Multi-class classification using a signomial function

Kyoungmi Hwang,

Kyungsik Lee,

Chungmok Lee,

and Sungsoo Park

Technical Report

Nov 18, 2013

Department of Industrial and Systems Engineering

Korea Advanced Institute of Science and Technology (KAIST)

MULTI-CLASS CLASSIFICATION USING A SIGNOMIAL FUNCTION

This is a pre-print of an article published in [Journal of the Operational Research Society]. The definitive publisher-authenticated version [Hwang et al (2013). Multi-class classification using a signomial function] will appear online at: [http://www.palgrave-journals.com/jors/index.html]

* 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

Multi-class classification using a signomial function

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 multi-class classification methods using signomial function. Each of these methods directly constructs a multi-class classifier by solving a single optimization problem. Since the number of possible signomial terms is extremely large, we propose a column generation method that iteratively generates good signomial terms. Both of these methods obtain better or comparable classification accuracies and provide more sparse classifiers than the multi-class SVMs.

Keywords: multi-class classification; signomial function; sparse classifier; column generation algorithm;

Introduction

The multi-class classification, which is an important problem in data mining and pattern recognition, refers to assigning each of the observations into one of $k > 2$ classes. The goal of the multi-class classification is to build a multi-class classifier which can determine the classes of any future data using a given training set.

Many of the algorithms introduced to solve multi-class classification problems are based on those developed to solve binary classification problems (Anand et al., 1995; Clark and Boswell, 1991; Debnath et al., 2004; Galar et al., 2011; Lorena et al., 2008). These methods first decompose a multi-class classification problem to a set of multiple binary classification problems and then combine the binary classifiers obtained from the binary classification problems in certain ways to construct a multi-class classifier. One-versus-all (OVA) (Anand et al., 1995; Clark and Boswell, 1991) and all-versus-all (AVA) (Debnath et al., 2004; Lam and Moy, 1996) are two of the most common decomposition methods. The former (OVA) converts a k-class classification problem into k binary classification problems where one class is separated from the remaining ones, while the latter (AVA) decomposes a k-class classification problem into $k(k-1)/2$ binary classification problems where the binary classifier for each pair of classes is constructed. Several strategies have been proposed for combining the outputs of the binary classifiers (Friedman, 1996; Hastie and Tibshirani, 1997;

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

Hüllermeier and Vanderlooy, 2010). One of these, the voting strategy, is a widely applied simple method in which each binary classifier votes for the predicted class, and the class with the highest number of votes is subsequently selected.

Such approaches, however, have a number of drawbacks. The voting strategy has been reported to lead to ties or contradictory voting among the different classes (Tax and Duin, 2002), resulting in the degradation of classification accuracy. Also, in the case of many classes, the voting can be time-consuming as the binary prediction has to be applied many times. In addition, it is hard to capture the correlations among classes using the voting strategy since a multi-class classification problem is divided into multiple independent binary classification problems (Crammer and Singer, 2002). Single machine approaches which directly construct a multi-class classifier by solving a single optimization problem have been proposed to overcome these drawbacks (Bennett and Mangasarian, 1994; Choo and Wedley, 1985; Crammer and Singer, 2002; Lee et al., 2004; Pavur and Loucopoulos, 1995; Vapnik, 1998; Weston and Watkins, 1999). These include the method developed by Bennett and Mangasarian (1994) using a piecewise linear classifier and multi-class support vector machine (SVM) methods that extend binary SVM to multi-class problems (Crammer and Singer, 2002; Lee et al., 2004; Vapnik, 1998; Weston and Watkins, 1999).

Multi-class SVM methods often perform well in terms of classification accuracy (Hsu and Lin, 2002*a*; Lee et al., 2004) and can even generate nonlinear classifiers using kernel functions. A good classifier should be able to perform two main functions: prediction and interpretation (Baesens et al., 2009). For example, when a doctor diagnoses a disease based on the presenting symptoms of a patient, he/she first has to determine whether there actually is a 'disease'; if so, the doctor has then to be able to explain his/her diagnosis based on the specific combination of presenting symptoms. The doctor may require a classifier, which is described with the symptoms (original variables) and which is as simple as possible so that the diagnosis can be reached easily. Unfortunately, in many cases, the diagnosis requires knowledge of complex combinations (often nonlinear) of symptoms. The knowledge can be obtained using the kernel trick inherent in SVM methods. However, with the kernel trick, SVM methods do not use the mapping function explicitly; rather, they produce a classifier in the form of a kernel expansion. Consequently, it is not easy to interpret the relationship between the original variables (symptoms) and classes (existence of a disease).

SVM methods often produce a sparse kernel expansion, but various SVM methods have been developed which produce even more sparse classifiers (Mangasarian, 1999; Psorakis et al., 2010; Weston et al., 2003). Such classifiers, however, are sparse in feature space, not in original space, which implies that many (not sparse) of the original variables are necessary to interpret the classifier obtained.

It is well-known that the performance of SVM methods is sensitive to the scale of the input data. As a result, elaborated data scaling steps have been incorporated in many SVM methods as a preprocessing procedure (Hsu et al., 2003). A classification method that is robust against the scale of the input data is of practical importance as this would enable the procedure for finding the proper data scaling to be eliminated from the algorithm.

In this paper, we seek to directly construct a (nonlinear) multi-class classifier which is sparse and explicitly

described in original space. To achieve this, we propose a multi-class classifier using signomial function, and develop two multi-class signomial classification methods.

Multi-class classification using signomial function

Let $\mathbf{x} = (x_1, ..., x_n)$ be a vector of positive real 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 $\mathbf{0} \notin D$. We consider the set D defined by four parameters, d_{min}, d_{max}, L, and T, as follows:

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

Let $X = \{1, 2, \ldots, m\}$ be the index set of m training examples $\mathbf{x}_i$ where $\mathbf{x}_i \in \mathbb{R}^n_{++}, i = 1, ..., m$. Let each example belong to class k, $k \in K := \{1, ..., c\}, c > 2$, where c is the number of classes, and X^k be the index set of examples that belong to class k, $k \in K$ satisfying $\bigcup_{k \in K} X^k = X$. We attempt to discriminate between c classes by a signomial function classifier (1), while keeping the number of terms in the function as small as possible.

To obtain such a signomial function, we propose two multi-class signomial classification methods in the following sections. Both of these are multi-class extensions of the binary classification method proposed by Lee et al. (2012). Each method gives a multi-class signomial classifier by solving a single optimization problem. Two different approaches have been adopted for minimizing the number of signomial terms in the resulting classifier: (1) to adopt the L_1-norm as the regularization term; (2) to adopt the L_0-norm as the regularization term. We denote these two methods as the L_1-norm and L_0-norm methods for multi-class signomial classification (L_1-MSC and L_0-MSC), respectively.

L_1-norm method for multi-class signomial classification (L_1-MSC)

L_1-norm multi-class signomial classification problem

In this section, we describe the L_1-norm method for multi-class signomial classification (L_1-MSC). The following optimization problem is developed to obtain a signomial classifier for a multi-class classification problem.

$$\min \sum_{k\in K} ||\mathbf{w}^k||_1 + C\sum_{k\in K}\sum_{i\in X^k}\sum_{l\in K\setminus\{k\}} \varepsilon_i^{kl} \tag{3}$$

$$\text{s.t.} \sum_{\mathbf{d}\in D^k} w_{\mathbf{d}}^k g_{\mathbf{d}}(\mathbf{x}_i) + b^k - \{\sum_{\mathbf{d}\in D^l} w_{\mathbf{d}}^l g_{\mathbf{d}}(\mathbf{x}_i) + b^l\} + \varepsilon_i^{kl} \geq 1, \qquad \forall l \in K\setminus\{k\}, i \in X^k, k \in K, \tag{4}$$

$$\mathbf{w}^k \in \mathbb{R}^{|D^k|}, b^k \in \mathbb{R}, \qquad \forall k \in K,$$

$$\varepsilon_i^{kl} \in \mathbb{R}_+, \qquad \forall l \in K\setminus\{k\}, i \in X^k, k \in K,$$

where C is the penalty parameter, and D^k is the set of exponents for class k, $k \in K$ defined by (2). L_1-norm $||\mathbf{w}^k||_1$ is defined as $\sum_{\mathbf{d}\in D^k} |w_{\mathbf{d}}^k|$, and ε_i^{kl} is the misclassification error which is positive if data i is misclassified. The objective function (3) is to minimize $\sum_{k\in K} ||\mathbf{w}^k||_1$ and the sum of misclassification errors. The parameter C is a positive real number and controls the relative importance of training error to the L_1-norm of $\mathbf{w}^k$.

We employ the L_1-norm as the regularization term to ensure that the number of terms in the resulting classifier is small. One appealing feature of the L_1-norm is that it can force sparsity in the resulting classifier with relatively little computational burden (Huang et al., 2009). For example, with respect to the L_1-norm SVM, there is empirical evidence for very sparse solutions being generated by the L_1-norm formulation (Bradley and Mangasarian, 1998; Fung and Mangasarian, 2004; Zhu et al., 2003).

We can reformulate the above problem into an equivalent linear programming problem by replacing $w_{\mathbf{d}}^k$ with $w_{\mathbf{d}}^{k+} - w_{\mathbf{d}}^{k-}$ where $w_{\mathbf{d}}^{k+} \geq 0$ and $w_{\mathbf{d}}^{k-} \geq 0$ as follows:

[L_1-MSCP]

$$\min \sum_{k\in K}\sum_{\mathbf{d}\in D^k} (w_{\mathbf{d}}^{k+} + w_{\mathbf{d}}^{k-}) + C\sum_{k\in K}\sum_{i\in X^k}\sum_{l\in K\setminus\{k\}} \varepsilon_i^{kl} \tag{5}$$

$$\text{s.t.} \sum_{\mathbf{d}\in D^k} (w_{\mathbf{d}}^{k+} - w_{\mathbf{d}}^{k-}) g_{\mathbf{d}}(\mathbf{x}_i) + b^k - \{\sum_{\mathbf{d}\in D^l} (w_{\mathbf{d}}^{l+} - w_{\mathbf{d}}^{l-}) g_{\mathbf{d}}(\mathbf{x}_i) + b^l\} + \varepsilon_i^{kl} \geq 1,$$

$$\forall l \in K\setminus\{k\}, i \in X^k, k \in K, \tag{6}$$

$$\mathbf{w}^{k+}, \mathbf{w}^{k-} \in \mathbb{R}_+^{|D^k|}, b^k \in \mathbb{R}, \qquad \forall k \in K,$$

$$\varepsilon_i^{kl} \in \mathbb{R}_+, \qquad \forall l \in K\setminus\{k\}, i \in X^k, k \in K.$$

We refer to this problem as the L_1-norm multi-class signomial classification problem (L_1-MSCP). By solving the L_1-MSCP, we obtain a signomial classifier for a multi-class classification problem. It is easy to see that the solution has either $w_{\mathbf{d}}^{k+}$ or $w_{\mathbf{d}}^{k-}$ equal to zero, so $|w_{\mathbf{d}}^k| = w_{\mathbf{d}}^{k+} + w_{\mathbf{d}}^{k-}$.

Let $(\hat{\mathbf{w}}^+, \hat{\mathbf{w}}^-, \hat{\mathbf{b}})$ be an optimal solution of the L_1-MSCP, and let $\hat{D}^k := \{\mathbf{d} \in D^k : \hat{w}_{\mathbf{d}}^{k+} \neq 0 \text{ or } \hat{w}_{\mathbf{d}}^{k-} \neq 0\}$ for all $k \in K$. After solving the L_1-MSCP, the classifier is as follows:

$$f(\mathbf{x}) = \underset{k\in K}{\text{argmax}}\Big(f_k(\mathbf{x}) = \sum_{\mathbf{d}\in \hat{D}^k} (\hat{w}_{\mathbf{d}}^{k+} - \hat{w}_{\mathbf{d}}^{k-}) g_{\mathbf{d}}(\mathbf{x}) + \hat{b}^k\Big), \tag{7}$$

where $\mathbf{x}$ is classified as class k if $f(\mathbf{x}) = k$. If there are more than one class with the same maximum value, we randomly select one of them.

Column generation for L_1-MSCP

The L_1-MSCP has as many $w_{\mathbf{d}}^{k+}$ and $w_{\mathbf{d}}^{k-}$ variables as the size of $D^k, k \in K$. Depending on the size of $D^k, k \in K$, the L_1-MSCP may have such a large number of variables that the problem is intractable. However, even then, the problem can be solved efficiently by the column generation algorithm (Bertsimas and Tsitsiklis, 1997). The algorithm generates exponents $\mathbf{d} \in \bigcup_{k \in K} D^k$ as needed – rather than given a priori – and therefore we do not need to solve the L_1-MSCP with all variables, the number of which might be large. Furthermore, provided that the column generation problem is solved exactly, the algorithm is guaranteed to find an optimal solution for the L_1-MSCP.

Suppose that we have $\tilde{D}^k \subset D^k$, which is the set of already generated exponents for class k, $k \in K$. The restricted master problem of the L_1-MSCP is given as follows:

$[L_1\text{-MSCP}(\tilde{D})]$

$$\min \sum_{k \in K} \sum_{\mathbf{d} \in \tilde{D}^k} (w_{\mathbf{d}}^{k+} + w_{\mathbf{d}}^{k-}) + C \sum_{k \in K} \sum_{i \in X^k} \sum_{l \in K \setminus \{k\}} \varepsilon_i^{kl} \tag{8}$$

$$\text{s.t.} \sum_{\mathbf{d} \in \tilde{D}^k} (w_{\mathbf{d}}^{k+} - w_{\mathbf{d}}^{k-}) g_{\mathbf{d}}(\mathbf{x}_i) + b^k - \{ \sum_{\mathbf{d} \in \tilde{D}^l} (w_{\mathbf{d}}^{l+} - w_{\mathbf{d}}^{l-}) g_{\mathbf{d}}(\mathbf{x}_i) + b^l \} + \varepsilon_i^{kl} \geq 1,$$

$$\forall l \in K \setminus \{k\}, i \in X^k, k \in K, \tag{9}$$

$$\mathbf{w}^{k+}, \mathbf{w}^{k-} \in \mathbb{R}_+^{|\tilde{D}^k|}, b^k \in \mathbb{R}, \qquad \forall k \in K,$$

$$\varepsilon_i^{kl} \in \mathbb{R}_+, \qquad \forall l \in K \setminus \{k\}, i \in X^k, k \in K.$$

Let $\alpha_i^{kl}, \forall l \in K \setminus \{k\}, i \in X^k, k \in K$ be nonnegative dual variables associated with constraints (9). Among the exponents $\mathbf{d} \in \bigcup_{k \in K} D^k \setminus \tilde{D}^k$, we identify a number of exponents which are profitable to the L_1-MSCP($\tilde{D}$). Let $z^k(\mathbf{d}) := \sum_{i \in X^k} \sum_{l \in K \setminus \{k\}} \hat{\alpha}_i^{kl} g_{\mathbf{d}}(\mathbf{x}_i) - \sum_{l \in K \setminus \{k\}} \sum_{j \in X^l} \hat{\alpha}_j^{lk} g_{\mathbf{d}}(\mathbf{x}_j)$, where $\hat{\boldsymbol{\alpha}}^{kl}, \forall l \in K \setminus \{k\}, k \in K$ is an optimal dual solution of the L_1-MSCP($\tilde{D}$). The set of profitable exponents for class k, $k \in K$ is as follows:

$$D_{profit}^k := \{\mathbf{d} : |z^k(d)| > 1, \mathbf{d} \in D^k \setminus \tilde{D}^k\}. \tag{10}$$

We generate the best profitable exponent $\mathbf{d} \in D_{profit}^k$ for each class k, $k \in K$. The column generation problem for class k to find such an exponent is constructed as follows:

$$[\text{CGP}_k] \quad \hat{z}^k := \max \{z^{k+}, z^{k-}\}, \tag{11}$$

$$z^{k+} := \max \{z^k(\mathbf{d}) : \mathbf{d} \in D^k \setminus \tilde{D}^k\}, \tag{12}$$

$$z^{k-} := \max \{-z^k(\mathbf{d}) : \mathbf{d} \in D^k \setminus \tilde{D}^k\}. \tag{13}$$

After solving problems (12) and (13) respectively, we select the maximum value of z^{k+} and z^{k-} as the objective value of CGP_k, and the solution corresponding to the maximum value as the solution of CGP_k, respectively. Since problems (12) and (13) are NP-hard (Theorem 1 in (Lee et al., 2012)), we solve these using a heuristic algorithm presented in the following subsection.

1. **Initialize**: For all $k \in K$, $\tilde{D}^k := \emptyset$, $\hat{D}^k := \emptyset$ and $U^k := \emptyset$.
2. **repeat**
3. Solve the L_1-MSCP($\tilde{D}$) with $\tilde{D}^k, k \in K$.
4. $(\hat{\mathbf{w}}^{k+}, \hat{\mathbf{w}}^{k-}, \hat{b}^k)$ for all $k \in K \leftarrow$ optimal solution to the L_1-MSCP($\tilde{D}$).
5. $\hat{\alpha}_i^{kl}$ for all $l \in K \setminus \{k\}, i \in X^k, k \in K \leftarrow$ optimal dual solution to the L_1-MSCP($\tilde{D}$).
6. **for** $k \in K$ **do**
7. Solve CGP_k.
8. $\hat{\mathbf{d}}^k \leftarrow$ solution of CGP_k.
9. $\hat{z}^k \leftarrow$ objective value of CGP_k.
10. **if** $\hat{z}^k \geq 1 + \epsilon$ **then** $\tilde{D}^k := \tilde{D}^k \cup \{\hat{\mathbf{d}}^k\}$ and $U^k := 1$.
11. **else** $U^k := 0$.
12. **end-do**
13. **until** $U^k = 0$ for all $k \in K$.
14. $\hat{D}^k := \{\mathbf{d} \in \tilde{D}^k : \hat{w}_{\mathbf{d}}^{k+} \neq 0 \text{ or } \hat{w}_{\mathbf{d}}^{k-} \neq 0\}$ for all $k \in K$.
15. Return the classifier $f(\mathbf{x}) = \underset{k \in K}{\operatorname{argmax}}\Big(f_k(\mathbf{x}) = \sum_{\mathbf{d} \in \hat{D}^k} (\hat{w}_{\mathbf{d}}^{k+} - \hat{w}_{\mathbf{d}}^{k-}) g_{\mathbf{d}}(\mathbf{x}) + \hat{b}^k\Big)$.

Figure 1: L_1-norm method for multi-class signomial classification (L_1-MSC)

Let $\hat{z}^k$ and $\hat{\mathbf{d}}^k$ be the objective value and the solution of CGP_k, respectively. For each class k, if $\hat{z}^k > 1$, then $\tilde{D}^k$ is updated by adding the $\hat{\mathbf{d}}^k$ to the master problem. After updating $\tilde{D}^k$ for all $k \in K$, we solve the L_1-MSCP($\tilde{D}$) again. The procedure is repeated until $\hat{z}^k \leq 1$ for all $k \in K$. In our implementation, however, we used $\hat{z}^k < 1 + \epsilon$ as the stopping condition instead of $\hat{z}^k \leq 1$. We set $\epsilon = 0.01$.

The overview of the L_1-MSC is described in Figure 1.

Heuristic algorithm for the column generation problem

In this subsection, we present a heuristic algorithm for problem (12); problem (13) can be solved analogously to problem (12). First, we construct a relaxed problem by dropping the integrality condition present in problem (12) as follows:

$$\max z^k(\mathbf{d}) := \sum_{l\in K\setminus\{k\}}\sum_{i\in X^k}\hat{\alpha}_i^{kl}\prod_{j=1}^{n}x_j^{d_j} - \sum_{l\in K\setminus\{k\}}\sum_{j\in X^l}\hat{\alpha}_j^{lk}\prod_{j=1}^{n}x_j^{d_j} \tag{14}$$

$$\text{s.t. } d_{min} \le d_j \le d_{max}, \qquad j = 1, ..., n, \tag{15}$$

$$\sum_{j=1}^{n}|d_j| \le L, \tag{16}$$

$$\mathbf{d} \in \mathbb{R}^n.$$

Since the objective function (14) is neither convex nor concave, the relaxed problem is solved using the Frank–Wolfe algorithm (Frank and Wolfe, 1956) to obtain a local optimal solution. To obtain a solution such that $T\mathbf{d}$ is an integer vector, we multiply the obtained solution $\mathbf{d}$ by T and then round down $T\mathbf{d}$.

The heuristic algorithm for solving problem (12) is as follows:

1. Choose an initial solution, $\mathbf{d}^{(0)} := \mathbf{0}$ and $z^{(0)} := z^k(\mathbf{d}^{(0)})$, and let $r := 1$.

2. Determine a search direction, $p^{(r)}$.

   In the Frank–Wolf algorithm, $p^{(r)}$ is determined through the solution of the following approximation problem, which is obtained by replacing the objective function (14) with its first-order Taylor expansion of $z^k(\mathbf{d})$ around $\mathbf{d}^{(r-1)}$: $z^{(r-1)} + \nabla z^k(\mathbf{d}^{(r-1)})^T \cdot (\mathbf{y} - \mathbf{d}^{(r-1)})$.

$$[\mathrm{SD}_{CG}^{(r)}] \quad \max c_0 + \sum_{j=1}^{n} c_j y_j \tag{17}$$

$$\text{s.t. } d_{min} \le y_j \le d_{max}, \qquad j = 1, ..., n, \tag{18}$$

$$\sum_{j=1}^{n}|y_j| \le L, \tag{19}$$

$$\mathbf{y} \in \mathbb{R}^n,$$

   where $c_0 = z^{(r-1)} - \nabla z^k(\mathbf{d}^{(r-1)})^T \cdot \mathbf{d}^{(r-1)}$ and $(c_1, \ldots, c_n) = \nabla z^k(\mathbf{d}^{(r-1)})^T$. This problem is a variant of the linear knapsack problem. To obtain an optimal solution to the above problem, we sort $c_j, j = 1, \ldots, n$ in the decreasing order of absolute values and then assign the value of y_j in a greedy manner. Let $\hat{\mathbf{y}}^{(r)}$ be an optimal solution. The search direction is $\mathbf{p}^{(r)} := \hat{\mathbf{y}}^{(r)} - \mathbf{d}^{(r-1)}$.

3. Determine the step length $\lambda^{(r)}$, such that

$$\lambda^{(r)} = \underset{\lambda\in\{i/s:i=0,\ldots,s\}}{\operatorname{argmax}} z^k(\mathbf{d}^{(r-1)} + \lambda\mathbf{p}^{(r)}). \tag{20}$$

4. New iteration point: $\mathbf{d}^{(r)} := \mathbf{d}^{(r-1)} + \lambda^{(r)}\mathbf{p}^{(r)}$ and $z^{(r)} := z^k(\mathbf{d}^{(r)})$.

5. If $z^{(r)} - z^{(r-1)} < \nu$ then go to step 6; otherwise, set $r := r + 1$ and go to step 2.

6. Round off $T\mathbf{d}^{(r-1)}$, and set $\hat{d}_j := \lfloor T \times d_j^{(r-1)} \rfloor / T, j = 1, \ldots, n$.

7. Return $\hat{\mathbf{d}}$ and $\hat{z} := z^k(\hat{\mathbf{d}})$.

In our implementation, we set $s = 5$ and $\nu = 0.01$.

L_0-norm method for multi-class signomial classification (L_0-MSC)

L_0-norm multi-class signomial classification problem

In this section, we describe the L_0-norm method for multi-class signomial classification (L_0-MSC). The L_0-norm of $\mathbf{w}^k$ is adopted as the regularization term instead of the L_1-norm, and an optimization problem is developed as follows:

[L_0-MSCP]

$$\min \sum_{k \in K} ||\mathbf{w}^k||_0 + C \sum_{k \in K} \sum_{i \in X^k} \sum_{l \in K \setminus \{k\}} \varepsilon_i^{kl} \tag{21}$$

$$\text{s.t.} \sum_{\mathbf{d} \in D^k} w_{\mathbf{d}}^k g_{\mathbf{d}}(\mathbf{x}_i) + b^k - \{ \sum_{\mathbf{d} \in D^l} w_{\mathbf{d}}^l g_{\mathbf{d}}(\mathbf{x}_i) + b^l \} + \varepsilon_i^{kl} \geq 1, \qquad \forall l \in K \setminus \{k\}, i \in X^k, k \in K, \tag{22}$$

$$\mathbf{w}^k \in \mathbb{R}^{|D^k|}, b^k \in \mathbb{R}, \qquad \forall k \in K,$$

$$\varepsilon_i^{kl} \in \mathbb{R}_+, \qquad \forall l \in K \setminus \{k\}, i \in X^k, k \in K,$$

where $||\mathbf{w}^k||_0$ is the cardinality of set $\{w_{\mathbf{d}}^k \mid w_{\mathbf{d}}^k \neq 0, \mathbf{d} \in D^k\}$ and ε_i^{kl} is the misclassification error. Here, C is the penalty parameter, and D^k is the set of exponents for class k, $k \in K$. Minimizing $||\mathbf{w}^k||_0$ will result in direct reduction in the number of signomial terms in the resulting classifier and may produce the most sparse signomial classifier. We refer to this problem as the L_0-norm multi-class signomial classification problem (L_0-MSCP).

Minimizing $||\mathbf{w}^k||_0$ makes the L_0-MSCP NP-hard: Let us suppose that we allow only instances in which with $|K| = 2$, $D^1 = D^2 = D$, $w_{\mathbf{d}}^1 = -w_{\mathbf{d}}^2 = w_{\mathbf{d}}/2$, $b^1 = -b^2 = b/2$, $\varepsilon_i^{12} = 0$ for all $i \in X^1$, and $\varepsilon_i^{21} = 0$ for all $i \in X^2$. In this case, the restricted problem of the L_0-MSCP is obtained as follows:

$$\min ||\mathbf{w}||_0 \tag{23}$$

$$\text{s.t.} \sum_{\mathbf{d} \in D} w_{\mathbf{d}} g_{\mathbf{d}}(\mathbf{x}_i) + b \geq 1, \qquad \forall i \in X^1, \tag{24}$$

$$\sum_{\mathbf{d} \in D} w_{\mathbf{d}} g_{\mathbf{d}}(\mathbf{x}_i) + b \leq -1, \qquad \forall i \in X^2, \tag{25}$$

$$\mathbf{w} \in \mathbb{R}^{|D|}, b \in \mathbb{R}.$$

This problem is NP-hard (Amaldi and Kann, 1998; Garey and Johnson, 1979) and a special case of the L_0-MSCP. Hence, we propose a heuristic algorithm for solving the L_0-MSCP. This algorithm is described in the next subsection.

Let $(\hat{\mathbf{w}}^k, \hat{b}^k)$ be a (possibly sub-optimal) solution of the L_0-MSCP, and let $\hat{D}^k := \{\mathbf{d} \in D^k : \hat{w}^k_{\mathbf{d}} \neq 0\}$ for all $k \in K$. The resulting classifier is as follows:

$$f(\mathbf{x}) = \underset{k \in K}{\operatorname{argmax}} \Big(f_k(\mathbf{x}) = \sum_{\mathbf{d} \in \hat{D}^k} \hat{w}^k_{\mathbf{d}} g_{\mathbf{d}}(\mathbf{x}) + \hat{b}^k \Big), \tag{26}$$

where $\mathbf{x}$ is classified in class k if $f(\mathbf{x}) = k$. If there are more than one class with the same maximum value, we randomly select one of them.

Heuristic algorithm for L_0-MSCP

We first generate only a set of profitable exponents (i.e, signomial terms) with a limited size – rather than enumerating all of the elements of $D^k, k \in K$ – since the size of D^k can be exponentially large, which makes the problem intractable. To generate as many profitable exponents as possible, we solve the L_1-MSCP with a large value of C (e.g, 10^6), which may produce a classifier having many signomial terms (i.e, a non-sparse classifier). Let $\acute{D}^k \subset D^k$ be a set of such exponents for class k. After applying the exponent-generating procedure, we attempt to build a classifier using the smallest possible number of exponents of $\acute{D}^k, k \in K$ by constructing the following problem:

[L_0-MSCP($\acute{D}$)]

$$\min \sum_{k \in K} ||\mathbf{w}^k||_0 + C \sum_{k \in K} \sum_{i \in X^k} \sum_{l \in K \setminus \{k\}} \varepsilon_i^{kl} \tag{27}$$

$$\text{s.t.} \sum_{\mathbf{d} \in \acute{D}^k} w^k_{\mathbf{d}} g_{\mathbf{d}}(\mathbf{x}_i) + b^k - \{ \sum_{\mathbf{d} \in \acute{D}^l} w^l_{\mathbf{d}} g_{\mathbf{d}}(\mathbf{x}_i) + b^l \} + \varepsilon_i^{kl} \geq 1, \qquad \forall l \in K \setminus \{k\}, i \in X^k, k \in K, \tag{28}$$

$$\mathbf{w}^k \in \mathbb{R}^{|\acute{D}^k|}, b^k \in \mathbb{R}, \qquad \forall k \in K,$$

$$\varepsilon_i^{kl} \in \mathbb{R}_+, \qquad \forall l \in K \setminus \{k\}, i \in X^k, k \in K.$$

We use a log function, $\sum_{d \in D^k} \ln(\delta + |w^k_{\mathbf{d}}|)$, instead of $||\mathbf{w}^k||_0$. Weston et al. (2003) also proposed the use of a log function as a substitute for L_0-norm for feature selection. We construct the following problem by using the log function:

$$\min \sum_{k \in K} \sum_{\mathbf{d} \in \acute{D}^k} \ln(\delta + |w^k_{\mathbf{d}}|) + C \sum_{k \in K} \sum_{i \in X^k} \sum_{l \in K \setminus \{k\}} \varepsilon_i^{kl} \tag{29}$$

$$\text{s.t.} \sum_{\mathbf{d} \in \acute{D}^k} w^k_{\mathbf{d}} g_{\mathbf{d}}(\mathbf{x}_i) + b^k - \{ \sum_{\mathbf{d} \in \acute{D}^l} w^l_{\mathbf{d}} g_{\mathbf{d}}(\mathbf{x}_i) + b^l \} + \varepsilon_i^{kl} \geq 1, \qquad \forall l \in K \setminus \{k\}, i \in X^k, k \in K, \tag{30}$$

$$\mathbf{w}^k \in \mathbb{R}^{|\acute{D}^k|}, b^k \in \mathbb{R}, \qquad \forall k \in K,$$

$$\varepsilon_i^{kl} \in \mathbb{R}_+, \qquad \forall l \in K \setminus \{k\}, i \in X^k, k \in K,$$

where $0 < \delta \ll 1$. The constant δ prevents $\delta + |w^k_{\mathbf{d}}|$ from being zero. The relation with the L_0-MSCP($\acute{D}$) is due to the form of the logarithm function at the objective function (29). A logarithmic function quickly falls

below zero near zero, but it increases only slowly when the domain value increases. Thus, decreasing $w_{\mathbf{d}}^k$ significantly decreases the objective value, while increasing $w_{\mathbf{d}}^k$ only increases it slightly. This means that, at the same training error, it is better to set as many $w_{\mathbf{d}}^k$ variables as possible to zero. Hence, we solve the above problem to obtain a solution of the L_0-MSCP($\acute{D}$).

The above problem can be reformulated by replacing $w_{\mathbf{d}}^k$ with $w_{\mathbf{d}}^{k+} - w_{\mathbf{d}}^{k-}$, where $w_{\mathbf{d}}^{k+} \geq 0$ and $w_{\mathbf{d}}^{k-} \geq 0$, as follows:

$$\min \quad h(\mathbf{w}^+, \mathbf{w}^-, \varepsilon) := \sum_{k\in K} \sum_{\mathbf{d}\in \acute{D}^k} \ln\{\delta + (w_{\mathbf{d}}^{k+} + w_{\mathbf{d}}^{k-})\} + C \sum_{k\in K} \sum_{i\in X^k} \sum_{l\in K\setminus\{k\}} \varepsilon_i^{kl} \tag{31}$$

$$\text{s.t.} \sum_{\mathbf{d}\in \acute{D}^k} (w_{\mathbf{d}}^{k+} - w_{\mathbf{d}}^{k-}) g_{\mathbf{d}}(\mathbf{x}_i) + b^k - \{\sum_{\mathbf{d}\in \acute{D}^l} (w_{\mathbf{d}}^{l+} - w_{\mathbf{d}}^{l-}) g_{\mathbf{d}}(\mathbf{x}_i) + b^l\} + \varepsilon_i^{kl} \geq 1,$$

$$\forall l \in K \setminus \{k\}, i \in X^k, k \in K, \tag{32}$$

$$\mathbf{w}^{k+}, \mathbf{w}^{k-} \in \mathbb{R}_+^{|\acute{D}^k|}, b^k \in \mathbb{R}, \qquad \forall k \in K,$$

$$\varepsilon_i^{kl} \in \mathbb{R}_+, \qquad \forall l \in K \setminus \{k\}, i \in X^k, k \in K.$$

Because the objective function (31) is a continuously differentiable concave function, the Frank–Wolfe algorithm (Frank and Wolfe, 1956) can be used to find a local optimal solution as follows:

1. Choose an initial solution, $\hat{D}^k := \emptyset$ for all $k \in K$, $w_{\mathbf{d}}^{k+,(0)} := 1$ and $w_{\mathbf{d}}^{k-,(0)} := 1$ for all $\mathbf{d} \in \acute{D}^k$, $k \in K$, $\varepsilon_i^{kl,(0)} := 1$ for all $l \in K \setminus \{k\}$, $i \in X^k$, $k \in K$, and $h^{(0)} := h(\mathbf{w}^{+,(0)}, \mathbf{w}^{-,(0)}, \varepsilon^{(0)})$, and let $r := 1$.

2. Determine a search direction, $p^{(r)}$.

   The following approximation problem is obtained by replacing the objective function (31) with its first-order Taylor expansion of $h(\mathbf{w}^+, \mathbf{w}^-, \varepsilon)$ around $(\mathbf{w}^{+,(r-1)}, \mathbf{w}^{-,(r-1)}, \varepsilon^{(r-1)})$: $h^{(r-1)} + \nabla h(\mathbf{w}^{+,(r-1)}, \mathbf{w}^{-,(r-1)}, \varepsilon^{(r-1)})^T \cdot (\mathbf{y}^+ - \mathbf{w}^{+,(r-1)}, \mathbf{y}^- - \mathbf{w}^{-,(r-1)}, \mathbf{z} - \varepsilon^{(r-1)})$.

$$[\mathrm{SD}^{(r)}] \quad \min \quad h^{(r-1)} + \sum_{k\in K} \sum_{\mathbf{d}\in \acute{D}^k} \frac{y_{\mathbf{d}}^{k+} + y_{\mathbf{d}}^{k-} - w_{\mathbf{d}}^{k+,(r-1)} - w_{\mathbf{d}}^{k+,(r-1)}}{\delta + w_{\mathbf{d}}^{k+,(r-1)} + w_{\mathbf{d}}^{k+,(r-1)}}$$

$$+ C \sum_{k\in K} \sum_{i\in X^k} \sum_{l\in K\setminus\{k\}} (z_i^{kl} - \varepsilon_i^{kl,(r-1)}) \tag{33}$$

$$\text{s.t.} \sum_{\mathbf{d}\in \acute{D}^k} (y_{\mathbf{d}}^{k+} - y_{\mathbf{d}}^{k-}) g_{\mathbf{d}}(\mathbf{x}_i) + b^k - \{\sum_{\mathbf{d}\in \acute{D}^l} (y_{\mathbf{d}}^{l+} - y_{\mathbf{d}}^{l-}) g_{\mathbf{d}}(\mathbf{x}_i) + b^l\} + z_i^{kl} \geq 1,$$

$$\forall l \in K \setminus \{k\}, i \in X^k, k \in K, \tag{34}$$

$$\mathbf{y}^{k+}, \mathbf{y}^{k-} \in \mathbb{R}_+^{|\acute{D}^k|}, b^k \in \mathbb{R}, \qquad \forall k \in K,$$

$$z_i^{kl} \in \mathbb{R}_+, \qquad \forall l \in K \setminus \{k\}, i \in X^k, k \in K,$$

   which is a linear programming problem. Let $(\hat{\mathbf{y}}^+, \hat{\mathbf{y}}^-, \hat{\mathbf{b}}, \hat{\mathbf{z}})$ be an optimal solution. Let $\mathbf{C}$ be an $m(c-1)$ sized column vector whose entries are all C, and for $k \in K$, let $\mathbf{V}^k$ be a $|D^k|$ sized column vector whose entries are $1/(\delta + w_{\mathbf{d}}^{k+,(r-1)} + w_{\mathbf{d}}^{k-,(r-1)})$, $d \in D^k$. Define $\mathbf{V}$ as the vector $[\mathbf{V}^{1T}, \ldots, \mathbf{V}^{cT}]$, and

Table 1: Data sets used for the experiments

Data set	#Classes	#Attributes	#Instances
Wine	3	13	178
Iris	3	4	150
Glass	7	9	214
Segment	7	19	2310
Balance	3	4	625
Thyroid	3	5	215

then $\nabla h(\mathbf{w}^{+,(r-1)}, \mathbf{w}^{-,(r-1)}, \varepsilon^{(r-1)})^T = (\mathbf{V}^T, \mathbf{V}^T, \mathbf{C}^T)$. The search direction is $\mathbf{p}^{(r)} := (\hat{\mathbf{y}}^+ - \mathbf{w}^{+,(r-1)}, \hat{\mathbf{y}}^- - \mathbf{w}^{-,(r-1)}, \hat{\mathbf{z}} - \varepsilon^{(r-1)})$.

3. Determine the step length $\lambda^{(r)}$, such that

$$\lambda^{(r)} = \underset{\lambda \in [0,1]}{\operatorname{argmin}} \; h\left((\mathbf{w}^{+,(r-1)}, \mathbf{w}^{-,(r-1)}, \varepsilon^{(r-1)}) + \lambda \mathbf{p}^{(r)}\right), \tag{35}$$

where $\hat{\lambda}^{(r)}$ is an optimal solution. Since the function $h(\mathbf{w}^+, \mathbf{w}^-, \varepsilon)$ is concave, the optimal solution $\hat{\lambda}^{(r)}$ is either 1 or 0.

4. New iteration point:
If $h^{(r-1)} > h(\hat{\mathbf{y}}^+, \hat{\mathbf{y}}^-, \hat{\mathbf{z}})$, then $\hat{\lambda}^{(r)} := 1$ and $(\mathbf{w}^{+,(r)}, \mathbf{w}^{-,(r)}, \mathbf{b}^{(r)}, \varepsilon^{(r)}) := (\hat{\mathbf{y}}^+, \hat{\mathbf{y}}^-, \hat{\mathbf{b}}, \hat{\mathbf{z}})$. Otherwise, $\hat{\lambda}^{(r)} := 0$ and $(\mathbf{w}^{+,(r)}, \mathbf{w}^{-,(r)}, \mathbf{b}^{(r)}, \varepsilon^{(r)}) := (\mathbf{w}^{+,(r-1)}, \mathbf{w}^{-,(r-1)}, \mathbf{b}^{(r-1)}, \varepsilon^{(r-1)})$.

5. Set $h^{(r)} := h(\mathbf{w}^{+,(r)}, \mathbf{w}^{-,(r)}, \varepsilon^{(r)})$.

6. If $h^{(r-1)} \leq h^{(r)}$ then go to step 7; otherwise, set $r := r + 1$ and go to step 2.

7. Set $\hat{\mathbf{w}}^{k+} := \mathbf{w}^{k+,(r)}$, $\hat{\mathbf{w}}^{k-} := \mathbf{w}^{k-,(r)}$, $\hat{b}^k := b^{k,(r)}$, and $\hat{D}^k := \{\mathbf{d} \in \acute{D}^k : \hat{w}^{k+}_{\mathbf{d}} \neq 0 \text{ or } \hat{w}^{k-}_{\mathbf{d}} \neq 0\}$ for all $k \in K$.

8. Return the classifier $f(\mathbf{x}) = \underset{k \in K}{\operatorname{argmax}}\Big(f_k(\mathbf{x}) = \sum_{\mathbf{d} \in \hat{D}^k} (\hat{w}^{k+}_{\mathbf{d}} - \hat{w}^{k-}_{\mathbf{d}}) g_{\mathbf{d}}(\mathbf{x}) + \hat{b}^k\Big)$.

In our implementation, we set $\delta = 10^{-12}$.

Computational experiments

Computational setting

We conducted experiments on the well-known multi-class classification problems with the aim of evaluating the performance of the methods proposed in the preceding sections. Six data sets were obtained from the UCI repository of machine learning databases (Frank and Asuncion, 2010). Table 1 provides the description of these data sets.

The performances of the proposed methods were compared with those of two multi-class SVM methods, namely, Weston and Watkins's multi-class SVM (Weston and Watkins, 1999) and Crammer and Singer's multi-class SVM (Crammer and Singer, 2002), both of which directly solve multi-class classification problems. The L_1-MSC and the L_0-MSC methods were implemented with the Xpress Mosel language using the linear programming solver provided by the Xpress package (Xpress, 2010). The multi-class SVMs were implemented in the BSVM software package (Hsu and Lin, 2006) using the decomposition method proposed by Hsu and Lin (2002*b*). We also implemented the predictions of all methods in $C_\sharp$ to measure prediction time in the same condition.

For the proposed methods, we have to translate input data into the range $[1, \infty)$, because the signomial function (1) is a sum of signomial terms where each term consists of products of power functions of $\mathbf{x}_j$, i.e, $\prod_{j=1}^{n} x_j^{d_j}$. In the column generation process, the terms are treated as products of exponential functions of d_j, and the partial derivatives of a sum of the terms are used in the Frank–Wolf algorithm. The sign of the partial derivative varies with the value of $\mathbf{x}$. By definition, the variable vector $\mathbf{x}$ defined on a signomial function (1) is required to be strictly positive. Therefore, the data were translated into the range $[1, \infty)$ to ensure that the signs of the partial derivatives remain steady: i.e, if the $\mathbf{x}$ value is in the range of $[1, \infty)$, then the sign of the partial derivative is always positive. Let $\mathrm{Min}_j := \min_{i=1,\ldots,m} x_{ij}$ for $j = 1, \ldots, n$. We translated the original data in such a way that if $\mathrm{Min}_j < 0$, then $x_{ij} := x_{ij} - \mathrm{Min}_j + 1$, otherwise $x_{ij} := x_{ij} + 1$. Note that the original data were translated without data scaling.

For the L_1-MSC and the L_0-MSC, we used the original data translated to be in the range $[1, \infty)$. For the multi-class SVMs, we used the scaled data which were linearly scaled to be in the range of $[-1, 1]$, as suggested by Hsu and Lin (2002*a*).

Additional experiments were conducted to determine the effect of data scaling on the performance of each method. To this end, we tested the multi-class SVMs on the original data used for the proposed methods (translated data), and the proposed methods were tested on the scaled data used for the multi-class SVMs. It should be noted that the scaled data were also translated by adding 2 so that they are in the range of $[1, 3]$ for the proposed methods.

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 in the ratio of 5:3:2, while ensuring that propositions of classes were similar in each subset. For various parameter settings, classifiers were trained on the training sets and then evaluated by using the corresponding validation sets. Those model parameters which achieved the best average accuracy on the validation sets were selected. We then predicted the test sets using the classifiers that are trained on the corresponding training sets with the selected parameters.

As performance criteria, we used the average classification accuracy on test sets, the average number of terms of resulting classifiers (in the case of the multi-class SVMs, the average proportion of support vectors in the training set), the average time to train classifiers, and the average time to predict the class of test sets.

For the proposed methods, we defined the set $D^k := \{\mathbf{d} \in \mathbb{R}^n : -1 \leq d_j \leq 1, j = 1, ..., n, \sum_{i=1}^{n} |d_i| \leq 1, 10\mathbf{d} \in \mathbb{Z}^n\}$ for $k \in K$. We tested the proposed methods using seven combinations of regularization parameter

C: C=[10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3]. For the multi-class SVMs, we used Gaussian radial basis function (RBF) kernel $K(x_i, x_j) := exp(-\gamma\|x_i - x_j\|^2)$. The multi-class SVMs were tested using 7×7 combinations of regularization parameter C and kernel parameter γ: C=[10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3] and γ=[10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3].

Computational results

The experimental results according to the respective performance criteria are presented in Tables 2–4. TestAcc, No. of terms, SVs/TrnS, TrnT, and PrdT denote the average classification accuracy and standard deviation on test sets, the average number of terms and standard deviation in the resulting classifiers, the average proportion of support vectors and standard deviation in the training set, the average time to train classifiers, and the average time to predict test sets, respectively. In Table 3, ΔTestAcc and ΔNo. of terms denote changes in the results presented in Tables 2 and 3. In Table 4, ΔTestAcc and ΔSVs/TrnS denote changes in the results presented in Tables 2 and 4. L_1-MSC and L_0-MSC denote the proposed methods, while W & W and C & S denote Weston and Watkins's (Weston and Watkins, 1999) and Crammer and Singer's (Crammer and Singer, 2002) multi-class SVMs, respectively.

To compare the performance of the proposed methods with the multi-class SVMs, we tested the proposed methods and the multi-class SVMs on the original data and the scaled data, respectively. Table 2 shows the performance results.

Overall, the L_1-MSC gave the best performance for the average classification accuracy. Compared with the multi-class SVMs, the L_1-MSC obtained better or competitive classification accuracies for most data sets (with the exception of the Wine data set), and the L_0-MSC achieved better or competitive classification accuracies for the Balance and the Thyroid data sets. In general, the classification accuracies of the L_1-MSC were better or, at the very least, comparable to those of the L_0-MSC, possibly because in the L_0-MSC, classifiers are trained after profitable exponents are generated, not at the same time.

Although the average classification accuracies of the L_0-MSC were worse than those of the L_1-MSC, the L_0-MSC gave much sparser classifiers than the L_1-MSC. There were, on average, 50% fewer terms in the resulting classifiers of the L_0-MSC than in those of the L_1-MSC. For example, for one training set of the Iris data set, the following classifier was obtained by the L_1-MSC with $C = 1$:

$$f(x)_{\text{L}_1\text{-MSC}} = \underset{\{1,2,3\}}{\text{argmax}} \begin{pmatrix} f_1(x) = 0.40x_2^1 - 1.35x_3^1 + 18.9, \\ f_2(x) = 0.53x_1^1 + 12.73, \\ f_3(x) = 1.19x_3^{0.6}x_4^{0.4} + 2.27x_3^{0.8}x_4^{0.2} \end{pmatrix}.$$

Using the same training set, the L_0-MSC obtained a more sparse classifier with $C = 1$ as follows:

$$f(x)_{\text{L}_0\text{-MSC}} = \underset{\{1,2,3\}}{\text{argmax}} \begin{pmatrix} f_1(x) = -1.82x_3^1 + 30.57, \\ f_2(x) = 24.30, \\ f_3(x) = 5.78x_3^{0.6}x_4^{0.4} \end{pmatrix}.$$

Both classifiers gave same classification accuracy (100%) on the corresponding test set.

The resulting classifiers of the proposed methods can be explicitly described in original space and interpreted with the original variables x_j, while for those of the multi-class SVMs (Crammer and Singer, 2002; Weston and Watkins, 1999), it is not easy to obtain explicit function descriptions in original space. For instance, in the case of the above classifier of L_0-MSC, we found that the variable x_3 plays an important role in the classifier. The Iris data set consists of four variables: x_1-sepal length, x_2-sepal width, x_3-petal length and x_4-petal width. If $x_3 < 3.45$, then an example is assigned to class 1, or if $x_3 >= 3.45$ and $x_3^{0.6} x_4^{0.4} < 4.20$, then class 2, otherwise class 3. The variable x_4 also influences the classification. This influence is, however, smaller than that of x_3, and x_4 (petal width) is most probably closely related to x_3 (petal length). Thus, as the value of x_3 increases, the class of an example may move from class 1 to class 2 and then from class 2 to class 3.

Moreover, the proposed methods gave sparser classifiers than the multi-class SVMs. For example, for the Wine data set, the resulting classifiers of the proposed methods are expressed as a sum of a linear combination of about six signomial terms and a constant term b. On the other hand, those of the multi-class SVMs are expressed as a sum of a linear combination of over 60 kernel functions and a constant term b. Thus, the proposed methods should enable an easier interpretation of data than the multi-class SVMs.

In terms of the prediction time, proposed methods performed better than the multi-class SVMs, especially on the Segment data set, since the proposed methods provide sparser classifiers than the multi-class SVMs. For a similar reason, the average prediction time of the L_0-MSC was smaller than that of the L_1-MSC. For the average training time, the performance of the multi-class SVMs was better or comparable to that of the proposed methods. However, the proposed methods generally took only several seconds to train the classifiers, with the exception of the Segment data set, and even for the Segment data set, the average training time was less than 6 minutes. It is noteworthy that in the case of off-line classification, the training time is not important as the prediction time.

Tables 3 and 4 show the results of the additional experiments that were performed to determine the effect of data scaling on the performance of each method. Table 3 presents the results of the proposed methods tested on the scaled data (the range of $[1, 3]$) and shows that although the performance results are slightly worse in some cases, they are not significantly different from those of the experiments conducted on the original data (see Table 2). Table 4 shows the results of the multi-class SVMs tested on the original data (the range of $[1, \infty)$). It shows that the performance results generally did not change much compared with the results on the scaled data (see Table 2). However, for the Wine data set, the average classification accuracy worsened by about 18% when we used the original data instead of the scaled data.

Conclusion

We have proposed two multi-class classification methods using signomial function, namely, L_1-MSC and L_0-MSC. The classification accuracies of the proposed methods are better or, at the very least, comparable to those of the multi-class SVMs. Proposed methods also give sparse classifiers and provide an explicit description of

Table 2: Performance results: Average classification accuracy (%) and standard deviation on test set, average number of terms and standard deviation in the resulting classifiers, average training time (s), and average prediction time (ms). The L_1-MSC and L_0-MSC were tested on the original data with a linear translation, and the multi-class SVMs were tested on the scaled data. For each data set, better or comparable performance results of the proposed methods are printed in bold, as compared with those of the multi-class SVMs (Crammer and Singer, 2002; Weston and Watkins, 1999). For the average number of terms, however, the smallest number is printed in bold.

Data set		L_1-MSC	L_0-MSC	W & W	C & S
Wine	TestAcc	98.57 ± 1.73	96.29 ± 3.36	98.57 ± 1.73	98.71 ± 1.96
	No. of terms (SVs/TrnS)[a]	6.40 ± 0.75	**6.10 ± 1.52**	(59.69 ± 2.39/89)	(80.46 ± 1.94/89)
	TrnT	0.31	0.61	0.03	0.03
	PrdT	**0.70**	**0.39**	1.95	4.37
Iris	TestAcc	**98.17 ± 2.53**	97.00 ± 3.04	97.83 ± 2.48	97.00 ± 3.04
	No. of terms (SVs/TrnS)[a]	4.95 ± 1.23	**3.45 ± 0.89**	(27.92 ± 1.44/75)	(12.23 ± 1.83/75)
	TrnT	0.15	0.21	0.02	0.08
	PrdT	**0.55**	**0.47**	1.64	1.09
Balance	TestAcc	**99.84 ± 0.56**	**99.56 ± 0.72**	96.41 ± 1.88	94.15 ± 1.48
	No. of terms (SVs/TrnS)[a]	16.20 ± 0.95	**11.25 ± 1.02**	(40.15 ± 2.08/312)	(39.23 ± 1.92/312)
	TrnT	1.66	1.86	0.06	0.38
	PrdT	**1.09**	**0.55**	1.40	3.12
Thyroid	TestAcc	**98.10 ± 1.83**	**96.55 ± 2.84**	96.19 ± 3.22	96.55 ± 2.73
	No. of terms (SVs/TrnS)[a]	8.15 ± 1.09	**6.05 ± 1.00**	(24.46 ± 1.33/109)	(21.23 ± 2.17/109)
	TrnT	0.35	0.43	0.02	0.02
	PrdT	**0.55**	**0.55**	1.40	1.40
Glass	TestAcc	**71.75 ± 5.26**	70.25 ± 6.33	71.75 ± 5.97	71.13 ± 6.56
	No. of terms (SVs/TrnS)[a]	31.85 ± 2.28	**20.95 ± 2.72**	(97.92 ± 3.23/109)	(109.00 ± 0.00/109)
	TrnT	2.06	59.59	0.04	0.03
	PrdT	**1.79**	**1.01**	3.35	4.45
Segment	TestAcc	**96.94 ± 0.69**	95.94 ± 1.01	96.93 ± 0.75	96.86 ± 0.96
	No. of terms (SVs/TrnS)[a]	68.95 ± 4.16	**22.00 ± 2.71**	(283.85 ± 9.93/1155)	(229.54 ± 6.97/1155)
	TrnT	151.56	346.95	0.33	1.14
	PrdT	**2.73**	**1.40**	170.26	137.12

[a]For the multi-class SVMs, the average proportion of support vectors and standard deviation in the training set is reported instead of the average number of terms and standard deviation in the resulting classifers and is parenthesized, since it is not easy to obtain explicit function descriptions of classifiers.

Table 3: Performance results of the L_1-MSC and the L_0-MSC, tested on the scaled data: Average classification accuracy (%) and standard deviation on test set, change (%) in the average classification accuracy on test set of Tables 2 and 3, average number of terms and standard deviation in the resulting classifiers, and change in the average number of terms in the resulting classifiers of Tables 2 and 3.

Data set	L_1-MSC				L_0-MSC			
	TestAcc	ΔTestAcc	No. of terms	ΔNo. of terms	TestAcc	ΔTestAcc	No. of terms	ΔNo. of terms
Wine	97.14 ± 2.62	−1.43	14.30 ± 1.84	7.90	97.14 ± 2.78	0.86	6.00 ± 1.56	−0.10
Iris	95.83 ± 2.84	−2.33	3.15 ± 0.88	−1.80	96.17 ± 2.92	−0.83	2.00 ± 0.00	−1.45
Balance	99.80 ± 0.44	−0.04	18.35 ± 0.81	2.15	99.56 ± 0.72	0.00	11.60 ± 1.23	0.35
Thyroid	96.67 ± 2.93	−1.43	7.50 ± 0.69	−0.65	97.02 ± 2.30	0.48	5.10 ± 0.64	−0.95
Glass	70.75 ± 7.39	−1.00	39.80 ± 3.14	7.95	68.75 ± 6.26	−1.50	28.50 ± 2.96	7.55
Segment	96.46 ± 0.80	−0.48	63.85 ± 3.69	−5.10	96.07 ± 0.78	0.13	27.00 ± 2.41	5.00

Table 4: Performance results of the multi-class SVMs (Crammer and Singer, 2002; Weston and Watkins, 1999), which were tested on the original data with a linear translation: Average classification accuracy (%) and standard deviation on test set, change (%) in the average classification accuracy on test set of Tables 2 and 4, average proportion of support vectors and standard deviation in the training set, and change in the average proportion of support vectors in the training set of Tables 2 and 4.

Data set	W & W				C & S			
	TestAcc	ΔTestAcc	SVs/TrnS	ΔSVs/TrnS	TestAcc	ΔTestAcc	SVs/TrnS	ΔSVs/TrnS
Wine	80.86 ± 6.82	−17.71	66.15 ± 2.27/89	6.46/89	80.57 ± 6.59	−18.14	66.15 ± 2.58/89	−14.31/89
Iris	98.50 ± 2.02	0.67	17.54 ± 1.71/75	−10.38/75	98.00 ± 2.51	1.00	15.15 ± 2.15/75	2.92/75
Balance	97.30 ± 1.18	0.89	45.69 ± 2.87/312	5.54/312	97.38 ± 1.33	3.23	37.85 ± 3.01/312	−1.38/312
Thyroid	95.84 ± 3.08	−0.36	21.62 ± 2.22/109	−2.85/109	95.95 ± 2.90	−0.59	20.00 ± 2.58/109	−1.23/109
Glass	71.50 ± 5.58	−0.25	93.85 ± 2.94/109	−4.08/109	72.38 ± 7.09	1.25	109.00 ± 0.00/109	0.00/109
Segment	95.39 ± 1.23	−1.54	540.85 ± 13.64/1155	257.00/1155	96.37 ± 1.18	−0.49	515.77 ± 13.65/1155	286.23/1155

the classifier in original space. Moreover, the proposed methods do not require an elaborated data scaling as a preprocessing process, since the methods are robust to the scaling of the input data.

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 (2012-006351).

References

Amaldi E and Kann V (1998). On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theoretical Computer Science* **209**(1–2): 237–260.

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

Baesens B, Mues C, Martens D and Vanthienen J (2009). 50 years of data mining and or: upcoming trends and challenges. *The Journal of the Operational Research Society* **60**(S1): S16–S23.

Bennett K P and Mangasarian O L (1994). Multicategory discrimination via linear programming. *Optimization Methods and Software* **3**(1): 27–39.

Bertsimas D and Tsitsiklis J (1997). *Introduction to linear optimization*. Athena Scientific: Belmont, USA.

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

Choo E U and Wedley W C (1985). Optimal criterion weights in repetitive multicriteria decision-making. *The Journal of the Operational Research Society* **36**(11): 983–992.

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.

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

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.

Frank A and Asuncion A (2010). UCI machine learning repository. http://archive.ics.uci.edu/ml.

Frank M and Wolfe F (1956). An algorithm for quadratic programming. *Naval Research Logistics Quarterly* **3**: 95–110.

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

Fung G M and Mangasarian O L (2004). A feature selection newton method for support vector machine classification. *Computational Optimization and Applications* **28**(2): 185–202.

Galar M, Fernández A, Barrenechea E, Bustince H and Herrera F (2011). An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes. *Pattern Recognition* **44**(8): 1761–1776.

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, USA.

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, USA, pp 507–513.

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.

Hsu C W and Lin C J (2002*a*). A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks* **13**(2): 415–425.

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

Hsu C W and Lin C J (2006). *BSVM: A SVM library for the solution of large classification and regression problems*. Software available at http://www.csie.ntu.edu.tw/∼cjlin/bsvm/.

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

Hüllermeier 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.

Lam K F and Moy J W (1996). Improved linear programming formulations for the multi-group discriminant problem. *The Journal of the Operational Research Society* **47**(12): 1526–1529.

Lee K, Kim N and Jeong M (2012). The sparse signomial classification and regression model. *to appear in Annals of Operations Research*.

Lee Y, Lin Y and Wahba G (2004). Multicategory support vector machines: Theory and application to the classification of microarray data and satellite radiance data. *Journal of the American Statistical Association* **99**(465): 67–82.

Lorena A C, Carvalho A C and Gama J M (2008). A review on the combination of binary classifiers in multiclass problems. *Artificial Intelligence Review* **30**(1–4): 19–37.

Mangasarian O L (1999). Arbitrary-norm separating plane. *Operations Research Letters* **24**(1–2): 15–23.

Pavur R and Loucopoulos C (1995). Examining optimal criterion weights in mixed integer programming approaches to the multiple-group classification problem. *The Journal of the Operational Research Society* **46**(5): 626–640.

Psorakis I, Damoulas T and Girolami M A (2010). Multiclass relevance vector machines: sparsity and accuracy. *IEEE Transactions on Neural Networks* **21**(10): 1588–1598.

Tax D M J and Duin R P W (2002). Using two-class classifiers for multiclass classification. In: Proceedings of the 16th International Conference on Pattern Recognition, ICPR '02. IEEE Computer Society: Quebec, Canada, pp 124–127.

Vapnik V N (1998). *Statistical learning theory*. Wiley: New York, USA.

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* **3**: 1439–1461.

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, pp 219–224.

Xpress (2010). Xpress-MP 7. http://www.fico.com/en.

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, Canada, pp 49–56.