

Table 3: Performance results (%): Average test error rate and standard deviation for all the numbers of selected variables. For each data set, the two best performance results are printed in bold, as compared with the other methods. All the methods were tested on the scaled data.

Data set	R^2W^2	BE–R^2W^2	RFE	Corr + kNN	Corr + CART	Corr + LR	BE–VSSC	k–VSSC
BLD	**32.17 ± 3.28**	36.77 ± 3.51	35.22 ± 3.67	40.47 ± 4.66	40.58 ± 4.59	49.48 ± 3.82	34.47 ± 4.24	**31.28 ± 3.52**
PID	**23.56 ± 2.47**	28.36 ± 3.40	26.36 ± 3.22	24.60 ± 2.12	28.92 ± 2.87	28.26 ± 2.49	24.57 ± 3.18	**23.00 ± 2.14**
BC	6.94 ± 2.11	4.12 ± 1.26	**3.95 ± 1.15**	4.34 ± 1.37	5.69 ± 1.84	5.70 ± 1.65	4.33 ± 1.34	**3.77 ± 1.07**
HD	26.05 ± 4.56	23.14 ± 3.80	19.62 ± 3.66	**19.55 ± 3.58**	28.10 ± 5.32	21.46 ± 4.05	19.97 ± 3.69	**18.71 ± 3.61**
WDBC	6.58 ± 2.52	4.03 ± 1.75	**3.08 ± 1.10**	4.79 ± 1.43	7.35 ± 1.85	6.25 ± 2.00	3.40 ± 1.44	**2.97 ± 1.10**
Average	19.06 ± 2.99	19.29 ± 2.75	17.64 ± 2.56	18.75 ± 2.63	22.13 ± 3.30	22.23 ± 2.80	**17.35 ± 2.78**	**15.95 ± 2.29**

5 Conclusion

We proposed two embedded variable selection methods using SC: BE–VSSC and k-VSSC. The proposed methods select a set of the input variables considering nonlinear interactions of variables and provide a signomial classifier with the selected variables. The classifiers trained with the variables selected by the proposed methods provide better or comparable average test error rates when compared with those of the existing methods. Thus, the proposed methods select variables that are desirable for predicting output. In the future, extending the proposed methods to multi-class classification or regression problems may be considered.

Acknowledgments

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2013-025297).

References

Bache, K. and Lichman, M. (2013), 'University of california, irvine (UCI) machine learning repository'.
URL: *http://archive.ics.uci.edu/ml*

Bay, S. D. (1998), Combining nearest neighbor classifiers through multiple feature subsets, *in* 'Proceedings of the 15th International Conference on Machine Learning', ICML '98, Morgan Kaufmann Publishers, Madison, WI, USA, pp. 37–45.

Bertsimas, D. and Tsitsiklis, J. N. (1997), *Introduction to Linear Optimization*, number 6 *in* 'Athena scientific series in optimization and neural computation', Athena Scientific, Belmont, MAMSC, USA.

Bi, J., Bennett, K., Embrechts, M., Breneman, C. and Song, M. (2003), "Dimensionality reduction via sparse support vector machines", *Journal of Machine Learning Research* , Vol. 3, JMLR.org, pp. 1229–1243.

Biesiada, J. and Duch, W. (2007), Feature selection for high-dimensional data – a Pearson redundancy based filter, *in* 'Computer Recognition Systems 2', Vol. 45 of *Advances in Soft Computing*, Springer, pp. 242–249.

Bradley, P. S., Mangasarian, O. L. and Street, W. N. (1998), "Feature selection via mathematical programming", *INFORMS Journal on Computing* , Vol. 10, INFORMS, Institute for Operations Research and the Management Sciences (INFORMS), Linthicum, Maryland, USA, pp. 209–217.

Breiman, L., Friedman, J. H., Olshen, R. A. and Stone, C. J. (1984), *Classification and Regression Trees*, Wadsworth International Group, Belmont, Calif.

Canu, S., Grandvalet, Y., Guigue, V. and Rakotomamonjy, A. (2005), 'SVM and kernel methods matlab toolbox', Perception Systemes et Information, INSA de Rouen, Rouen, France.

Chapelle, O., Vapnik, V., Bousquet, O. and Mukherjee, S. (2002), "Choosing multiple parameters for support vector machines", *Machine Learning* , Vol. 46, Kluwer Academic Publishers, Hingham, MA, USA, pp. 131–159.

Cover, T. and Hart, P. (1967), "Nearest neighbor pattern classification", *IEEE Transactions on Information Theory* , Vol. 13, pp. 21–27.

Cun, Y. L., Denker, J. S. and Solla, S. A. (1989), Optimal brain damage, *in* 'Proceedings of the 2nd Annual Conference on Neural Information Processing Systems', NIPS '89, Morgan Kaufmann Publishers, Denver, CO, USA, pp. 598–605.

Dash, M., Choi, K., Scheuermann, P. and Liu, H. (2002), Feature selection for clustering – a filter solution, *in* 'Proceedings of the 2nd International Conference on Data Mining', ICDM '02, IEEE Computer Society, Maebashi, Japan, pp. 115–122.

Fung, G. M. and Mangasarian, O. L. (2004), "A feature selection newton method for support vector machine classification", *Computational Optimization and Applications* , Vol. 28, Kluwer Academic Publishers, Norwell, MA, USA, pp. 185–202.

Garey, M. R. and Johnson, D. S. (1979), *Computers and Intractability: A Guide to the Theory of NP-Completeness*, A Series of Books in the Mathematical Sciences, W. H. Freeman and Company, New York, NY, USA.

Grandvalet, Y. and Canu, S. (2002), Adaptive scaling for feature selection in SVMs, *in* 'Proceedings of the 15th Annual Conference on Neural Information Processing Systems', NIPS '02, MIT Press, Vancouver, BC, Canada, pp. 553–560.

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

Guyon, I., Weston, J., Barnhill, S. and Vapnik, V. (2002), "Gene selection for cancer classification using support vector machines", *Machine Learning* , Vol. 46, Springer Netherlands, pp. 389–422.

Hermes, L. and Buhmann, J. M. (2000), Feature selection for support vector machines, *in* 'Proceedings of the 15th International Conference on Pattern Recognition', Vol. 2 of *ICPR '00*, IEEE Computer Society, Barcelona, Spain, pp. 716–719.

Hosmer, D. and Lemeshow, S. (2005), *Applied Logistic Regression (Wiley Series in Probability and Statistics)*, 2nd edn, Wiley-Interscience Publication, New York, NY, USA.

Hsu, C. W., Chang, C. C. and Lin, C. J. (2003), A practical guide to support vector classification, Technical report, Department of Computer Science, National Taiwan University, Taipei 106, Taiwan.

Jebara, T. and Jaakkola, T. (2000), Feature selection and dualities in maximum entropy discrimination, *in* 'Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence', UAI '00, Morgan Kaufmann Publishers, Stanford, CA, USA,, pp. 291–300.

Kohavi, R. and John, G. H. (1997), “Wrappers for feature subset selection”, *Artificial Intelligence* , Vol. 97, Elsevier Science Publishers Ltd., Essex, UK, pp. 273–324.

Kohavi, R. and Sommerfield, D. (1995), Feature subset selection using the wrapper method: Overfitting and dynamic search space topology, *in* ‘Proceedings of the 1st International Conference on Knowledge Discovery and Data Mining’, KDD ’95, AAAI Press, Montreal, QC, Canada, pp. 192–197.

Lal, T. N., Chapelle, O., Weston, J. and Elisseeff, A. (2006), *Feature Extraction: Foundations and Applications (Studies in Fuzziness and Soft Computing)*, Vol. 207, Springer, Berlin, Germany, chapter 5. Embedded methods, pp. 137–165.

Lawler, E. L. and Wood, D. E. (1966), “Branch-and-bound methods: A survey”, *Operations Research* , Vol. 14, INFORMS, pp. pp. 699–719.

Lee, K., Kim, N. and Jeong, M. K. (2012), “The sparse signomial classification and regression model”, *Annals of Operations Research* , Springer, pp. 1–30.

Maldonado, S. and Weber, R. (2009), “A wrapper method for feature selection using support vector machines”, *Information Sciences* , Vol. 179, Elsevier Science Inc., New York, NY, USA, pp. 2208–2217.

Maldonado, S., Weber, R. and Basak, J. (2011), “Simultaneous feature selection and classification using kernel-penalized support vector machines”, *Information Sciences* , Vol. 181, Elsevier Science Inc., New York, NY, USA, pp. 115–128.

MATLAB (2010), *version 7.10.0 (R2010a)*, The MathWorks Inc., Natick, Massachusetts.

Murty, K. G. and Kabadi, S. N. (1987), “Some NP-complete problems in quadratic and nonlinear programming”, *Mathematical Programming* , Vol. 39, Springer-Verlag New York, Inc., Secaucus, NJ, USA, pp. 117–129.

Perkins, S., Lacker, K. and Theiler, J. (2003), “Grafting: fast, incremental feature selection by gradient descent in function space”, *Journal of Machine Learning Research* , Vol. 3, JMLR.org, pp. 1333–1356.

Rakotomamonjy, A. (2003), “Variable selection using SVM based criteria”, *Journal of Machine Learning Research* , Vol. 3, JMLR.org, pp. 1357–1370.

Rivals, I. and Personnaz, L. (2003), “MLPs (mono layer polynomials and multi layer perceptrons) for nonlinear modeling”, *Journal of Machine Learning Research* , Vol. 3, JMLR.org, pp. 1383–1398.

Stoppiglia, H., Dreyfus, G., Dubois, R. and Oussar, Y. (2003), “Ranking a random feature for variable and feature selection”, *Journal of Machine Learning Research* , Vol. 3, JMLR.org, pp. 1399–1414.

Tibshirani, R. (1996), “Regression shrinkage and selection via the lasso”, *Journal of the Royal Statistical Society. Series B (Methodological)* , Vol. 58, Blackwell Publishing for the Royal Statistical Society, pp. 267–288.

Tipping, M. E. (2001), “Sparse Bayesian learning and the relevance vector machine”, *Journal of Machine Learning Research* , Vol. 1, JMLR.org, pp. 211–244.

Torkkola, K. (2003), “Feature extraction by non-parametric mutual information maximization”, *Journal of Machine Learning Research* , Vol. 3, JMLR.org, pp. 1415–1438.

Weston, J., Elisseeff, A., BakIr, G. and Sinz, F. (2006), ‘Spider toolbox’.
URL: *http://people.kyb.tuebingen.mpg.de/spider*

Weston, J., Elisseeff, A., Schölkopf, B. and Tipping, M. (2003), “Use of the zero-norm with linear models and kernel methods”, *Journal of Machine Learning Research* , Vol. 3, JMLR.org, pp. 1439–1461.

Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T. and Vapnik, V. (2000), Feature selection for SVMs, *in* ‘Proceedings of the 13th Annual Conference on Neural Information Processing Systems’, NIPS ’00, MIT Press, Denver, CO, USA, pp. 563–532.

Xpress (2012), ‘Xpress-MP 7.3’.
URL: *http://www.fico.com/en*

Yu, L. and Liu, H. (2003), Feature selection for high-dimensional data: A fast correlation-based filter solution, *in* ‘Proceedings of the 20th International Conference on Machine Learning’, ICML ’03, AAAI Press, Washington, DC, USA, pp. 56–63.