Journal of the Operational Research Society (2016)
© 2016 The Operational Research Society. All rights reserved. 0160-5682/16
www.palgrave.com/journals

Variable selection methods for multi-class classification using signomial function

Kyoungmi Hwang[1], Kyungsik Lee[2]* and Sungsoo Park[3]*

[1]Test and Package Automation Group, Giheung Hwaseong Complex, Samsung Electronics, 158 Baebang-ro, Asan-si, Chungcheongnam-do 336-711, Republic of Korea; [2]Department of Industrial Engineering and Institute for Industrial Systems Innovation, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul 151-742, Republic of Korea; and [3]Department of Industrial and Systems Engineering, KAIST, 291 Daehak-ro, Yuseong-gu, Daejeon 305-701, Republic of Korea

We develop several variable selection methods using signomial function to select relevant variables for multi-class classification by taking all classes into consideration. We introduce a ℓ_1-norm regularization function to measure the number of selected variables and two adaptive parameters to apply different importance weights for different variables according to their relative importance. The proposed methods select variables suitable for predicting the output and automatically determine the number of variables to be selected. Then, with the selected variables, they naturally obtain the resulting classifiers without an additional classification process. The classifiers obtained by the proposed methods yield competitive or better classification accuracy levels than those by the existing methods.

Journal of the Operational Research Society (2016). doi:10.1057/s41274-016-0127-x

Keywords: variable selection; multi-class classification; embedded method; signomial function

1. Introduction

In the classification context, variable selection is the process of selecting, from the entire set of input variables, those that can positively affect classifier performance and efficiency. Thereby, variable selection improves the prediction performance of classifiers and the comprehensibility of the results while also reducing the computational load (Guyon and Elisseeff, 2003). In this paper, we focus on the issue of variable selection in multi-class classification problems. Given a set of training examples $\{\mathbf{x}_i\}_{i=1}^{m}$ where each $\mathbf{x}_i$ consists of n input variables x_{ij} and belongs to one class $y_i \in K$, $|K| > 2$. We seek to find, considering all classes simultaneously, a common relevant subset of n input variables that is useful for predicting the class of a new example.

Variable selection methods can be divided into three categories: filter, wrapper and embedded (Guyon *et al*, 2002; Guyon and Elisseeff, 2003; Lal, *et al*, 2006). Filter methods score the merits of variables using intrinsic data properties such as information, distance, dependency and consistency, and then select a subset of variables as a preprocessing step independently of the choice of learning machine (Dhillon, *et al*, 2003; Torkkola, 2003; Li, *et al*, 2004; Yang and Pedersen 1997; Zhang *et al*, 2008; Bolon-Canedo *et al*, 2012; Forman, 2004; You and Li, 2011; Rajapakse and Mundra, 2013). Filter methods usually are fast, but because they do not consider variable subsets' effects on the learning process, they can select a redundant one. Wrapper methods directly use predetermined learning machines as a black box with which to score subsets of variables (Kohavi and Sommerfield, 1995; Kohavi and John, 1997; Pudil *et al*, 1994; Yang and Honavar, 1998; Somol *et al*, 2004). These methods do not need the specific structure of a classification function and so can be combined with any learning machine. They are usually good but incur a high computational cost and are inappropriate for high-dimensional data. Hybrid filter–wrapper methods, which apply both filter and wrapper methods in combination, also have been developed (Ruiz *et al*, 2006; Gutlein *et al*, 2009; Peng *et al*, 2010; Akadi *et al*, 2011; Bermejo *et al*, 2012).

Embedded methods, unlike filter and wrapper methods, incorporate variable selection as part of the training process and therefore are specific to a learning machine. Embedded methods can be roughly categorized into three types: forward–backward, scaling factor optimization, and direct optimization methods (Guyon and Elisseeff, 2003; Lal *et al*, 2006). Forward-backward methods iteratively add or remove variables by

**Correspondence: Kyungsik Lee, Department of Industrial Engineering and Institute for Industrial Systems Innovation, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul 151-742, Republic of Korea.*
E-mail: optima@snu.ac.kr
**Correspondence: Sungsoo Park, Department of Industrial and Systems Engineering, KAIST, 291 Daehak-ro, Yuseong-gu, Daejeon 305-701, Republic of Korea.*
E-mail: sspark@kaist.ac.kr

estimating changes in the objective function (Cun *et al*, 1989; Hermes and Buhmann, 2000; Guyon *et al*, 2002; Rakotomamonjy, 2003; Stoppiglia *et al*, 2003; Rivals and Personnaz, 2003; Perkins *et al*, 2003; Maldonado and Weber, 2009). Scaling factor optimization methods select relevant variables using scaling factors, which are hyper-parameters adjusted by model selection (Weston *et al*, 2000; Jebara and Jaakkola, 2000; Tipping, 2001; Grandvalet and Canu, 2002; Chapelle *et al*, 2002; Maldonado *et al*, 2011). Direct optimization methods include a penalized term that measures the number of selected variables in the optimization problem used for training of a classifier (Bradley and Mangasarian, 1998; Weston *et al*, 2003; Bi *et al*, 2003; Fung and Mangasarian, 2004; Zhou *et al*, 2002; Zhu *et al*, 2003; Mangasarian, 1999, 2006; Zou, 2007; Zou and Hastie, 2005; Wang *et al*, 2006, 2008).

For variable selection in multi-class classification, filter methods can be naturally extended to multi-class cases and can also deal directly with them. However, some filter methods decompose a multi-class classification problem into several binary classification problems and combine the variable scores obtained for each of them (Forman, 2004; You and Li, 2011; Rajapakse and Mundra, 2013). Wrapper methods, in order to score subsets of variables, only need to use a classifier that can handle a multi-class case.

The extension of an embedded method to a multi-class case is, compared with filter and wrapper methods, much less trivial. Although there are many multi-class classification problems in practice, many embedded methods have been suggested for binary classification. Many algorithms for multi-class classification decompose a multi-class classification problem into a set of multiple binary classification problems (Clark and Boswell, 1991; Anand *et al*, 1995; Debnath *et al*, 2004) and combine the outputs of the binary classifiers to construct a multi-class classifier (Friedman, 1996; Hastie and Tibshirani, 1997; Hullermeier and Vanderlooy, 2010). Similarly, multiple variable selections for binary classification problems also can be substituted for variable selection for multi-class classification. Among such embedded methods, some select different variable subsets for each binary classifier using a variable selection method for binary classification (Veenman and Bolck, 2011; Ramaswamy *et al*, 2001; Chai and Domeniconi, 2004). Other methods compute the selection criteria values of all of the variables for each of the binary classifiers and select, by combining those values, a common set of variables for all of the binary classifiers (Chen *et al*, 2006; Duan *et al*, 2007; Chapelle and Keerthi, 2008; Liu *et al*, 2008; Shieh and Yang, 2008; Zhou and Tuck, 2007).

However, there are a number of drawbacks associated with embedded methods that consider a multi-class classification problem as multiple binary classification problems (Wang and Shen, 2007b). First, when a binary classification becomes highly unbalanced with small examples in one class, it is easy to ignore the small class. If this occurs, the relevant variables for the ignored class also are ignored. Second, even though certain variables might be relevant only to one binary classification, they can remain in the multi-class classifier, which degrades the classification performance. Moreover, they cannot capture correlations between different classes (Crammer and Singer, 2002). To overcome these limitations, it is necessary to perform variable selection by treating multiple classes jointly in multi-class classification problems.

There are several embedded methods that simultaneously take all classes into consideration in the variable selection process. Decision trees, for example (Quinlan, 1986, 1993; Breiman *et al*, 1984), which include algorithms for selection of variables during the classification process, can handle multi-class classification problems. For binary classification, Guyon *et al*, (2002) proposed SVM-RFE (support vector machine-recursive feature elimination) to recursively train an SVM classifier and eliminate variables according to their weights. A multi-class extension of SVM-RFE that directly handles multiple classes also has been proposed for variable selection in multi-class classification (Zhou and Tuck, 2007; Zhao and Yand, 2010). Additionally, there are several direct optimization methods with a regularization penalty term for all classes in which variables are naturally selected for multiple classes without any additional selection process (Wang and Shen, 2006, 2007a; Weston *et al*, 2003; Zhang *et al*, 2008; Li and Jia, 2010). Examples include the ℓ_1-norm penalty (the lasso penalty) (Wang and Shen, 2006, 2007a), the ℓ_0-norm penalty (Weston *et al*, 2003), the super-norm penalty (Zhang *et al*, 2008), and the elastic-net penalty, the latter being a mixture of the ℓ_2-norm and the ℓ_1-norm penalties (Li and Jia, 2010). However, most of them are limited in that they are applicable only to linear classifiers.

In this paper, we propose several variable selection methods for multi-class classification using a signomial function. Hwang et al. (2013) developed embedded variable selection methods for binary classification using the signomial classification method proposed by Lee et al. (2014), but these methods cannot be naturally extended to the multi-class case. We attempt to find an optimal variable subset by taking all classes into consideration in multi-class classification problems considering the nonlinear interactions of variables. To do this, we introduce a ℓ_1-norm regularization function that measures the number of selected variables. Also, we impose relative-importance weights on different variables. The proposed methods select variables suitable for predicting the output and automatically determine the number of variables to be selected. With the selected variables, they naturally obtain the resulting classifiers without any additional learning process.

The remainder of this paper is organized as follows. In Section 2, we review related studies. We describe variable selection for multi-class classification using a signomial function in Section 3. Section 4 develops a multi-class variable selection method using a ℓ_1-norm regularization function and then proposes two adaptive parameters to apply different importance weights for different input variables. Computational experiments are reported in Section 5, and concluding remarks are given in the final section.

2. Related studies

In this section, we provide a description of several variable selection methods, including two multi-class feature scoring methods, namely Chi-squared (CHI) and Information gain (IG), as well as one multi-class variable selection method based on recursive feature elimination.

2.1. Multi-class feature scoring methods

CHI and IG are filter methods that score variables based on a certain criterion. Yang and Pedersen (1997) and Forman (2003) conducted comparative studies of filter methods and reported that CHI and IG performed effectively. If input variables have continuous variables, CHI and IG need to convert them to discretized variables by a discretization method (Yang and Webb, 2001; Fayyad and Irani, 1993).

2.1.1. Chi-square method (CHI) CHI (Yang and Pedersen, 1997; Forman, 2003) measures the lack of independence between each variable and class label by calculating the χ^2 statistic. If the variable X and the classes are independent, the χ^2 statistic has a natural value of zero. The χ^2 statistic of variable X is defined

$$\chi^2(X) = \sum_i \sum_j \frac{(O_{ij} - E_{ij})^2}{E_{ij}}, \tag{1}$$

where O_{ij} is the observed frequency count for the ith level of the categorical variable X for class j, and E_{ij} is the expected frequency count for the ith level of the categorical variable X for class j. Variables with high χ^2 values deviate significantly from the independence assumption and therefore are considered relevant.

2.1.2. Information gain method (IG) IG (Yang and Pedersen, 1997; Forman, 2003; Quinlan, 1993) evaluates the merit of a variable by measuring the information gain with respect to the class, which is a correlation measure based on the information-theoretical concept of entropy. The entropy of class variable Y is defined

$$H(Y) = -\sum_p P(y_p)\log(P(y_p)), \tag{2}$$

and the entropy of class variable Y after determining the value of variable X is defined as

$$H(Y|X) = -\sum_q P(x_q) \sum_p P(y_p|x_q)\log(P(y_p|x_q)), \tag{3}$$

where $P(y_p)$ denotes the prior probabilities of all values of Y and $P(y_p|x_q)$ is the posterior probabilities of Y given the values of X. Information gain is the amount of the decrease in entropy of the class when the variable is given vs. absent. It is defined

$$\mathrm{IG}(Y,X) = H(Y) - H(Y|X). \tag{4}$$

If a variable X_1 has a higher information gain than a variable X_2 (i.e., $\mathrm{IG}(Y,X_1) > \mathrm{IG}(Y,X_2)$), the class variable Y is regarded as more correlated to X_1 than to X_2.

2.2. Multi-class support vector machine-recursive feature elimination (MSVM-RFE)

Guyon et al. (2002) proposed an SVM-RFE algorithm that recursively trains an SVM classifier and selects variables in a sequential backward elimination procedure. SVM is a classification algorithm that constructs a decision function $f(\mathbf{x}) = \mathbf{w}^T\phi(\mathbf{x}) + b$ to separate examples $\{\mathbf{x}_i\}_{i=1}^m$ from two classes $\{-1, 1\}$, where decision functions can be obtained by solving the following optimization problem:

$$\min \mathrm{J}(\mathbf{w}, \varepsilon) = \frac{1}{2}||\mathbf{w}||_2 + \mathrm{C}\sum_{\mathrm{i}=1}^{\mathrm{m}} \varepsilon_{\mathrm{i}} \tag{5}$$

$$\begin{aligned} \text{s.t. } & y_i\{\mathbf{w}^T\phi(\mathbf{x}_i) + b\} \geq 1 - \varepsilon_i, \quad \forall i = 1, \ldots, m, \\ & \mathbf{w} \in \mathbb{R}^{|\phi(\mathbf{x})|}, b \in \mathbb{R}, \\ & \varepsilon_i \in \mathbb{R}_+, \quad \forall i = 1, \ldots, m. \end{aligned} \tag{6}$$

In these equations, $\phi(\mathbf{x})$ maps the training data $\mathbf{x}$ to a higher dimensional space, y_i denotes the class label of $\mathbf{x}_i$ and $C > 0$ is the penalty parameter. The dual problem of (5) is

$$\min \frac{1}{2}\sum_{p=1}^{m}\sum_{q=1}^{m} \alpha_p\alpha_q y_p y_q K(\mathbf{x}_p, \mathbf{x}_q) - \sum_{p=1}^{m} \alpha_p \tag{7}$$

$$\begin{aligned} \text{s.t. } & \sum_{p=1}^{m} y_p\alpha_p = 0, \\ & 0 \leq \alpha_i \leq C, \quad \forall i = 1, \ldots, m, \end{aligned} \tag{8}$$

where $K(\mathbf{x}_p, \mathbf{x}_q) = \phi(\mathbf{x}_p)^T\phi(\mathbf{x}_q)$. Then,

$$\mathbf{w} = \sum_{p=1}^{m} \alpha_p y_p \phi(\mathbf{x}_p). \tag{9}$$

SVM-RFE uses, as a variable selection criterion, the change of the objective function $J(\mathbf{w}, \varepsilon)$ caused by removing a variable x_j. The selection criterion for variable x_j is defined, as a second-order term in the Taylor series of the objective function, as

$$c_j = \frac{1}{2}\frac{\partial^2 J}{\partial w_j^2}(\Delta w_j)^2. \tag{10}$$

SVM-RFE removes the variable with the least influence on the weight vector norm $||\mathbf{w}||_2$; the selection criterion can thus be written

$$
\begin{aligned}
c_j =& \left| \|\mathbf{w}\|_2 - \|\mathbf{w}^{(-j)}\|_2 \right| \\
=& \frac{1}{2} \left| \sum_{p=1}^{m} \sum_{q=1}^{m} \alpha_p^* \alpha_q^* y_p y_q K(\mathbf{x}_p, \mathbf{x}_q) \right. \\
& \left. - \sum_{p=1}^{m} \sum_{q=1}^{m} \alpha_p^{*(-j)} \alpha_q^{*(-j)} y_p y_q K^{(-j)}(\mathbf{x}_p, \mathbf{x}_q) \right|,
\end{aligned} \tag{11}
$$

where α^* is the optimal solution of (7) and where the notation $(-j)$ indicates that the variable x_j has been removed. To reduce the computational burden, α_p^* is assumed to be unchanged (i.e., $\alpha_p^* = \alpha_p^{*(-j)}$). At each recursive step, SVM-RFE trains an SVM classifier and computes c_j for the remaining variables, after which it eliminates the variable with the minimum c_j. This elimination procedure is repeated until only a single variable remains.

SVM-RFE has been extended for variable selection in multi-class classification (Zhou and Tuck, 2007; Zhao and Yand, 2010; Shieh and Yang, 2008; Duan *et al*, 2007). To deal with multiple classes, multi-class problems can be decomposed into several binary classification problems (Zhou and Tuck, 2007; Shieh and Yang, 2008; Duan *et al*, 2007). Assuming that all classes equally contribute to the classification, the variable that simultaneously minimizes all of the variable selection criteria of binary classification problems is removed. In this paper, as the variable selection criterion, we use the summation of the variable selection criteria of the one-against-all SVM using a Gaussian kernel (Zhou and Tuck, 2007; Shieh and Yang, 2008).

3. Variable selection for multi-class classification using signomial function

Let $\mathbf{x} = (x_1, \ldots, x_n)$ 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 $\mathbf{d} = (d_1, \ldots, d_n)$ is a real vector. Then, the signomial function of $\mathbf{x}$ is defined

$$
f(\mathbf{x}) = \sum_{\mathbf{d} \in D} w_{\mathbf{d}} g_{\mathbf{d}}(\mathbf{x}) + b, \tag{12}
$$

where $b \in \mathbb{R}$, $w_{\mathbf{d}} \in \mathbb{R}, \forall \mathbf{d} \in D$, and where D is a finite subset of $\mathbb{R}^n$ such that $\mathbf{0} \notin D$. If $D = \{\mathbf{d} \in \mathbb{Z}_+^n : 1 \leq \sum_{j=1}^{n} d_j \leq k\}$ for a positive integer k, then $f(\mathbf{x})$ is a polynomial function of a degree less than or equal to k.

In this paper, we consider the set D, the set of exponents $\mathbf{d}$, which is prespecified by the four parameters, $d_{\min}, d_{\max}, L$, and T as

$$
D = \left\{ \mathbf{d} \in \mathbb{R}^n : d_{\min} \leq d_j \leq d_{\max}, j = 1, \ldots, n, \sum_{i=1}^{n} |d_i| \leq L, T\mathbf{d} \in \mathbb{Z}^n \right\}, \tag{13}
$$

where $T > 0$ and $L > 0$. From the above definition of D, each exponent d_j takes a value on an equally spaced grid that is obtained by discretizing the closed interval $[d_{\min}, d_{\max}]$. Here, T controls the level of granularity of the grid, so that each d_j is an integer multiple of $1 / T$. If we set $d_{\min} = 0$, $d_{\max} = 1$, $T = 1$, and $L = k$ for some $k \in \mathbb{Z}_+$, then $f(\mathbf{x})$ is a polynomial function of a degree less than or equal to k. If $T > 1$ at the above parameters, exponents can take fractional values. In Table 1 we show a number of example signomial functions that can be obtained by changing the parameters of (13).

We consider a given set $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$ of m training examples $\mathbf{x}_i$ where $\mathbf{x}_i \in \mathbb{R}_{++}^n$ consists of n input variables. Suppose that each example $\mathbf{x}_i$ belongs to class y_i, $y_i \in K := \{1, \ldots, c\}, c > 2$, where c is the number of classes and X^k is a set of examples belonging to class k, with $k \in K$ satisfying $\bigcup_{k \in K} X^k = X$. We attempt to select a subset of the n input variables, which is useful for predicting the class of a new example using a signomial function (12).

Hwang et al. (2015), using a signomial function, developed multi-class signomial classification (MSC) methods. MSCs use $\mathbf{f} = \{f_1(\mathbf{x}), \ldots, f_c(\mathbf{x})\}$ as a decision function vector, where each $f_k(\mathbf{x})$ takes the form of a signomial function (12) and represents the strength of the evidence that an example $\mathbf{x}$ belongs to class k, such that $k \in K$. A multi-class signomial classifier $f_M(\mathbf{x})$ is defined by $\mathbf{f}$ as

$$
f_M(\mathbf{x}) = \underset{k \in K}{\operatorname{argmax}} \left\{ f_k(\mathbf{x}) = \sum_{\mathbf{d} \in D^k} w_{\mathbf{d}}^k g_{\mathbf{d}}(\mathbf{x}) + b^k \right\}. \tag{14}
$$

Here, $f_M(\mathbf{x})$ assigns an example $\mathbf{x}$ to the class having the largest $f_k(\mathbf{x})$. If there are more than one k with a maximum value, we randomly select one of them.

To obtain the decision function vector $\mathbf{f}$, MSCs minimize the regularized functional

$$
F(\mathbf{f}) = R(\mathbf{w}) + C_1 L(\mathbf{f}, \mathbf{y}), \tag{15}
$$

Table 1 Examples of signomial functions

$d_{\min}$	$d_{\max}$	T	L	*Example function*
0	3	1	3	$f(\mathbf{x}) = w_1 x_1^3 + w_2 x_1^2 x_4^1 + w_3 x_2^1 x_5^1$
0	3	2	3	$f(\mathbf{x}) = w_1 x_1^{1.5} + w_2 x_1^{0.5} x_4^{2.5} + w_3 x_2^1 x_5^{1.5}$
−3	3	4	3	$f(\mathbf{x}) = w_1 x_1^{1.25} + w_2 x_1^{-1.5} x_4^{1.25} x_5^{0.25} + w_3 x_2^{-1.25} x_5^{1.75}$
−1	1	10	1	$f(\mathbf{x}) = w_1 x_1^1 + w_2 x_1^{-0.5} x_5^{0.1} + w_3 x_2^{-0.5} x_4^{0.2} x_5^{-0.3}$

where $C_1 > 0$ is the penalty parameter, $L(\mathbf{f},\mathbf{y}) = \sum_{i=1}^{m} \sum_{k\in K\setminus\{y_i\}} |1 - f_{y_i}(\mathbf{x}_i) + f_k(\mathbf{x}_i)|_+$, which is a hinge loss function, and R is a regularization function, $R(\mathbf{w}) = \sum_{k\in K} \|\mathbf{w}^k\|_1$ or $R(\mathbf{w}) = \sum_{k\in K} \|\mathbf{w}^k\|_0$. When $\sum_{k\in K} \|\mathbf{w}^k\|_1$ is adopted, MSC is referred to as the ℓ_1-norm method for multi-class signomial classification (ℓ_1-MSC). MSCs minimize $R(\mathbf{w})$ to minimize the number of signomial terms in the resulting multi-class classifier. For variable selection, however, it is necessary to directly minimize the number of input variables rather than that of decision function terms.

Let $\boldsymbol{\sigma} \in \mathbb{R}^n_+$ be a vector of indicator variables, where, if $\sigma_j > 0$, the jth input variable is selected, while otherwise it is not. Let $S(\boldsymbol{\sigma})$ be a regularization function that measures the number of selected variables. For variable selection purposes, we use ℓ_1-MSC and add $S(\boldsymbol{\sigma})$ to the objective function (15). This is formulated as

$$\min_{\sigma,\mathbf{w},b} S(\boldsymbol{\sigma}) + C_2 F(\mathbf{f}), \tag{16}$$

where $C_2 > 0$ is the parameter that controls the number of selected variables. In the next section, we present our approach to solve the variable selection problem (16).

But before closing this section, we need to note how categorical variables can be handled by our approach. If there are categorical input variables, we can handle them by introducing binary variables (dummy variables) for each categorical variable. For an m-category variable x_j, we first introduce m binary variables x_{jk} for $k = 1,\ldots,m$. If the value of x_j belongs to the kth-category, we set the binary variable x_{jk} to 1 and the others to zero. Additionally, we define additional m binary variables $\bar{x}_{jk}$ such that $\bar{x}_{jk} = 1 - x_{jk}$ for $k = 1,\ldots,m$. Then, to ensure that each data point is positive, we add a small positive value, for example $\epsilon = 10^{-6}$, to the value of each dummy variable. The classifier obtained by ℓ_1-MSC with parameters $d_{\min} = 0$, $d_{\max} = 1$, $T = 1$, and $L = q$ for some $q \in \mathbb{Z}_+$ will be a special type of signomial function that can be used for interpretation purposes.

For instance, let us suppose we have a 4-category variable x_j that takes a value from among A, B, C, and D. We first define 8 binary variables x_{jk} and $\bar{x}_{jk}$ for $k = 1,\ldots,4$ as explained above. Then, the values A, B, C, and D are represented as four 8-dimensional binary vectors (1, 0, 0, 0, 0, 1, 1, 1), (0, 1, 0, 0, 1, 0, 1, 1), (0, 0, 1, 0, 1, 1, 0, 1) and (0, 0, 0, 1, 1, 1, 1, 0), respectively. After adding a small positive value to the value of each dummy variable, the resulting classifier for some specific class might contain terms such as $-2.3x_{j1}\bar{x}_{j2}$, $4\bar{x}_{j1}x_{j4}$, and so on. We can see that the first term has a negative impact on a point being a member of the specific class, while the second has a positive impact. The meaning of the first term is that if $x_j = \mathrm{A}$ and $x_j \neq \mathrm{B}$, the corresponding point could be a member of one of the other classes. The meaning of the second term can be interpreted in a similar way.

4. Multi-class variable selection method using ℓ_1-norm regularization function

We develop variable selection methods for multi-class classification using a signomial function. We define a regularization function in (16) for variable selection in multi-class classification and then propose that different importance weights be imposed on different input variables.

4.1. ℓ_1-norm regularization function

We propose a ℓ_1-norm regularization function and a variable selection method using the ℓ_1-norm regularization function for multi-class classification. To construct a multi-class classifier, we find the decision function vector $\mathbf{f}$ by minimizing the objective function (15). This can be formulated as

[Problem 1]

$$\min \sum_{k\in K} ||\mathbf{w}^k||_1 + C \sum_{k\in K} \sum_{i\in X^k} \sum_{l\in K\setminus\{k\}} \varepsilon_i^{kl} \tag{17}$$

$$\begin{aligned}
\text{s.t.} \ & \sum_{\mathbf{d}\in D^k} w^k_{\mathbf{d}} g_{\mathbf{d}}(\mathbf{x}_i) + b^k - \left\{ \sum_{\mathbf{d}\in D^l} w^l_{\mathbf{d}} g_{\mathbf{d}}(\mathbf{x}_i) + b^l \right\} + \varepsilon_i^{kl} \geq 1, \\
& \forall l \in K\setminus\{k\}, i \in X^k, k \in K, \\
& \mathbf{w}^k \in \mathbb{R}^{|D^k|}, b^k \in \mathbb{R}, \quad \forall k \in K, \\
& \varepsilon_i^{kl} \in \mathbb{R}_+, \quad \forall l \in K\setminus\{k\}, i \in X^k, k \in K,
\end{aligned} \tag{18}$$

where C is the penalty parameter and D^k is the set of exponents for class k, with $k \in K$ defined by (13). The ℓ_1-norm $||\mathbf{w}^k||_1$ is defined $\sum_{\mathbf{d}\in D^k} |w^k_{\mathbf{d}}|$ for $k \in K$, and ε_i^{kl} is the misclassification error, which is positive if data $\mathbf{x}_i$ of class k are misclassified by the classifier $f_l(\mathbf{x})$ for $k, l \in K$. The objective function (17) is to minimize $\sum_{k\in K} ||\mathbf{w}^k||_1$ and the sum of any misclassification errors. The parameter C is a positive real number that controls the relative importance of the training error to the ℓ_1-norm of $\mathbf{w}^k$.

By replacing w^k_d with $w^{k+}_d - w^{k-}_d$ and $|w^k_{\mathbf{d}}|$ with $w^{k+}_{\mathbf{d}} + w^{k-}_{\mathbf{d}}$, where $w^{k+}_d \geq 0$ and $w^{k-}_d \geq 0$, we convert Problem 1 to a linear programming (LP) problem that we call Problem 2. The objective function of Problem 2 is

$$\min \sum_{k\in K} \sum_{\mathbf{d}\in D^k} (w^{k+}_{\mathbf{d}} + w^{k-}_{\mathbf{d}}) + C \sum_{k\in K} \sum_{i\in X^k} \sum_{l\in K\setminus\{k\}} \varepsilon_i^{kl}. \tag{19}$$

The exponent set D^k can be exponentially large, which makes Problem 2 practically intractable. We, however, can generate exponents $\mathbf{d} \in D^k$ as needed rather than in advance using a column generation algorithm, and can therefore solve Problem 2 efficiently (Bertsimas and Tsitsiklis 1997).

Suppose that $(w^{k+}_{\mathbf{d}} - w^{k-}_{\mathbf{d}}) \neq 0$ for any $\mathbf{d} \in D^k$ with $d_j \neq 0$, $k \in k$. This means that the terms with the jth variable contribute $(w_{\mathbf{d}}^{k+} - w_{\mathbf{d}}^{k-})$ to the resulting classifier $f_k(\mathbf{x})$. The jth variable appears in the resulting classifier $f_k(\mathbf{x})$ and has been selected. On the other hand, if the jth variable makes no contribution to the resulting classifier $f_k(\mathbf{x})$ for all $k \in K$, $(w_{\mathbf{d}}^{k+} - w_{\mathbf{d}}^{k-}) = 0$ for all $\mathbf{d} \in D^k$ with $d_j \neq 0$, $k \in K$, and the jth variable is not selected for any $f_k(\mathbf{x})$, $k \in K$. Let $\eta_j \in \mathbb{R}_+$ be a variable that measures the contribution of the jth variable to the resulting classifier $f_k(\mathbf{x})$ for all $k \in K$, $\sum_{k\in K}\sum_{\mathbf{d}\in D^k, d_j \neq 0}(w_{\mathbf{d}}^{k+} + w_{\mathbf{d}}^{k-})$, for $j = 1, \ldots, n$. If $\eta_j > 0$, the jth input variable is selected; otherwise, it is not.

We define a regularization function $S(\boldsymbol{\eta})$ as the ℓ_1-norm of $\boldsymbol{\eta}$, $\sum_{j=1}^{n}\eta_j$. Minimizing the ℓ_1-norm can force $\boldsymbol{\eta}$ to be sparse (Zhu *et al*, 2003; Huang *et al*, 2009). We add the ℓ_1-norm regularization function to the objective function (19) and remove the regularization function $\sum_{k\in K}\sum_{\mathbf{d}\in D^k}(w_{\mathbf{d}}^{k+} + w_{\mathbf{d}}^{k-})$ from the objective function (19). Minimizing $||\boldsymbol{\eta}||_1$ can perform regularization, and removal of the regularization term can alleviate the burden of parameter tuning. We can then construct the following problem:

[Problem 3]

$$\min \sum_{j=1}^{n}\eta_j + C\sum_{k\in K}\sum_{i\in X^k}\sum_{l\in K\backslash\{k\}}\varepsilon_i^{kl} \tag{20}$$

$$\begin{aligned}\text{s.t.} &\sum_{\mathbf{d}\in D^k}(w_{\mathbf{d}}^{k+} - w_{\mathbf{d}}^{k-})g_{\mathbf{d}}(\mathbf{x}_i) + b^k \\ &- \left\{\sum_{\mathbf{d}\in D^l}(w_{\mathbf{d}}^{l+} - w_{\mathbf{d}}^{l-})g_{\mathbf{d}}(\mathbf{x}_i) + b^l\right\} + \varepsilon_i^{kl} \geq 1, \\ &\forall l \in K\backslash\{k\}, i \in X^k, k \in K,\end{aligned} \tag{21}$$

$$\sum_{k\in K}\sum_{\mathbf{d}\in D^k, d_j\neq 0}(w_{\mathbf{d}}^{k+} + w_{\mathbf{d}}^{k-}) \leq \eta_j, \quad \forall j = 1,\ldots,n, \tag{22}$$

$$\begin{aligned}&\mathbf{w}^{k+}, \mathbf{w}^{k-} \in \mathbb{R}_+^{|D^k|}, b^k \in \mathbb{R}, && \forall k \in K, \\ &\varepsilon_i^{kl} \in \mathbb{R}_+, && \forall l \in K\backslash\{k\}, i \in X^k, k \in K, \\ &\eta_j \in \mathbb{R}_+, && \forall j = 1,\ldots,n.\end{aligned}$$

Instead of enumerating all of the elements of the exponent set D^k, $k \in K$, we use only a subset of exponents of a limited size as D^k can be exponentially large. To generate a subset of exponents for class k for all $k \in K$, we set the penalty parameter C of Problem 2 to a large value (e.g., 10^3) and then solve Problem 2 using a column generation algorithm (Bertsimas and Tsitsiklis 1997) (see (Hwang *et al*, 2015)). This generates as many profitable exponents as possible. Let $\hat{D}^k \subset D^k$ be a subset of such exponents for class k, $k \in K$. By replacing D^k with $\hat{D}^k$, we construct the restricted problem of Problem 3, which we call Problem 4. Problem 4 can be solved using a standard LP technique. After solving Problem 4, we obtain an optimal subset of the n input variables and the multi-class signomial classifier with the selected variables.

Let $(\hat{\mathbf{w}}^+, \hat{\mathbf{w}}^-, \hat{\mathbf{b}}, \hat{\boldsymbol{\eta}})$ be an optimal solution to Problem 4. If $\hat{\eta}_j > 0$, the jth variable is selected in an optimal subset of the variables; otherwise, it is not. The multi-class signomial classifier is obtained as

$$f_M(\mathbf{x}) = \underset{k\in K}{\operatorname{argmax}}\left(f_k(\mathbf{x}) = \sum_{\mathbf{d}\in\hat{D}^k}(\hat{w}_{\mathbf{d}}^{k+} - \hat{w}_{\mathbf{d}}^{k-})g_{\mathbf{d}}(\mathbf{x}) + \hat{b}^k\right). \tag{23}$$

Here, $\mathbf{x}$ is classified as belonging in class k if $f_M(\mathbf{x}) = k$. If there are more than one k with a maximum value, we randomly select one of them. We refer to the above variable selection method as the ℓ_1-norm multi-class variable selection method (ℓ_1-MVS). An overview of ℓ_1-MVS is provided in Figure 1.

4.2. Adaptive parameters

In the previous subsection, the ℓ_1-norm regularization function is used to select an optimal subset of n input variables. The regularization function gives an equal weight of 1 to each of the n input variables. There might, however, be different importance weights for each of the n input variables. We propose two adaptive parameters to apply different importance weights for different input variables. Variables with small weights can be selected more easily than those with large parameters.

We introduce two adaptive parameters, the positive real numbers τ_j and τ_{kj}. Here, τ_j represents the weight of the jth input variable for $j = 1, \ldots, n$. We impose τ_j on η_j for $j = 1, \ldots, n$ to apply the different importance weights to each of the n input variables. We modify Problem 3 to employ τ_j as follows:

[Problem 5]

$$\min \sum_{j=1}^{n}\tau_j\eta_j + C\sum_{k\in K}\sum_{i\in X^k}\sum_{l\in K\backslash\{k\}}\varepsilon_i^{kl} \tag{24}$$

$$\begin{aligned}\text{s.t.} &\sum_{\mathbf{d}\in D^k}(w_{\mathbf{d}}^{k+} - w_{\mathbf{d}}^{k-})g_{\mathbf{d}}(\mathbf{x}_i) + b^k \\ &- \left\{\sum_{\mathbf{d}\in D^l}(w_{\mathbf{d}}^{l+} - w_{\mathbf{d}}^{l-})g_{\mathbf{d}}(\mathbf{x}_i) + b^l\right\} + \varepsilon_i^{kl} \geq 1, \\ &\forall l \in K\backslash\{k\}, i \in X^k, k \in K,\end{aligned} \tag{25}$$

$$\sum_{k\in K}\sum_{\mathbf{d}\in D^k, d_j\neq 0}(w_{\mathbf{d}}^{k+} + w_{\mathbf{d}}^{k-}) \leq \eta_j, \quad \forall j = 1,\ldots,n, \tag{26}$$

$$\begin{aligned}&\mathbf{w}^{k+}, \mathbf{w}^{k-} \in \mathbb{R}_+^{|D^k|}, b^k \in \mathbb{R}, && \forall k \in K, \\ &\varepsilon_i^{kl} \in \mathbb{R}_+, && \forall l \in K\backslash\{k\}, i \in X^k, k \in K, \\ &\eta_j \in \mathbb{R}_+, && \forall j = 1,\ldots,n.\end{aligned}$$

Also, it needs to be noted that τ_{kj} represents the weight of the jth input variable of class k for $j = 1, \ldots, n$ and $k \in K$. To employ τ_{kj}, we define $\eta_j^k \in \mathbb{R}_+$ as a variable that measures the

```
Initialize: $\hat{D}^k := \emptyset$ for all $k \in K$, and $V := \{1, \ldots, n\}$.
Exponent generating procedure:
  Solve the Problem 2 with a large value (e.g., $10^3$).
  $\hat{D}^k$ for all $k \in K \leftarrow$ the generate profitable exponents to the Problem 2.
Variable selection procedure:
  Solve the Problem 4.
  $(\hat{\mathbf{w}}^+, \hat{\mathbf{w}}^-, \hat{\mathbf{b}}, \hat{\boldsymbol{\eta}}) \leftarrow$ the optimal solution to the Problem 3.
  for $j \in 1, \ldots, n$ do
    if $\hat{\eta}_j = 0$ then
      $V := V \setminus \{j\}$.
    end-if
  end-do
  $f_M(\mathbf{x}) := \underset{k \in K}{\operatorname{argmax}}\Big(f_k(\mathbf{x}) = \sum_{\mathbf{d} \in \hat{D}^k} (\hat{w}_{\mathbf{d}}^{k+} - \hat{w}_{\mathbf{d}}^{k-}) g_{\mathbf{d}}(\mathbf{x}) + \hat{b}^k\Big)$.
Return the variable subset $V$ and the classifier $f_M(\mathbf{x})$.
```

Figure 1 ℓ_1-norm multi-class variable selection method (ℓ_1-MVS).

contribution of the jth variable to the resulting classifier $f_k(\mathbf{x})$ for $k \in K$, $\sum_{\mathbf{d} \in D^k, d_j \neq 0} (w_{\mathbf{d}}^{k+} + w_{\mathbf{d}}^{k-})$, for $j = 1, \ldots, n$. If $\eta_j^k > 0$, the jth input variable is selected in the classifier $f_k(\mathbf{x})$; otherwise, it is not. By replacing constraint (22) of Problem 3 with the constraint

$$\sum_{\mathbf{d} \in D^k, d_j \neq 0} (w_{\mathbf{d}}^{k+} + w_{\mathbf{d}}^{k-}) \leq \eta_j^k, \quad \forall j = 1, \ldots, n, k \in K$$

and modifying its objective function, we construct Problem 6. The objective function of Problem 6 is

$$\sum_{k \in K} \sum_{j=1}^{n} \tau_{kj} \eta_j^k + C \sum_{k \in K} \sum_{i \in X^k} \sum_{l \in K \setminus \{k\}} \varepsilon_i^{kl}.$$

Problems 5 and 6 can be solved by ℓ_1-MVS, as explained in Section 4.1. We refer to these variable selection methods as adaptive multi-class variable selection I (adaptive MVS$_{\mathrm{I}}$) and II (adaptive MVS$_{\mathrm{II}}$), respectively.

Let $(\hat{\mathbf{w}}^+, \hat{\mathbf{w}}^-)$ be an optimal solution to Problem 2 and $\bar{w}_j^k := \sum_{\mathbf{d} \in D^k, d_j \neq 0} (\hat{w}_{\mathbf{d}}^{k+} + \hat{w}_{\mathbf{d}}^{k-})$. In our implementation, we set τ_j and τ_{kj} as

$$\tau_j = 1 - \frac{\sum_{k \in K} \bar{w}_j^k}{\sum_{k \in K} \sum_{j=1}^{n} \bar{w}_j^k}, \qquad \forall j = 1, \ldots, n. \tag{27}$$

$$\tau_{kj} = 1 - \frac{\bar{w}_j^k}{\sum_{j=1}^{n} \bar{w}_j^k}, \qquad \forall j = 1, \ldots, n, k \in K. \tag{28}$$

5. Computational experiments

5.1. Computational setting

We conducted experiments on several multi-class classification problems from the Statlog collection (Michie *et al*, 1994) and from the UCI Repository of machine learning databases (Bache and Lichman 2013). We chose the Image Segmentation (IS) and DNA data sets from the Statlog collection, and the Cardiotocography (CARD), Multiple Features (MF) and Gas Sensor Array Drift (GAS) data sets from the UCI Repository. Table 2 provides descriptions of these data sets.

The performances of the proposed methods were compared with combinations of three variable selection methods and five classification methods. The tested methods are presented in Table 3. The variable selection methods are two multi-class feature scoring methods, namely Chi-squared (CHI) (Yang and Pedersen, 1997; Forman, 2003) and Information Gain (IG) (Yang and Pedersen, 1997; Forman, 2003; Quinlan, 1993), along with the multi-class support vector machine-recursive feature elimination (MSVM-RFE) (Zhou and Tuck, 2007; Shieh and Yang, 2008).

The classification methods are k-nearest neighbor (kNN) (Cover and Hart, 1967; Bay, 1998), classification and regression tree (CART) (Breiman *et al*, 1984), boosted classification tree (BCT) (Freund and Schapire, 1997) and two multi-class SVMs, in this case Weston and Watkins's multi-class SVM (WW) (Weston and Watkins, 1999) and Crammer and Singer's multi-class SVM (CS) (Crammer and Singer 2002). BCT uses the AdaBoost.M2 algorithm (Freund and Schapire, 1997), a multi-class extension of AdaBoost, with weak learners based on classification trees with default parameters. The default parameters are as follows: the maximal number of branch nodes to be split is 1, the minimum size of leaf in order to

Table 2 Data sets used in experiments

Data set	*#Classes*	*#Attributes*	*#Instances*
IS	7	19	2310
CARD	3	21	2126
DNA	3	180	3186
MF	10	649	2000
GAS	6	128	13910

Table 3 Tested methods

Method	*Variable selection*	*Classification*
Multi-class feature scoring methods	CHI	WW CS kNN CART BCT
	IG	WW CS kNN CART BCT
Recursive feature elimination method	MSVM-RFE	WW CS
Proposed methods	ℓ_1-MVS adaptive MVS$_{\mathrm{I}}$ adaptive MVS$_{\mathrm{II}}$	

Table 4 CHI performance results: Average classification accuracy (%) and standard deviation for test sets, average time (s) and standard deviation to select variables and train classifiers, and average number of selected variables

Data set	*CHI*					
	CART			*BCT*		
	TestAcc	*SelTrnT*	*SelVar*	*TestAcc*	*SelTrnT*	*SelVar*
IS	93.51 ± 1.44	9.92 ± 5.9	14.80	90.03 ± 1.79	14.17 ± 6.42	14.80
CARD	91.29 ± 1.79	0.28 ± 0.03	15.40	88.67 ± 1.44	2.9 ± 1.17	15.40
DNA	90.58 ± 1.55	2.52 ± 0.13	82.80	88.03 ± 1.38	4.48 ± 1.83	82.80
MF	89.56 ± 1.77	18.77 ± 1.77	131.65	74.61 ± 13.25	66.83 ± 12.87	131.65
GAS	95.51 ± 0.55	10.08 ± 0.48	51.10	69.02 ± 0.87	140.02 ± 35.44	51.10

Data set	*CHI*								
	WW			*CS*			*kNN*		
	TestAcc	*SelTrnT*	*SelVar*	*TestAcc*	*SelTrnT*	*SelVar*	*TestAcc*	*SelTrnT*	*SelVar*
IS	95.24 ± 1.21	1.06 ± 0.27	14.08	95.73 ± 1.41	0.16 ± 0.00	14.80	94.84 ± 1.12	13.08 ± 5.73	14.80
CARD	89.67 ± 1.14	0.46 ± 0.05	15.40	89.27 ± 1.84	0.55 ± 0.10	15.40	88.51 ± 1.37	2.65 ± 0.58	15.40
DNA	95.62 ± 1.06	2.66 ± 0.16	82.80	95.75 ± 0.88	3.12 ± 0.53	82.80	77.88 ± 2.28	4.46 ± 0.77	82.80
MF	92.75 ± 2.55	19.48 ± 1.82	131.65	91.35 ± 3.82	24.52 ± 4.36	131.65	85.99 ± 4.32	21.86 ± 2.05	131.65
GAS	98.21 ± 0.64	33.79 ± 16.98	51.10	98.36 ± 0.68	2559.46 ± 4284.82	51.10	98.12 ± 0.32	60.76 ± 17.94	51.10

We selected as many variables as the number of variables selected by ℓ_1-MVS

Table 5 IG performance results: Average classification accuracy (%) and standard deviation for test sets, average time (s) and standard deviation to select variables and train classifiers, and average number of selected variables

Data set	*IG*					
	CART			*BCT*		
	TestAcc	*SelTrnT*	*SelVar*	*TestAcc*	*SelTrnT*	*SelVar*
IS	93.30 ± 1.29	0.36 ± 0.03	14.80	90.03 ± 1.79	4.70 ± 1.22	14.80
CARD	91.21 ± 1.59	0.32 ± 0.04	15.40	88.64 ± 1.46	2.90 ± 1.15	15.40
DNA	90.59 ± 1.51	2.98 ± 0.19	82.80	88.03 ± 1.38	4.94 ± 1.90	82.80
MF	90.49 ± 1.52	18.49 ± 1.29	131.65	82.91 ± 7.18	101.74 ± 16.47	131.65
GAS	95.39 ± 0.51	10.23 ± 1.06	51.10	68.91 ± 0.65	140.06 ± 39.46	51.10

Data set	*IG*								
	WW			*CS*			*kNN*		
	TestAcc	*SelTrnT*	*SelVar*	*TestAcc*	*SelTrnT*	*SelVar*	*TestAcc*	*SelTrnT*	*SelVar*
IS	95.23 ± 1.16	0.80 ± 0.27	14.80	95.51 ± 1.41	1.10 ± 0.42	14.80	95.26 ± 1.11	1.66 ± 0.08	14.80
CARD	89.55 ± 1.13	0.50 ± 0.07	15.40	89.16 ± 1.77	0.58 ± 0.12	15.40	88.45 ± 1.29	2.90 ± 0.50	15.40
DNA	95.62 ± 1.06	3.19 ± 0.23	82.80	95.75 ± 0.88	3.54 ± 0.58	82.80	77.88 ± 2.28	5.27 ± 0.91	82.80
MF	96.08 ± 1.12	18.85 ± 1.30	131.65	96.20 ± 1.03	19.84 ± 1.45	131.65	92.85 ± 1.16	21.23 ± 1.34	131.65
GAS	98.40 ± 0.37	34.39 ± 19.58	51.10	98.57 ± 0.29	2300.44 ± 4657.16	51.10	98.19 ± 0.24	51.73 ± 3.97	51.10

We selected as many variables as the number of variables selected by ℓ_1-MVS

obtain deep trees is 1, and the minimum size of parents of each branch node is 2. Because, unlike the proposed methods, the variable selection methods (CHI, IG and MSVM-RFE) cannot give classifiers, the classification methods are additionally required to train classifiers after the variable selection process. We used WW, CS, kNN, CART and CT for the multi-class feature scoring methods (CHI and IG), and WW and CS for MSVM-RFE.

Additional experiments without considering variable selection were conducted on the same data sets to determine the effect of variable selection. These used WW, CS, kNN, CART, BCT and ℓ_1-MSC (Hwang *et al*, 2015).

Here, ℓ_1-MVS, adaptive $\mathrm{MVS_I}$, adaptive $\mathrm{MVS_{II}}$ and ℓ_1-MSC were implemented with the Xpress Mosel language using the linear programming solver provided by the Xpress package (Xpress, 2015). CHI and IG were tested with the R language (Ihaka and Gentleman, 1996). MSVM-RFE was implemented with the SVM-KM Toolbox (Canu *et al*, 2005). WW and CS were implemented in the BSVM software package (Hsu and Lin, 2012) using the decomposition method proposed by Hsu and Lin (2002). We used MATLAB (Matlab, 2010) for kNN, CART and BCT.

Let $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$ be an original data set where $\mathbf{x}_i \in \mathbb{R}^n$ for $i = 1, \ldots, m$ consists of n input variables. Let $\mathrm{Min}_j := \min_{i=1,\ldots,m} x_{ij}$ and $\mathrm{Max}_j := \max_{i=1,\ldots,m} x_{ij}$ for $j = 1, \ldots, n$. For the proposed methods, it was necessary to use translated data within the $[1, \infty)$ range on account of the definition of the signomial function (12). We translated the original data in such a way that if $\mathrm{Min}_j < 0$, $x_{ij} := x_{ij} - \mathrm{Min}_j + 1$; otherwise $x_{ij} := x_{ij} + 1$. We conducted experiments on the original data translated within the $[1, \infty)$ range. Note that for MSVM-RFE, we scaled the GAS data set to the [0, 1] range in such a way that $x_{ij} := (x_{ij} - \mathrm{Min}_j)/(\mathrm{Max}_j - \mathrm{Min}_j)$. The MSVM-RFE needs to calculate an inverse matrix of $K \in \mathbb{R}^{m \times m}$, Gaussian radial basis function kernel $K_{pq} = K(x_p, x_q) := \exp(-\gamma \|x_p - x_q\|^2)$ for $p, q = 1, \ldots, m$, but the attribute ranges of the GAS data set are too variable for calculation of an inverse matrix. Therefore, for the GAS data set, we conducted MSVM-RFE experiments on the [0, 1] scaled data.

We selected model parameters with all of the input variables. 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 20 times with a ratio of 5:3:2 while ensuring that the proportions of the classes were similar in each subset. For various parameter settings, the classifiers were trained on the training set and then evaluated using the corresponding validation set. The model parameters that achieved the highest level of accuracy on the validation set were selected, and the selected parameters were then applied to the corresponding test set to evaluate the performance of the methods.

For the proposed methods and for ℓ_1-MSC, we defined the set $D^k := \{\mathbf{d} \in \mathbb{R}^n : -1 \le d_j \le 1, j = 1, \ldots, n, \sum_{i=1}^{n} |d_i| \le 1, 10\mathbf{d} \in \mathbb{Z}^n\}$ for $k \in K$. Thus, we search exponents in an equal-interval grid having the range of $[-1, 1]$ and the 1 / 10 scale, and choose less than or equal to 10 non-zero exponents of $g_{\mathbf{d}}(\mathbf{x})$, the absolute sum of which is less than or equal to 1. For example, when n is 5, we might find exponents such as $d_1 = (1, 0, 0, 0, 0)$, $d_2 = (-0.5, 0, 0, 0, 0.2)$, and $d_3 = (0, -0.5, 0, 0.2, -0.3)$ and then obtain the resulting classifier $f(\mathbf{x}) = w_1 x_1^1 + w_2 x_1^{-0.5} x_5^{0.2} + w_3 x_2^{-0.5} x_4^{0.2} x_5^{-0.3}$. We tested the proposed methods using seven regularization parameters C: $C = [10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3]$. For MSVM-RFE,

Table 6 MSVM-RFE performance results: Average classification accuracy (%) and standard deviation for test sets, average time (s) and standard deviation to select variables and train classifiers, and average number of selected variables

Data set	*MSVM-RFE*					
	WW			*CS*		
	TestAcc	*SelTrnT*	*SelVar*	*TestAcc*	*SelTrnT*	*SelVar*
IS	95.28 ± 1.30	474.98 ± 63.89	14.80	95.90 ± 1.17	475.13 ± 63.86	14.80
CARD	89.91 ± 1.08	629.08 ± 125.77	15.40	89.43 ± 1.58	629.15 ± 125.77	15.40
DNA	95.65 ± 0.90	2718.39 ± 479.35	82.80	95.63 ± 0.83	2718.65 ± 479.41	82.80
MF	97.26 ± 0.86	1271.26 ± 435.14	131.65	97.21 ± 0.80	1272.07 ± 435.13	131.65
GAS	98.93 ± 0.22	116220.23 ± 13451.53	51.10	98.95 ± 0.18	118281.33 ± 17399.25	51.10

We selected as many variables as the number of variables selected by ℓ_1-MVS

WW and CS, we used a Gaussian radial basis function kernel $K(x_p, x_q) := \exp(-\gamma\|x_p - x_q\|^2)$. These methods were tested using 7×7 combinations of regularization parameters C and the kernel parameters γ: $C = [10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3]$ and $\gamma = [10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3]$. Note that for the MF and GAS data sets, we conducted WW and CS experiments with 11×11 combinations of C and γ: $C = [1, 10^1, \ldots, 10^9, 10^{10}]$ and $\gamma = [10^{-10}, 10^{-9}, \ldots, 10, 1]$. kNN was tested using the different numbers k of nearest neighbors: $k = [1, 2^1, 2^2, \ldots]$. CART was tested using seven different pruning levels p: $p = [0, 1, 2, 3, 4, 5, 6]$. BCT was tested using the different numbers w of weak learners: $w = [1, 2, \ldots, 499, 500]$ except for the GAS data set. For the GAS data set, we tested using the different numbers w' of weak learners: $w' = [200, 400, \ldots, 800, 1000]$.

As performance criteria, we used the average classification accuracy of the test sets, the average time to select variables and train classifiers and the average number of the selected variables in the optimal variable subset. CHI, IG and MSVM-RFE require the setting of the variable number to be selected. Therefore, for these methods, we selected as many variables as the number selected by ℓ_1-MVS and then constructed the resulting classifiers using WW, CS, kNN, CART and BCT.

5.2. Computational results

The computational results are presented in Tables 4, 5, 6 and 7 to 8. The TestAcc denotes the average classification accuracy and standard deviation for test sets, and the SelVar denotes the average number of selected variables. In Tables 4, 5, 6 and 7, the SelTrnT denotes the average time and standard deviation of variable selection and classifier training processes.

To evaluate the performance of the proposed methods, we tested them with three existing variable selection methods (CHI, IG and MSVM-RFE) on the five data sets. Tables 4 and 5 show the performance results of CHI and IG, respectively. After the variable selection processes of CHI and IG, the five classification methods (WW, CS, kNN, CART and BCT) trained classifiers with the selected variables. Table 6 provides the performance results for the MSVM-RFE method. For the MSVM-RFE, WW and CS classification methods were used for training classifiers. The performances of the proposed methods are indicated in Table 7. Table 8 presents the results of additional experiments in which variable selection was not considered with WW, CS, kNN, CART, BCT or ℓ_1-MSC.

The proposed methods yielded competitive or better classification accuracy levels for most of the data sets (with the exception of the CARD data set) as compared with the other twelve methods (combinations of variable selection and classification methods; see Table 3). The comparable or better performance results of the proposed methods are indicated in bold in Table 7, as compared with those of the CHI, IG and MSVM-RFE. The twelve existing methods chose as many variables as the number of variables selected by ℓ_1-MVS and then trained the classifiers with those variables. The classifiers obtained by ℓ_1-MVS gave competitive or better classification accuracy levels than those of the twelve existing methods using the same number of variables for most of the data sets (with the exception of the CARD data set). In other words, ℓ_1-MVS selected variables that are suitable for predicting the output.

CHI, IG and MSVM-RFE can select variables, but they need other classification methods (WW, CS, kNN, CART and BCT) to train classifiers using those variables. The classifiers yielded varying levels of classification accuracy according to the classification method used. It was impossible to determine the best variable selection method for predicting the output in this case. For CHI, IG and MSVM-RFE, it is necessary to select a proper classification method. Under the same variable selection method, multi-class SVMs (WW and CS) showed better average classification accuracy levels than those of the other methods.

In terms of the average time for variable selection and classifier training, the CHI and IG incurred less computational cost than the proposed methods and MSVM-RFE. This is due to the fact that the CHI and IG methods individually score variables based on a certain criterion, while the proposed methods and MSVM-RFE consider non-linear cases by introducing a signomial function and a kernel function,

Table 7 Proposed methods' performance results: Average classification accuracy (%) and standard deviation for test sets, average time (*s*) and standard deviation to select variables and train classifiers, and average number of selected variables

Data set	ℓ_1-*MVS*			*adaptive* MVS_{I}			*adaptive* MVS_{II}		
	TestAcc	*SelTrnT*	*SelVar*	*TestAcc*	*SelTrnT*	*SelVar*	*TestAcc*	*SelTrnT*	*SelVar*
IS	**96.48 ± 0.75**	322.34 ± 170.65	14.80	**96.49 ± 0.92**	190.04 ± 39.02	14.35	**96.65 ± 0.71**	190.29 ± 39.92	14.55
CARD	88.83 ± 1.63	705.16 ± 199.20	15.40	89.10 ± 1.54	761.35 ± 253.47	15.55	89.02 ± 1.48	535.40 ± 109.82	15.50
DNA	**95.88 ± 0.93**	5628.41 ± 616.33	82.80	**95.86 ± 0.95**	5604.64 ± 737.78	82.10	**95.82 ± 0.91**	5538.71 ± 822.47	82.70
MF	**97.28 ± 0.63**	4470.63 ± 1104.25	131.65	97.09 ± 0.78	4426.84 ± 1221.30	135.20	97.04 ± 0.76	4365.61 ± 1089.04	131.95
GAS	**99.10 ± 0.13**	5673.36 ± 1961.40	51.10	**99.09 ± 0.19**	6098.83 ± 2423.18	52.00	**99.10 ± 0.17**	7256.75 ± 4392.09	51.55

For each data set, the comparable or better performance results of the proposed methods are indicated in bold, as compared with those of the CHI, IG and MSVM-RFE methods

Table 8 Performance results: Average classification accuracy (%) and standard deviation for test sets, and average number of selected variables

Data set	*WW*		*CS*		*kNN*		*CART*		*BCT*		ℓ_1-*MSC*	
	TestAcc	*SelVar*	*TestAcc*	*SelVar*	*TestAcc*	*SelVar*	*TestAcc*	*SelVar*	*TestAcc*	*SelVar*	*TestAcc*	*SelVar*
IS	95.29 ± 1.24	19.00	96.07 ± 1.30	19.00	95.00 ± 1.22	19.00	93.59 ± 1.12	19.00	90.05 ± 1.79	19.00	96.95 ± 0.70	14.40
CARD	89.92 ± 1.08	21.00	89.75 ± 1.05	21.00	88.71 ± 1.04	21.00	91.33 ± 1.29	21.00	88.65 ± 1.45	21.00	89.49 ± 1.48	15.80
DNA	95.46 ± 0.81	180.00	95.46 ± 0.78	180.00	87.36 ± 1.56	180.00	90.09 ± 1.51	180.00	88.03 ± 1.38	180.00	95.83 ± 1.07	91.20
MF	97.06 ± 0.95	649.00	97.11 ± 0.78	649.00	94.05 ± 0.83	649.00	89.49 ± 2.03	649.00	83.54 ± 7.20	649.00	96.93 ± 0.65	151.70
GAS	99.04 ± 0.15	128.00	99.05 ± 0.15	128.00	98.48 ± 0.16	128.00	95.64 ± 0.57	128.00	68.91 ± 0.65	128.00	98.96 ± 0.23	52.10

respectively. Notably, for the GAS data set, the MSVM-RFE took longer than the proposed methods, since the MSVM-RFE needs to calculate an inverse of the kernel matrix. Although the proposed methods incurred more computational cost than the CHI and IG, they could reduce the time to determine the number of variables to be selected and suitable classification methods.

Of the proposed methods, the number of variables selected by ℓ_1-MVS was similar to that selected by adaptive MVS_I and adaptive MVS_II. Compared to ℓ_1-MVS, the classification accuracies of adaptive MVS_I and adaptive MVS_II were comparable or slightly better for the IS and CARD data sets. This indicates that imposing different importance weights on input variables had a good effect on the classification accuracy in the case of data sets with a small number of variables.

Table 7 shows, by cross-referencing with Table 8, that for the DNA, MF and GAS data sets, the classifiers obtained by the proposed methods achieved comparable or better classification accuracy levels than ℓ_1-MSC without considering variable selection. This means that for data sets with a large number of variables, the proposed methods can reduce the number of variables while improving the classification accuracy. The classification accuracies of the proposed methods were worse than those of ℓ_1-MSC for the IS and CARD data sets. The proposed methods, however, showed competitive or better levels of classification accuracy than WW, CS, kNN, CART or BCT for most of the data sets (except for the CARD data set).

6. Conclusion

We have proposed several variable selection methods for multi-class classification problems, specifically the ℓ_1-MVS, the adaptive MVS_I and the adaptive MVS_II methods. The proposed variable selection methods are embedded in ℓ_1-MSC and conduct variable selection by treating multiple classes jointly while also considering the nonlinear interaction of the variables. The proposed methods automatically determine the number of variables to be selected, and they obtain classifiers without any additional training process. Classifiers trained using the variables selected by the proposed methods yielded competitive or better classification accuracy levels than those of twelve existing methods with the same number of selected variables. Imposing different importance weights on input variables had a beneficial effect on classification accuracy when using data sets with a small number of variables. For data sets with a large number of variables, the proposed methods reduced the number of variables while improving the classification accuracy.

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

Anand R, Mehrotra K, Mohan CK and Ranka S (1995). Efficient classification for multiclass problems using modular neural networks. *IEEE Transactions on Neural Networks* **6**(1):117–124.

Bache K and Lichman M (2013). University of California, Irvine (UCI) machine learning repository. http://archive.ics.uci.edu/ml

Bay SD (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.

Bermejo P, de la Ossa L, Gamez JA and Puerta JM (2012). Fast wrapper feature subset selection in high-dimensional datasets by means of filter re-ranking. *Knowledge-Based Systems* **25**(1):35–44.

Bertsimas D and Tsitsiklis JN (1997). *Introduction to linear optimization*. 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* **3**(Mar):1229–1243.

Bolon-Canedo V, Sanchez-Marono N and Alonso-Betanzos A (2012). An ensemble of filters and classifiers for microarray data classification. *Pattern Recognition* **45**(1):531–539.

Bradley P and Mangasarian O (1998). Feature selection via concave minimization and support vector machines, *in* Proceedings of the 15th International Conference on Machine Learning, ICML '98, Morgan Kaufmann Publishers, Madison, WI, USA, pp. 82–90.

Breiman L, Friedman J, Stone CJ and Olshen RA (1984). *Classification and regression trees*. Wadsworth and Brooks: Monterey, CA.

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.

Chai H and Domeniconi C (2004). An evaluation of gene selection methods for multi-class microarray data classification. *in* Proceedings of the 2nd European Workshop on Data Mining and Text Mining in Bioinformatics (in conjunction with ECML/PKDD), Pisa, Italy, pp. 3–10.

Chapelle O and Keerthi SS (2008). Multi-class feature selection with support vector machines. *in* Proceedings of the American statistical association, ASA, Denver, CO, USA.

Chapelle O, Vapnik V, Bousquet O and Mukherjee S (2002). Choosing multiple parameters for support vector machines. *Machine Learning* **46**(1):131–159.

Chen X, Zeng X and van Alphen D (2006). Multi-class feature selection for texture classification. *Pattern Recognition Letters* **27**(14):1685–1691.

Clark P and Boswell R (1991). Rule induction with CN2: Some recent improvements. *in* Proceedings of the European Working Session on Machine Learning, EWSL '91, Springer-Verlag, Porto, Portugal, pp. 151–163.

Cover T and Hart P (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory* **13**(1):21–27.

Crammer K and Singer Y (2002). On the algorithmic implementation of multiclass kernel-based vector machines. *The Journal of Machine Learning Research* **2**(Dec):265–292.

Cun YL, Denker JS and Solla SA (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.

Debnath R, Takahide N and Takahashi H (2004). A decision based one-against-one method for multi-class support vector machine. *Pattern Analysis and Applications* **7**(2):164–175.

Dhillon IS, Mallela S and Kumar R (2003). A divisive information-theoretic feature clustering algorithm for text classification. *Journal of Machine Learning Research*. **3**(Mar):1265–1287.

Duan KB, Rajapakse JC and Nguyen MN (2007). One-versus-one and one-versus-all multiclass SVM-RFE for gene selection in cancer classification. *in* Proceedings of the 5th European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics, EvoBIO '07, Springer-Verlag, Valencia, Spain, pp. 47–56.

El Akadi A, Amine, A, El Ouardighi, A and Aboutajdine, D (2011). A two-stage gene selection scheme utilizing MRMR filter and GA wrapper. *Knowledge and Information Systems* **26**(3):487–500.

Fayyad U and Irani K (1993). Multi-interval discretization of continuous-valued attributes for classification learning. *in* Proceedings of the 13th International Joint Conference on Artificial Intelligence, IJCAI '93, Morgan Kaufmann, Chambery, France, pp. 1022–1029.

Forman G (2003). An extensive empirical study of feature selection metrics for text classification. *The Journal of Machine Learning Research* **3**(Mar):1289–1305.

Forman G (2004). A pitfall and solution in multi-class feature selection for text classification. *in* Proceedings of the 21st International Conference on Machine Learning, ICML '04, ACM, Banff, Alberta, Canada, pp. 38–45.

Freund Y and Schapire RE (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences* **55**(1):119–139.

Friedman, J. H (1996). Another approach to polychotomous classification (Vol. 56), Technical report, Department of Statistics, Stanford University, Stanford, CA, USA.

Fung GM and Mangasarian OL (2004). A feature selection Newton method for support vector machine classification. *Computational Optimization and Applications* **28**(2):185–202.

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.

Gutlein M, Frank E, Hall M and Karwath A (2009). Large-scale attribute selection using wrappers. *in* Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining, CIDM '09, IEEE, Nashville, TN, USA, pp. 332–339.

Guyon I and Elisseeff A (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research* **3**(Mar):1157–1182.

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

Hastie T and Tibshirani R (1997). Classification by pairwise coupling. *in* Proceedings of the 10th Annual Conference on Neural Information Processing Systems, NIPS '97, MIT Press, Denver, CO, USA, pp. 507–513.

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

Hsu CW and Lin CJ (2002). A simple decomposition method for support vector machines. *Machine Learning* **46**(1):291–314.

Hsu CW and Lin CJ (2012). *BSVM: A SVM library for the solution of large classification and regression problems*. http://www.csie.ntu.edu.tw/~cjlin/bsvm

Huang K, Zheng D, King I and Lyu MR (2009). Arbitrary norm support vector machines. *Neural Computation* **21**(2):560–582.

Hullermeier E and Vanderlooy S (2010). Combining predictions in pairwise classification: An optimal adaptive voting strategy and its relation to weighted voting. *Pattern Recognition* **43**(1):128–142.

Hwang K, Lee K, Lee C and Park S (2013). Embedded variable selection method using signomial classification. Technical Report 2013-03, Department of Industrial and Systems Engineering, KAIST, Daejeon, Republic of Korea.

Hwang K, Lee K, Lee C and Park S (2015). Multi-class classification using a signomial function. *Journal of the Operational Research Society* **66**(3):434-449.

Ihaka R and Gentleman R (1996). R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics* **5**:299–314.

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 GH (1997). Wrappers for feature subset selection. *Artificial Intelligence* **97**(1–2):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 TN, 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.

Lee K, Kim N and Jeong MK (2014). The sparse signomial classification and regression model. *Annals of Operations Research* **216**(1):257–286.

Li JT and Jia YM (2010). Huberized multiclass support vector machine for microarray classification. *Acta Automatica Sinica* **36**(3):399–405.

Li T, Zhang C and Ogihara M (2004). A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression. *Bioinformatics* **20**(15): 2429–2437.

Liu J, Ranka S and Kahveci T (2008). Classification and feature selection algorithms for multi-class CGH data. *Bioinformatics* **24**(13):i86–i95.

Maldonado S and Weber R (2009). A wrapper method for feature selection using support vector machines. *Information Sciences* **179**(13):2208–2217.

Maldonado S, Weber R and Basak J (2011). Simultaneous feature selection and classification using kernel-penalized support vector machines. *Information Sciences* **181**(1):115–128.

Mangasarian OL (1999). Generalized support vector machines. *Advances in Neural Information Processing Systems*:135–146.

Mangasarian OL (2006). Exact 1-norm support vector machines via unconstrained convex differentiable minimization. *The Journal of Machine Learning Research* **7**(Jul):1517–1530.

Matlab (2010). Matlab statistics toolbox. http://www.mathworks.com

Michie D, Spiegelhalter DJ and Taylor CC (1994). Statlog collection. ftp://ftp.ncc.up.pt/pub/statlog/

Peng Y, Wu Z and Jiang J (2010). A novel feature selection approach for biomedical data classification. *Journal of Biomedical Informatics* **43**(1):15–23.

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

Pudil P, Novovicova J and Kittler J (1994). Floating search methods in feature selection. *Pattern Recognition Letters* **15**(11): 1119–1125.

Quinlan JR (1986). Induction of decision trees. *Machine Learning* **1**(1):81–106.

Quinlan JR (1993). *C4.5: Programs for machine learning (morgan kaufmann series in machine learning)*, Morgan Kaufmann, San Francisco, CA, USA.

Rajapakse JC and Mundra PA (2013). Multiclass gene selection using pareto-fronts. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **10**(1):87–97.

Rakotomamonjy A (2003). Variable selection using SVM based criteria. *Journal of Machine Learning Research* **3**(Mar): 1357–1370.

Ramaswamy S, Tamayo P, Rifkin R, Mukherjee S, Yeang CH, Angelo M, Ladd C, Reich M Latulippe E, Mesirov JP et al (2001). Multiclass cancer diagnosis using tumor gene expression signatures. *Proceedings of the National Academy of Sciences of the United States of America* **98**(26):15149–15154.

Rivals I and Personnaz L (2003). MLPs (mono layer polynomials and multi layer perceptrons) for nonlinear modeling. *Journal of Machine Learning Research* **3**(Mar):1383–1398.

Ruiz R, Riquelme JC and Aguilar-Ruiz JS (2006). Incremental wrapper-based gene selection from microarray data for cancer classification. *Pattern Recognition* **39**(12):2383–2392.

Shieh M and Yang C (2008). Multiclass SVM-RFE for product form feature selection. *Expert Systems with Applications* **35**(1–2): 531–541.

Somol P, Pudil P and Kittler J (2004). Fast branch & bound algorithms for optimal feature selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26**(7):900–912.

Stoppiglia H, Dreyfus G, Dubois R and Oussar Y (2003). Ranking a random feature for variable and feature selection. *Journal of Machine Learning Research* **3**(Mar):1399–1414.

Tipping ME (2001). Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research* **1**(Jun): 211–244.

Torkkola K (2003). Feature extraction by non parametric mutual information maximization. *Journal of Machine Learning Research* **3**(Mar):1415–1438.

Veenman CJ and Bolck A (2011). A sparse nearest mean classifier for high dimensional multi-class problems. *Pattern Recognition Letters* **32**(6):854–859.

Wang L and Shen X (2006). Multi-category support vector machines, feature selection and solution path. *Statistica Sinica* **16**(2): 617–633.

Wang L and Shen X (2007*a*). On ℓ_1-norm multiclass support vector machines. *Journal of the American Statistical Association* **102**(478):583–594.

Wang L and Shen X (2007*b*). On ℓ_1-norm multiclass support vector machines: Methodology and theory. *Journal of the American Statistical Association* **102**(478):583–594.

Wang L, Zhu J and Zou H (2006). The doubly regularized support vector machine. *Statistica Sinica* **16**(2):589–615.

Wang L, Zhu J and Zou H (2008). Hybrid huberized support vector machines for microarray classification and gene selection. *Bioinformatics* **24**(3):412–419.

Weston J, Elisseeff A, Scholkopf B and Tipping M (2003). Use of the zero-norm with linear models and kernel methods. *Journal of Machine Learning Research* **3**(Mar):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:563–532.

Weston J and Watkins C (1999). Support vector machines for multi-class pattern recognition. *in* Proceedings of the 7th European Symposium on Artificial Neural Networks, ESANN '99, Citeseer, Bruges, Belgium:219–224.

Xpress (2015). http://www.fico.com/en

Yang J and Honavar V (1998). Feature subset selection using a genetic algorithm. *in* Feature extraction, construction and selection, Kluwer Academic Publishers, Norwell, MA, USA: 117–136.

Yang Y and Pedersen JO (1997). A comparative study on feature selection in text categorization. *in* Proceedings of the 14th International Conference on Machine Learning, ICML '97, Morgan Kaufmann Publishers, Nashville, TN, USA, pp. 412–420.

Yang Y and Webb GI (2001). Proportional k-interval discretization for naive-Bayes classifiers. *in* Proceedings of the 12th European Conference on Machine learning, ECML '01, Springer, Freiburg, Germany, pp. 564–575.

You M and Li GZ (2011). Feature selection for multi-class problems by using pairwise-class and all-class techniques. *International Journal of General Systems* **40**(4):381–394.

Zhang HH, Liu Y, Wu Y and Zhu J (2008). Variable selection for the multicategory SVM via adaptive sup-norm regularization. *Electronic Journal of Statistics* **2**:149–167.

Zhang Y, Ding C and Li T (2008). Gene selection algorithm by combining reliefF and mRMR. *BMC Genomics* **9**(2), p. S27.

Zhao Y and Yand Z (2010). Improving MSVM-RFE for multiclass gene selection. *in* B. S. L.-Y. W. Luonan Chen, Xiang-Sun Zhang and Y. Wang, eds, 'Proceedings of the 4th International Conference on Computational Systems Biology, ISB '10, World-publishing-corporation, Suzhou, China: 43–50.

Zhou W, Zhang L and Jiao L (2002). Linear programming support vector machines. *Pattern Recognition* **35**(12):2927–2936.

Zhou X and Tuck DP (2007). MSVM-RFE: extensions of SVM-RFE for multiclass gene selection on DNA microarray data. *Bioinformatics* **23**(9):1106–1114.

Zhu J, Rosset S, Hastie T and Tibshirani R (2003). 1-norm support vector machines. *in* 'Proceedings of the 16th Annual Conference on Neural Information Processing Systems, NIPS '03, MIT Press, Vancouver and Whistler, BC, Canada, pp. 49–56.

Zou H (2007). An improved 1-norm SVM for simultaneous classification and variable selection. *Journal of Machine Learning Research-Proceedings Track* **2**:675–681.

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

Received 24 February 2014;
accepted 13 September 2016