J Comb Optim
DOI 10.1007/s10878-008-9189-2

Selective support vector machines

Onur Seref · O. Erhun Kundakcioglu · Oleg A. Prokopyev · Panos M. Pardalos

Abstract In this study we introduce a generalized support vector classification problem: Let X_i, $i = 1, \ldots, n$ be mutually exclusive sets of pattern vectors such that all pattern vectors $x_{i,k}$, $k = 1, \ldots, |X_i|$ have the same class label y_i. Select only one pattern vector x_{i,k^*} from each set X_i such that the margin between the set of selected positive and negative pattern vectors are maximized. This problem is formulated as a quadratic mixed 0-1 programming problem, which is a generalization of the standard support vector classifiers. The quadratic mixed 0-1 formulation is shown to be $\mathcal{NP}$-hard. An alternative approach is proposed with the free slack concept. Primal and dual formulations are introduced for linear and nonlinear classification. These formulations provide flexibility to the separating hyperplane to identify the pattern vectors with large margin. Iterative elimination and direct selection methods are developed to select such pattern vectors using the alternative formulations. These methods are compared with a naïve method on simulated data. The iterative elimination method is also applied to neural data from a visuomotor categorical discrimination task to classify highly cognitive brain activities.

Keywords Classification · Support vector machines · Quadratic mixed 0-1 programming

O. Seref (✉)
Department of Business Information Technology, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061, USA
e-mail: seref@vt.edu

O.E. Kundakcioglu · P.M. Pardalos
Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL 32611, USA

O.A. Prokopyev
Department of Industrial Engineering, University of Pittsburgh, Pittsburgh, PA 15261, USA

1 Introduction

Support Vector Machines (SVM) are the state-of-the-art supervised machine learning methods (Vapnik 1995). SVM method is used to classify pattern vectors which are assumed to belong to two linearly separable sets from two different classes. The classification function is defined by a hyperplane that separates two classes. SVM classifier finds the hyperplane that maximizes the distance between the convex hulls of both classes. SVMs can be extended to classification of nonlinear data by implicitly embedding the original data in a nonlinear space using *kernel functions* (Shawe-Taylor and Cristianini 2004). SVM training is based on optimizing a quadratic convex function that is subject to linear constraints, which is a well-known problem with many efficient solutions (Bennet and Campbell 2000). Due to the strong statistical learning background and efficient implementations, SVM methods have found a wide spectrum of application areas recently ranging from pattern recognition (Lee and Verri 2002) and text categorization (Joachims 1998) to biomedicine (Brown et al. 2000; Noble 2004; Lal et al. 2004) and financial applications (Huang et al. 2004; Trafalis and Ince 2002).

In this study, we introduce the concept of selective classification which is a generalization of SVM classification. In selective classification, we consider n sets of positive and negative labeled pattern vectors with t pattern vectors in each set. All of the pattern vectors in a set share the same label. Given this input, the objective is to select a single pattern vector from each of the n sets such that the selected positive and negative pattern vectors produce the best possible solution for a binary classification problem P. In the SVM context, this classification problem P, given in Formulation 1, is the quadratic optimization problem that maximizes the margin between positive and negative pattern vectors. The standard SVM problem can be considered as a special case of selective SVM classification where $t = 1$.

Selective classification resembles the multiple instance learning (MIL) problem in its input (Dietterich et al. 1997). However, MIL involves classifying positive and negative *bags* of pattern vectors, where each bag contains a number of pattern vectors sharing the same label. Given a classification function for MIL problem, at least one pattern vector in a positive bag should be classified correctly for that bag to be counted as correctly classified. For a negative bag to be correctly classified, all of the pattern vectors in it should be classified correctly. The MIL problem is to find a classification function that obtains a high classification accuracy for the bags. The objective in selective classification is not classifying the bags. It is, rather, to select a single pattern vector from each set (bag) to maximize the margin between the selected positive and negative pattern vectors.

The selective classification problem poses a hard combinatorial optimization problem. In this paper we show that the selective SVM problem is $\mathcal{NP}$-hard. We provide alternative approaches to the hard selection. We introduce the *restricted free slack* concept, which provides flexibility to the hyperplane by decreasing the influence of the pattern vectors that are misclassified or very close to the hyperplane. The resulting optimization problem is also convex and quadratic with linear constraints, and therefore can be kernelized through its Lagrangian dual. We present theoretical results on how the restricted free slack is distributed among the pattern vectors. We introduce

algorithms based on these results. These algorithms are tested on simulated data and compared with naive methods. This algorithm is also tested on a neural database to improve the classification accuracy and the performance of an SVM based feature selection method.

The remainder of the chapter is organized as follows. In Sect. 2, we start with a general review of SVM method including the derivation of the fundamental formulations, which will be helpful for the subsequent sections. Then we introduce the concept of selective classification in Sect. 3, where the combinatorial selective classification problem is shown to be $\mathcal{NP}$-hard. The alternative formulations are discussed in Sect. 4. In Sect. 5, different algorithms based on the selective classification formulations are presented. In Sect. 6, computational results from the application of the proposed methods on simulated data as well as real-life neural data from a visuomotor categorical discrimination task are presented. Finally, we conclude our results in Sect. 7.

2 Support vector machines

Let $\mathbf{X}$ be a set of pattern vectors $\mathbf{x}_i \in \mathbb{R}^d$, with class labels $y_i \in \{1, -1\}$. The problem of classifying these pattern vectors is finding a function $f(\cdot)$ which correctly assigns a class label for a given pattern vector $\mathbf{x}$. We want to separate the positive and negative classes by a hyperplane $(\mathbf{w}, b) \in \mathbb{R}^d$. The distance between the hyperplane and the closest pattern vectors (support vectors) γ is called the *margin* of the separating hyperplane. This hyperplane is represented by the support vectors, which give the classifier its name, support vector machine (SVM). The hyperplane with the maximum margin is called the *maximum margin* hyperplane. Note that there is an inherent degree of freedom to represent the same hyperplane by multiplying $\mathbf{w}$ and b with $\lambda > 0$. Depending on the value of λ the functional distance between $\mathbf{x}$ and the hyperplane changes. Assuming that this value is greater than a constant value, such as a unit distance of 1, it is easy to show that the geometric margin becomes $1/\|\mathbf{w}\|$ (Cristianini and Shawe-Taylor 2000). The pattern vectors in the positive and negative classes may not be separable. Therefore, we introduce slack variables ξ_i for misclassified pattern vectors and penalize them by $C/2$ in the objective function for a chosen value of $C > 0$. Now we can formulate the problem as follows

$$\min \quad \|\mathbf{w}\|^2 + \frac{C}{2}\sum_{i=1}^{n} \xi_i^2 \tag{1a}$$

$$\text{subject to} \quad y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) \geq 1 - \xi_i \quad i = 1, \ldots, n. \tag{1b}$$

Although the above formulation finds a linear separating hyperplane, its Lagrangian dual can be modified to use nonlinear maps to embed the pattern vectors in a higher dimensional space in such a way that a hyperplane can separate the mapped pattern vectors in the embedded space. This embedding is done via the *kernel trick*. The mapping is defined over dot product *Hilbert spaces*. The Lagrangian dual of the soft margin classification problem provides a formulation in which dot products can be

introduced. The dot product can be replaced with kernels to introduce the nonlinear mapping. The Lagrangian dual of formulation (1) is given as follows

$$\max \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle - \frac{1}{2C} \sum_{i=1}^{n} \alpha_i^2 \tag{2a}$$

$$\text{subject to} \quad \sum_{i=1}^{n} y_i \alpha_i = 0, \tag{2b}$$

$$\alpha_i \geq 0 \quad i = 1, \ldots, n. \tag{2c}$$

The convenience of the dual formulation is that it has only one constraint, and the dot products allow nonlinear kernels $K(\mathbf{x}_i, \mathbf{x}_j)$ to be incorporated in the formulation by replacing the dot product $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$. In this study, we use the *Gaussian kernel*, which is one of the most common kernels used both theoretically and practically. Gaussian kernel is given as follows, where the parameter σ is the *bandwidth*.

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left\{ \frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma} \right\}. \tag{3}$$

Next, we introduce selective support vector machines, starting with the selective margin maximization problem, and various relaxations.

3 A combinatorial selective SVM problem

Here, we introduce a new combinatorial classification problem in which each pattern vector in a standard classification problem is replaced with a set of possible pattern vectors sharing the same class label. We give the definition of the selective margin maximization problem as follows.

Definition 3.1 (Selective margin maximization problem) Let $\mathbf{X} = \{\mathbf{X}_1, \ldots, \mathbf{X}_n\}$ be sets of pattern vectors with t pattern vectors $\mathbf{x}_{i,1}, \ldots, \mathbf{x}_{i,t}$ in each set $\mathbf{X}_i$. Let $\mathbf{y} = \{y_1, \ldots, y_n\}$ be the corresponding labels for each set $\mathbf{X}_i$ with each pattern vector $\mathbf{x}_{i,k}$ having the same label y_i. Choose exactly one pattern vector $\mathbf{x}_{i,k^*}$ from each set $\mathbf{X}_i$ such that the margin between the selected positive and negative pattern vectors is maximized.

The selective margin maximization problem can be formulated as a quadratic mixed 0-1 programming problem as follows.

$$\min \quad \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{2} \sum_{i=1}^{n} \sum_{k=1}^{t} \xi_{i,k}^2 \tag{4a}$$

subject to

$$y_i(\langle \mathbf{w} \cdot \mathbf{x}_{i,k} \rangle + b) \geq 1 - \xi_{i,k} - M(1 - \upsilon_{i,k}) \quad i = 1, \ldots, n; \; k = 1, \ldots, t, \tag{4b}$$

$$\sum_{k=1}^{t} \upsilon_{i,k} = 1 \quad i = 1, \ldots, n, \tag{4c}$$

$$\upsilon_{i,k} \in \{0, 1\} \quad i = 1, \ldots, n;\ k = 1, \ldots, t. \tag{4d}$$

Note that this formulation is similar to (1), except for the extra term $M(1 - \upsilon_{i,k})$ in (4b) and the new constraints (4c) and (4d). M is a sufficiently large positive number. Binary variables $\upsilon_{i,k}$ indicate whether kth pattern vector from set i is selected or not. Note that when $\upsilon_{i,k} = 0$, the right side of (4b) becomes sufficiently small such that the constraint is always satisfied, which is equivalent to removing the point from the training set. Constraint (4c) ensures that only one pattern vector is included from each set.

It is clear that for sufficiently high penalty C, the selective SVM formulation can be considered as a hard selection problem without the slack variables ξ_i, whose solution would provide a hyperplane that can completely separate the selected positive and negative pattern vectors. Now, consider the following decision problem:

Definition 3.2 (Decision selective SVM (*D-SSVM*) problem) Let $\mathbf{X}_i = \{\mathbf{x}_{i,j}\}$ denote a set of d-dimensional vectors, where $j = 1, \ldots, t$. Assume that there are n such sets and all vectors $\mathbf{x}_{i,j}$ in each set $\mathbf{X}_i$ are labeled with the same label $y_i \in \{+1, -1\}$. Let $\boldsymbol{\upsilon}^*$ denote a selection where a single vector $\mathbf{x}_{i,j^*}$ is selected from each set $\mathbf{X}_i$. Is there a selection $\boldsymbol{\upsilon}^*$ such that all positive and negative pattern vectors can be separated by a hyperplane $(\mathbf{w}, b)$?

Theorem 3.3 *D-SSVM is $\mathcal{NP}$-complete for $t \geq 2$.*

Proof It is clear that one can find a separating hyperplane in polynomial type, for example using a standard SVM formulation, to determine whether the positive and negative pattern vectors are separable in a selection $\boldsymbol{\upsilon}^*$, therefore the *D-SSVM* is in $\mathcal{NP}$. Next, we show that this decision problem is $\mathcal{NP}$-complete for $t \geq 2$ by a reduction from the classical PARTITION problem: Given a set of positive integers $S = \{s_1, s_2, \ldots, s_n\}$, does there exist a subset $S' \subseteq S$ such that

$$\sum_{i:s_i \in S'} s_i = \sum_{i:s_i \in S \setminus S'} s_i = \frac{1}{2} \sum_{i=1}^{n} s_i ? \tag{5}$$

This problem is known to be $\mathcal{NP}$-complete (Garey and Johnson 1979). Now, let us consider the following equivalent formulation of the PARTITION problem: Given a set of n positive integers $S = \{s_1, s_2, \ldots, s_n\}$, does there exist a vector $\mathbf{w} \in \{-1, +1\}^n$, such that $\sum_{i=1}^{n} s_i w_i = 0$?

Suppose we are given an instance of the PARTITION problem. Let $d = n + 1$. Let $\mathbf{e}_i$ be a d-dimensional vector whose components are zeros except for component i, which is equal to 1. Let s_+ and s_- be d-dimensional vectors such that $s_+ = (s_1, s_2, \ldots, s_n, 1)$ and $s_- = (s_1, s_2, \ldots, s_n, -1)$.

Next we construct an instance of the D-SSVM problem as follows.

1. For $i = 1, \ldots, n$ add the sets of vectors, $\{e_i, -e_i\}$ with positive labels, $\{-e_i, e_i\}$ with negative labels.
2. Add the sets of vectors $\{e_{n+1}, e_{n+1}\}$ with positive labels, $\{-e_{n+1}, -e_{n+1}\}$ with negative labels.
3. Add the sets of vectors $\{s_+, s_+\}$ with positive labels, $\{s_-; s_-\}$ with negative labels.

Note that, regarding item 1 of the construction, following are the corresponding inequalities in the selective SVM formulation.

$$w_i + b \geq 1 - M(1 - \upsilon_{i,1}), \tag{6a}$$

$$-w_i + b \geq 1 - M(1 - \upsilon_{i,2}), \tag{6b}$$

$$\upsilon_{i,1} + \upsilon_{i,2} = 1, \tag{6c}$$

$$w_i - b \geq 1 - M(1 - \upsilon'_{i,1}), \tag{6d}$$

$$-w_i - b \geq 1 - M(1 - \upsilon'_{i,2}), \tag{6e}$$

$$\upsilon'_{i,1} + \upsilon'_{i,2} = 1. \tag{6f}$$

It can be verified that (6a)–(6b) and (6d)–(6e) have a feasible solution if and only if

$$\upsilon_{i,1} = \upsilon'_{i,1} = 1 \quad \text{and} \quad \upsilon_{i,2} = \upsilon'_{i,2} = 0, \quad \text{or} \tag{7a}$$

$$\upsilon_{i,1} = \upsilon'_{i,1} = 0 \quad \text{and} \quad \upsilon_{i,2} = \upsilon'_{i,2} = 1. \tag{7b}$$

From item 2 of the construction we have

$$w_{n+1} + b \geq 1, \tag{8a}$$

$$w_{n+1} - b \geq 1. \tag{8b}$$

From the solution to the system of inequalities above, in order to minimize the objective $\sum_{i=1}^{d} w_i^2$ the values of w_i, $i = 1, \ldots, n$, can either be 1 or -1, the value of w_{n+1} should be 1, and b=0. From item 3 of the construction we have

$$\sum_{i=1}^{n} s_i w_i + w_{n+1} + b \geq 1, \tag{9a}$$

$$-\sum_{i=1}^{n} s_i w_i + w_{n+1} - b \geq 1. \tag{9b}$$

Taking into account our observations above, from (9a)–(9b) we can conclude that the objective $\sum_{i=1}^{d} w_i^2$ is equal to d if and only if $\sum_{i=1}^{n} s_i w_i = 0$.

The presented reduction is polynomial, therefore, the decision version of the selective SVM problem is $\mathcal{NP}$-complete. □

Corollary 3.4 *Selective SVM problem* (*Formulation* 4) *is* $\mathcal{NP}$*-hard.*

4 An alternative selective SVM problem

We introduce an alternative approach to the combinatorial selective SVM problem to find good solutions efficiently. The main idea is to provide some flexibility for the hyperplane. This flexibility is provided by *restricted free slack* to decrease the influence of the pattern vectors with small or negative (misclassified) distance from the hyperplane. Given this flexibility, the hyperplane realigns itself with respect to the further pattern vectors with larger margin.

We provide restricted free slack amount of V for all pattern vectors. Note that a very small amount of free slack would make a very small difference compared to the standard SVM formulation, whereas a very large free slack would yield trivial solutions. Depending on the selection scheme, the amount of total free slack may vary. The corresponding formulation is given as follows.

$$\min \quad \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{2}\sum_{i=1}^{n}\xi_{i,k}^2 \tag{10a}$$

subject to

$$y_i(\langle \mathbf{w}\cdot\mathbf{x}_{i,k}\rangle + b) \geq 1 - \xi_{i,k} - \upsilon_{i,k} \quad i = 1,\ldots,n;\ k = 1,\ldots,t, \tag{10b}$$

$$\sum_{i=1}^{n}\sum_{k=1}^{t}\upsilon_{i,k} \leq V, \tag{10c}$$

$$\upsilon_{i,k} \geq 0 \quad i = 1,\ldots,n;\ k = 1,\ldots,t. \tag{10d}$$

Note that this formulation is similar to the standard SVM formulation with a convex quadratic objective function and linear constraints. The Lagrangian dual of this formulation can also be derived for nonlinear classification. The dual formulation is given as follows.

$$\max \quad \left\{\sum_{i=1}^{n}\sum_{k=1}^{t}\alpha_{i,k} - \frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{t}\sum_{j=1}^{n}\sum_{l=1}^{t} y_i y_j \alpha_{i,k}\alpha_{j,l}\langle \mathbf{x}_{i,k}\cdot\mathbf{x}_{j,l}\rangle \right.$$
$$\left. - \frac{1}{2C}\sum_{i=1}^{n}\sum_{k=1}^{t}\alpha_{i,k}^2 - \beta V\right\} \tag{11a}$$

subject to

$$\sum_{i=1}^{n}\sum_{k=1}^{t} y_i\alpha_{i,k} = 0, \tag{11b}$$

$$0 \leq \alpha_{i,k} \leq \beta \quad i = 1,\ldots,n;\ k = 1,\ldots,t. \tag{11c}$$

Let $A = \{(i,k) : \alpha_{i,k} > 0\}$. From the optimality conditions, $\mathbf{w} = \sum_{(i,k)\in A} y_i\alpha_{i,k}\mathbf{x}_{i,k}$. Using complementary slackness, we know that if $0 < \alpha_{i,k}, \beta$ for any (i,k), then the corresponding constraint in formulation 10 is binding and $\upsilon_{i,k} = 0$. If such (i,k) exists, then $b = y_i(1 - \alpha_{i,k}/C) - \langle \mathbf{w}\cdot\mathbf{x}_{i,k}\rangle$. If there is no such (i,k), i.e., $\alpha_{i,k} = \beta$

for all $(i,k) \in A$, then using all of the corresponding constraints in formulatio 10 for $(i,k) \in A$ and Lemma 4.1, we find $b = (\sum_{(i,k)\in A} y_i)(|A|(1-\beta/C) - V - \sum_{(i,k)\in A} y_i \langle \mathbf{w} \cdot \mathbf{x}_{i,k} \rangle)$.

Kernel induced maps to nonlinear feature spaces can be used by replacing the linear dot product $\langle \mathbf{x}_{i,k} \cdot \mathbf{x}_{j,l} \rangle$ in (11) with a kernel $\mathbf{K}(\mathbf{x}_{i,k}, \mathbf{x}_{j,l})$. Then the classification function is given as $f(\mathbf{x}) = \text{sign}(\sum_{i=1}^{n} \sum_{k=1}^{t} y_i \alpha_{i,k} \langle \mathbf{x} \cdot \mathbf{x}_{i,k} \rangle + b)$.

Next, we show that the pooled free slack acquired by each pattern vector is either 0 or linearly proportional to its distance from the optimal hyperplane depending on the total slack provided. For the following Lemmas 4.1, 4.2, and Theorem 4.3, let $D = (1,\ldots,n) \times (1,\ldots,k)$ be the set of indices for all pattern vectors, and $\mathbf{w} \neq \mathbf{0}$, b, $\boldsymbol{\xi}$, $\boldsymbol{\upsilon}$ be the solution to problem (10), with an objective function value z^*.

Lemma 4.1 *Constraint* (10c) *is binding, i.e.,* $\sum_{(i,k)\in D} \upsilon_{i,k} = V$ *in the optimal solution.*

Proof Assume that $V - \sum_{(i,k)\in D} \upsilon_{i,k} > 0$ in the optimal solution, where $\mathbf{w} \neq \mathbf{0}$. From complementary slackness, the corresponding dual variable $\beta = 0$ in the dual formulation (11), which forces the dual objective, and thus the primal objective to be 0. This implies $\mathbf{w} = \mathbf{0}$, thus a contradiction. □

Lemma 4.2 *If* $\xi_{i,k} + \upsilon_{i,k} > 0$, *for some* $(i,k) \in D$ *then the corresponding constraint* (10b) *is binding, i.e.,* $y_i(\langle \mathbf{w} \cdot \mathbf{x}_{i,k} \rangle + b) = 1 - \xi_{i,k} - \upsilon_{i,k}$.

Proof Assume that $\xi_{i,k} + \upsilon_{i,k} > 0$ and the corresponding constraint is nonbinding, i.e. $y_i(\langle \mathbf{w} \cdot \mathbf{x}_{i,k} \rangle + b) - 1 + \xi_{i,k} + \upsilon_{i,k} = \Delta > 0$. Then, $\xi'_{i,k} = \xi_{i,k} - \Delta$ contradicts optimality, and $\upsilon'_{i,k} = \upsilon_{i,k} - \Delta$ contradicts Lemma 4.1. □

Theorem 4.3 *Let,* $\xi_{\max} = \max_{(i,k)\in D}\{\xi_{i,k}\}$ *in the optimal solution. Then,*

$$\textit{Let } (i,k) \in D, \textit{ then} \quad \xi_{i,k} < \xi_{\max} \Longrightarrow \upsilon_{i,k} = 0 \quad \textit{and} \quad \upsilon_{i,k} > 0 \Longrightarrow \xi_{i,k} = \xi_{\max}.$$

Proof Assume that $\boldsymbol{\xi}$ and $\boldsymbol{\upsilon}$ in the optimal solution does not necessarily satisfy Theorem (4.3). Let,

- $d_{i,k} = \xi_{i,k} + \upsilon_{i,k}$,
- $\upsilon'_{i,k} = \max_{(i,k)\in D}\{d_{i,k} - \xi'_{\max}, 0\}$, where $\xi'_{\max}$ is such that $\sum_{(i,k)\in D} \upsilon'_{i,k} = V$,
- $\xi'_{i,k} = d_{i,k} - \upsilon'_{i,k}$,
- $\delta_{i,k} = \xi_{i,k} - \xi'_{i,k}$.

Note that $\xi'_{\max}$, $\xi'_{i,k}$ and $\upsilon'_{i,k}$ values satisfy Lemmas 4.1 and 4.2, Theorem 4.3, and do not violate any of the constraints in (10). It is easy to verify that $\sum_{(i,k)\in D} \delta_{i,k} = 0$.

Let $S \subseteq D$ be the set of indices with $\upsilon'_{i,k} = 0$, and $z' = \|\mathbf{w}^*\|^2 + \sum_{(i,k)\in D} \xi'^2_{i,k}$. The objective function value, $z^* = \|\mathbf{w}^*\|^2 + \sum_{(i,k)\in D} \xi^2_{i,k}$, can be written as,

$$z^* = z' + \sum_{(i,k)\in S} 2\xi'_{i,k}\delta_{i,k} + \sum_{(i,k)\in D\setminus S} 2\xi'_{i,k}\delta_{i,k} + \sum_{(i,k)\in D} \delta^2_{i,k}. \tag{12}$$

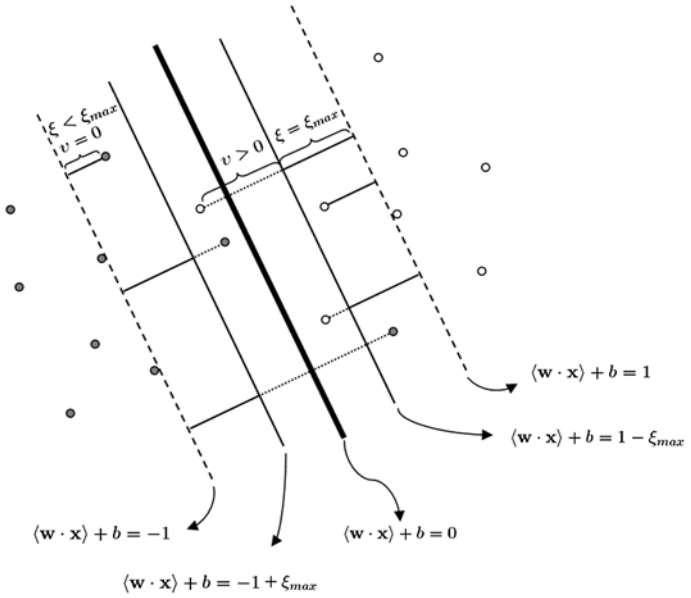

Fig. 1 Example showing the relationship between penalized slack and free slack

Note that $\delta_{i,k} \leq 0 \; \forall (i,k) \in D$, by definition, and $\xi'_{i,k} = \xi'_{\max} \; \forall (i,k) \in D \setminus S$. Since,

$$\sum_{(i,k)\in S} \xi'_{i,k}\delta_{i,k} \geq \xi'_{\max} \sum_{(i,k)\in S} \delta_{i,k},$$

and $\sum_{(i,k)\in D} \delta_{i,k} = 0$, the relationship between z^* and z' is,

$$z^* \geq z' + \sum_{(i,k)\in D} \delta_{i,k}. \tag{13}$$

From expression (13), z^* can only be optimal if and only if $\delta_{i,k} = 0$, and thus $\xi_{i,k} = \xi'_{i,k}$ and $\upsilon_{i,k} = \upsilon'_{i,k}$ for all $(i,k) \in D$. □

Theorem 4.3 basically states that all pattern vectors with a functional margin $d_{i,k} = yi(\langle \mathbf{w} \cdot \mathbf{x}_{i,k} \rangle + b) < 1$ incur penalty for $\xi_{i,k} = \min\{1 - d_{i,k}, \xi_{\max}\}$. For pattern vectors $\xi_{i,k} = \xi_{\max}$ the free slack is equal to $\upsilon_{i,k} = 1 - \xi_{\max} - d_{i,k}$, the sum of which is always equal to V. Examples are demonstrated in Fig. 1.

This result implies, without loss of generality, the free slack for a positive pattern vector is distributed linearly proportional to its distance from the hyperplane $\langle \mathbf{w} \cdot \mathbf{x}_{i,k} \rangle + b = 1 - \xi_{\max}$, as shown in Fig. 2. In this figure, free slack for each point is shown in the third dimension. The figure on the left is the top view showing the original data. The figures on the right are front views, only showing the amount of slack assigned.

This result leads to a few possible methods to maximize the margin between the selected points, which are discussed in the next section.

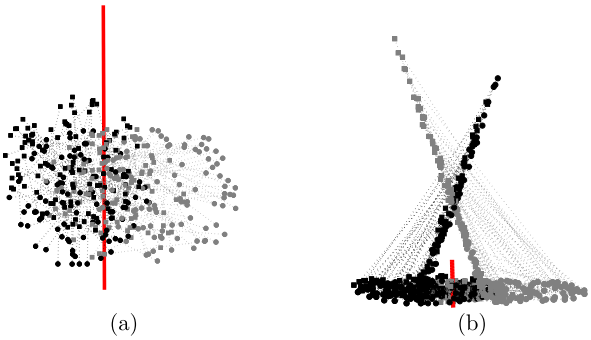

Fig. 2 Distribution of restricted free slack shown in the third dimension on a two dimensional data: (**a**) Top view. (**b**) Front view

5 Selection methods

The solution to the alternative problem allows pattern vectors that are close to the hyperplane to use free slack and provide more flexibility for the separating hyperplane. The selection is done regarding the orientation of the hyperplane. Since we do not use the combinatorial formulation, we refer to the alternative formulation as *soft selection* for the remainder of the paper. The methods introduced in this section are based on the soft selection formulation and the result which states that the amount of free slack acquired by each pattern vector is linearly proportional to its distance from the hyperplane. Two methods are proposed: an iterative elimination method, and a direct selection method.

5.1 Iterative elimination

The soft selection formulations are mainly developed to give the separating hyperplane more flexibility and, at the same time, to identify those pattern vectors which are misclassified or very close to the hyperplane. Such pattern vectors require more free slack among the points in their set. We can have a more separated subset of positive and negative pattern vectors if we remove such pattern vectors. Our intuitive basic approach is as follows: at each iteration, supply an incremental amount of free slack of n (1 unit per set), solve the soft selection problem, identify the pattern vector with the minimum distance for each set and remove it, and repeat the iterations with the updated set of pattern vectors until only one pattern vector per set remains. This approach is summarized in Algorithm 1.

In Algorithm 1, $\mathbf{X}(0)$ is the original input of pattern vectors, $\mathbf{y}$ is the vector of labels for each set $\mathbf{X}_i$, n is total the free slack amount provided for the soft selection problem, $(\mathbf{w}, b)$ is the hyperplane, the amount $y_i(\langle \mathbf{w} \cdot \mathbf{x}_{i,k} \rangle + b)$ is the distance of $\mathbf{x}_{i,k}$ from the hyperplane $(\mathbf{w}, b)$, $t(0)$ is the initial number of pattern vectors in each set, and r is the set of pattern vectors to be removed at each iteration. Note that this distance can be negative if the pattern vector is misclassified.

Note that when total free slack is zero, the soft selection problem reduces to a standard SVM problem. Based on this observation, we also consider a naïve elimination method, which is basically the iterative elimination method with zero total free slack at each iteration. The naïve elimination is included to compare the performance of the

Algorithm 1 Iterative elimination

$\mathbf{X} \Leftarrow \mathbf{X}(0)$
$t \Leftarrow t(0)$
while $t > 1$ **do**
  $\{\mathbf{w}, b\} \Leftarrow$ SOFT SELECTION$(\mathbf{X}, \mathbf{y}, n)$
  $r \Leftarrow \emptyset$
  for $i = 1$ to n **do**
    $k^* = \arg\min_{k=1,\ldots,t}\{y_i(\langle \mathbf{w} \cdot \mathbf{x}_{i,k} \rangle + b)\}$
    $r \Leftarrow r \cup \mathbf{x}_{i,k^*}$
  end for
  $\mathbf{X} \Leftarrow \mathbf{X} \setminus r$
  $t \Leftarrow t - 1$
end while
return $\mathbf{X}$

iterative elimination and direct selection methods to a standard SVM based approach, which does not depend on the soft selection formulation.

5.2 Direct selection

The alternative to the iterative elimination method is to provide enough free slack to eliminate $t - 1$ points in a single iteration. This time, for each set, we directly select the pattern vector with maximum distance from the hyperplane. The direct selection algorithm can be summarized as solving the soft selection problem with $n(t - 1)$ amount of total free slack, and from each set, returning the pattern vector furthest from the resulting hyperplane. The direct selection is summarized in Algorithm 2. The notation is similar to that of Algorithm 1.

Algorithm 2 Direct selection

$\{\mathbf{w}, b\} \Leftarrow$ SOFT SELECTION$(\mathbf{X}(0), \mathbf{y}, n(t-1))$
$\mathbf{X} \Leftarrow \emptyset$
for $i = 1$ to n **do**
  $k^* = \arg\max_{k=1,\ldots,t}\{y_i(\langle \mathbf{w} \cdot \mathbf{x}_{i,k} \rangle + b)\}$
  $\mathbf{X} \Leftarrow \mathbf{X} \cup \mathbf{x}_{i,k^*}$
end for
return $\mathbf{X}$

6 Computational results

In this section, we show the computational results of the proposed methods developed in Sect. 5. We start with the description of how the data is generated and how the performances of the methods are compared. Then, we present comparative results of the iterative elimination method, direct selection method and the naïve elimination method.

6.1 Simulated data and performance measure

The simulated data is generated using two parameters that determine the *dimensionality* and the *separability* of the pattern vectors. Let S_k, $k = 1, \ldots, t$ denote the set of pattern vectors formed by including the kth pattern vector from each set $\mathbf{X}_i$. For each S_k, n random pattern vectors are generated, uniformly distributed in a hypersphere with radius r. The center of each hypersphere is also distributed uniformly in a hypersphere with radius c. We keep r is constant so that c determines the separability of the data, which is the first parameter. The dimension of the data, denoted by d, is the second parameter. In Fig. 3 three instances with different separability values (a) $c = 0$ (b) $c = r/2$ and (c) $c = r$ are shown for $d = 2$.

For the simulated data, the performance measure is the objective function value obtained by the standard SVM formulation for the final set of selected pattern vectors. For this purpose we set the restricted total free slack to zero, in which case we obtain the standard SVM formulation. The results are later normalized for each combination of dimension d and separability c with all of the results obtained from the compared methods. The normalization is done by measuring the mean $\mu_{d,c}$ and the standard deviation $\sigma_{d,c}$ of all the objective function values obtained from all of the compared methods, and normalizing each objective function value using the mean and the standard deviation.

6.2 Iterative elimination vs. naïve elimination

Simulated data is generated as explained in Sect. 6.1 for $d = 2, 4, \ldots, 20$ and $c = 0, r/2, r$. Note that $t = 6$ and free slack parameter $p = 1$ (per set). For each combination of the parameters 100 instances of simulated data sets are generated and tested using iterative elimination and naïve elimination. The results are normalized as explained in Sect. 6.1. Let z_{PFS} and z_N denote the average normalized objective function values obtained from iterative elimination and naïve elimination.

In Fig. 4, the values $z_N - z_{PFS}$, for $d = 2, 4, \ldots, 20$ are plotted for each c value. It is clear from the figure that as the dimensionality increases the iterative elimination is significantly superior to the naïve elimination method. The difference becomes more apparent for higher levels of data separation. This result clearly shows the success of the iterative elimination due to the flexibility of the separating hyperplane incorporated by the restricted free slack.

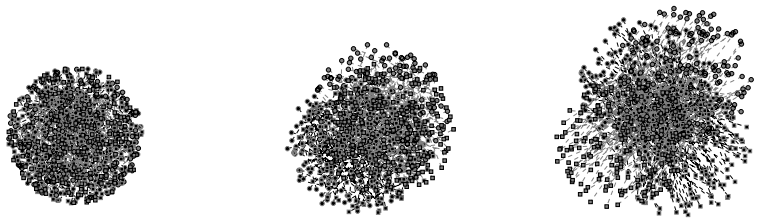

Fig. 3 2-D data with separability (**a**) $c = 0$, (**b**) $c = r/2$, (**c**) $c = r$

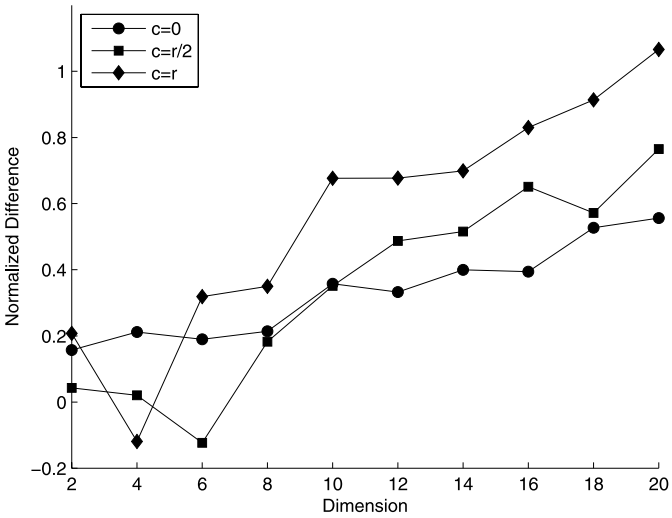

Fig. 4 Normalized difference between iterative elimination and naïve elimination methods

6.3 Direct selection

We generated and evaluated data as explained in Sect. 6.1 for $d = 2, 4, \ldots, 20$ and $c = 0, r/2, r$ for total slack parameter $p = 1, \ldots, 5$ with 100 instances each. There are $t = 6$ pattern vectors in each set. In Fig. 5, the effect of the increase in total slack is shown. The three graphs in the figure are in the order of increasing separation in the data. In each graph, the objective function values for the highest total slack parameter $p = 5$ is assumed to be the base value and the differences between the others and the base, $z_i - z_5$, $i = 1, \ldots, 4$ are graphed. The amount of free slack does not contribute significantly for completely overlapping data in graph (a). However, it is clear from graph (b) that when there is some separability in the data, increasing amount of slack improves the performance of the method for higher dimensions. This difference is even more amplified in graph (c) for higher separability values. In graphs (b) and (c), the increase in the difference between $p = 5$ and the others for higher dimensional data are also apparent. Based on these results, we can conclude that free slack parameter can be set as $t - 1$ (per set) for a dataset with t pattern vectors in each set.

Next we compare the direct selection method with iterative elimination. Again, that are $t = 6$ pattern vectors in each set. Free slack parameter for direct selection is $p = t - 1$. In Fig. 6, the performances of iterative elimination and direct selection are shown with the values $z_{DS} - z_{IE}$, where z_{IE} and z_{DS} are the normalized objective function values obtained from iterative elimination and direct selection methods, respectively. The results fluctuate and there is no significant dominance of one method over the other. However, we observe from the figure that, on the average, the iterative elimination method performs slightly better than the direct elimination method.

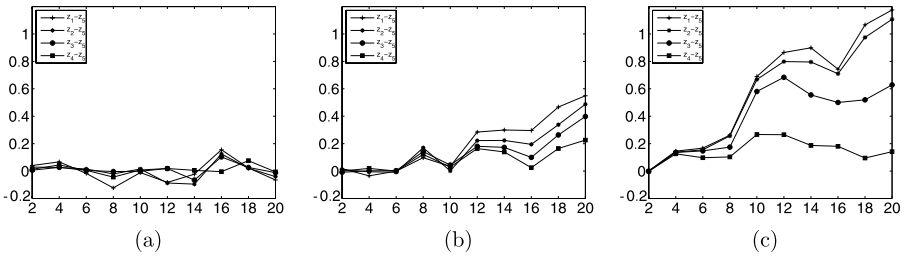

Fig. 5 Effect of the amount of free slack on data with separability (**a**) $c = 0$, (**b**) $c = r/2$, (**c**) $c = r$

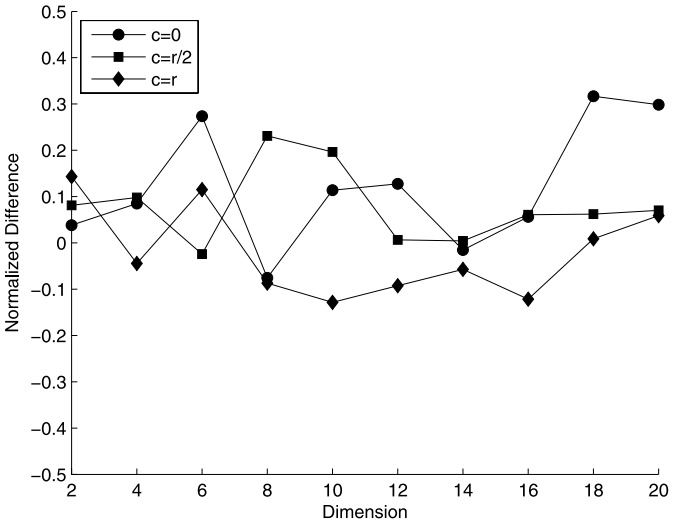

Fig. 6 Comparison of iterative elimination and direct selection methods

6.4 An application to a visuomotor pattern discrimination task

We applied the selective classification methods developed to a real life neuroscience problem. The neural data we study is the local field potentials (LFP) collected from multiple channels implanted in different cortical areas of a macaque monkey during a visual discrimination task. This task involves recognizing a visual *go* stimuli which is followed by a motor response. The visuomotor task is repeated multiple times for the same experiment with different stimuli-response combinations. These differences are grouped as different classes of data for classification. The main objective is to be able to detect these differences over the time course of the task, which requires extensive computational effort to achieve robust results from the multi-dimensional and highly nonlinear neural data.

The visual stimuli are designed to create *lines* and *diamonds*. The go stimuli is chosen to be either lines or diamonds from one session to another. We are interested in detecting different cognitive stages of the visual discrimination task over the time line. We distinguish different sets of labels for each cognitive stage. Three different

stages are anticipated: (i) the detection of the visual stimulus, (ii) the categorical discrimination of the stimulus, and (iii) the motor response. The first and the third stages are relatively easy to detect, however the second stage has not been detected in previous studies (Ledberg et al. 2007). This stage involves a complex cognitive process whose onset and length vary over time.

The classification is performed with the patter vectors collected at a specific time T^* from each trial. The classification accuracy obtained from each time point shows the time intervals when the two observed states of the monkey brain are different. However, there are temporal variations in each trial regarding the timing of the observed stages. The motivation behind the development of selective classification methods is to perform classification while accounting for these temporal variations in the underlying complex cognitive processes. The standard SVM classifier is hindered by the noisy recordings due to these temporal variations. We assume that one recording among the t recordings in a time window from each trial comes from the underlying cognitive process we want to detect. Selecting the most distinguishable recording from each trial at a given time window centered around T^* is a hard problem. Therefore we use the iterative elimination to detect and remove noisy recordings iteratively to achieve better recordings for the given time window.

The data consists of around 4000 trials. Because of the computational limitations of the optimization software (CPLEX 10.1), the entire data could not be processed simultaneously. Therefore we consider 200 trials at a time with equal numbers of positive and negative recordings. Nonlinear iterative elimination method is applied with a window of 3 recordings from each trial for each time point. This window correspond to 15 milliseconds. The recordings with the minimum distance is eliminated from each set at each iteration. This is repeated until there is only one pattern vector remains from each trial.

Each independent batch of 200 trials resulted in a consistently separated cumulative set of selected recordings. The classification accuracy of the selected recordings from each time window is evaluated with the standard SVM classifier using 10-fold classification. In Fig. 7(a), the comparison of the classification accuracy results from iterative elimination and the results from the standard SVM classification. The iterative elimination shows a consistent increase around 10%. This increase can be adjusted by the *baseline* approach. In order to create a baseline, we randomly assign class labels to pattern vectors and apply the iterative elimination methods, so that we can detect the increase in the accuracy for random data and subtract it from the original accuracy results. The baseline is also given in Fig. 7(a). The difference between the original accuracy results and the baseline results are given in Fig. 7(b). The peak around 160 milliseconds in this graph is very clear. This result matches the anticipated interval of the categorical discrimination stage. The second peak around 275 milliseconds is too late for the categorical differentiation, however would probably be related to post processing of the categorical difference.

In Fig. 8 the results for the feature (channel) selection are presented. We used an SVM based adaptive scaling method for feature selection. This method finds the channels that contribute to SVM classification. When adaptive scaling method is applied over the time line, it produces normalized weight vectors for each time point that can be transfered into a raster plot.

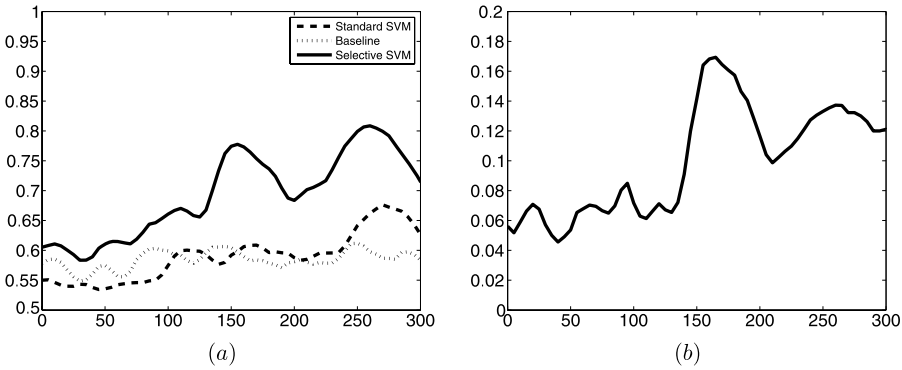

Fig. 7 Comparative classification accuracy results. (**a**) Standard SVM, baseline and after applying selective SVM. (**b**) Difference between the baseline and selective SVM results

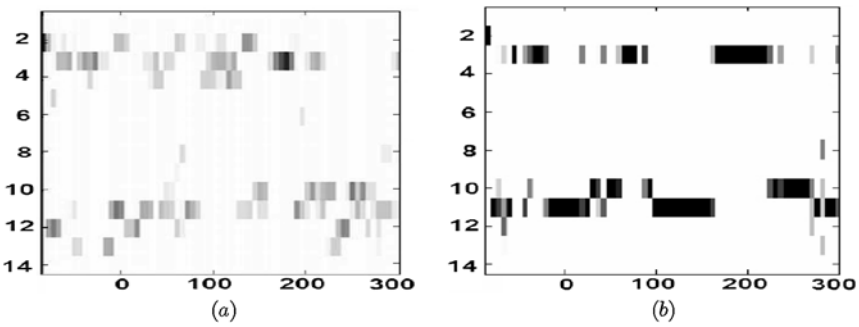

Fig. 8 Raster plots for the adaptive scaling feature selection method (**a**) after DTW applied, (**b**) after selective SVM applied

In Fig. 8(a) the results obtained without iterative elimination are presented. In this plot, channels are significantly intermittent over time and the overall picture is not conclusive. The raster plot in Fig. 8(b) shows the results obtained by iterative elimination. Due to the sparseness influence of the adaptive scaling method, we can clearly see the influence of three major channels on the data. We focus on the time intervals around the peaks observed in the classification accuracy graphs. The first peak corresponds to electrode 3, which is around the superior temporal gyrus. Physical damage in temporal lobe is known to impair visual discrimination (Horel and Misantone 1976; Mendola and Corkin 1999) and our results agree with the literature. The second peak corresponds to electrode 10 which is close to the inferior parietal lobe, which also is known to be involved in visual discrimination and have a complementary role with the temporal lobe (Eacott and Gaffan 1991).

7 Conclusion

In this paper, we introduce a novel selective classification method which is a generalization of the standard SVM classifier. Sets of pattern vectors sharing the same

label are given as input. One pattern vector is selected from each set in order to maximize the classification margin with respect to the selected positive and negative pattern vectors. The problem of selecting the best pattern vectors is referred to as the *hard selection* problem. The hard selection problem is shown to be $\mathcal{NP}$-hard. We propose alternative linear and nonlinear approaches with tractable formulations, which we call *soft selection* problems. The selective nature of the these formulations is maintained by the restricted free slack concept. The intuition behind this concept is to reverse the combinatorial selection problem by detecting influential pattern vectors which require free slack to decrease their effect on the classification functions. Iteratively removing such pattern vectors, we can find those patter vectors with a larger margin. An *iterative elimination* method is proposed for this purpose. Another alternative approach is to provide enough free slack to identify all $t-1$ out of t pattern vectors to be removed at once, which leads to the *direct selection* method. The iterative elimination and the direct selection methods are found to produce similar results. Iterative elimination method is also compared with a naïve elimination method which uses standard SVM to eliminate pattern vectors. The results show that iterative elimination is superior to the naïve elimination method both in linear and nonlinear classification.

The motivation for the development of selective classification methods comes from the classification of cognitive states in a visuomotor pattern discrimination task. Due to the temporal noise in the data, the classification results obtained are poor with standard SVM methods. A sliding small time window of recordings are considered as sets of pattern vectors in selective classification. Well separated recordings are selected by the iterative elimination method. The selected recordings are evaluated with standard SVM methods, which result in a significant increase in the classification accuracy over the entire time line of the task. The increase is adjusted by a baseline method which isolates the actual improvement peaks. These peaks clearly mark the categorical discrimination stage of the visuomotor task, which involves a complex cognitive process that has not been detected by previous studies. This result suggest that the proposed selective classification methods are capable of providing promising solutions for other classification problems in neuroscience.

References

Bennet K, Campbell C (2000) Support vector machines: Hype or hallelujah? SIGKDD Explorations, 2(2):1–13

Brown M, Grundy W, Lin D, Cristianini N, Sugne C, Furey T, Ares M, Haussler D (2000) Knowledge-base analysis of microarray gene expressiondata by using support vector machines. PNAS, 97(1):262–267

Cristianini N, Shawe-Taylor J (2000) An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press, Cambridge

Dietterich TG, Lathrop RH, Lozano-Perez T (1997) Solving the multiple instance problem with axis-parallel rectangles. Artif Intell, 89:31–71

Eacott MJ, Gaffan D (1991) The role of monkey inferior parietal cortex in visual discrimination of identity and orientation of shapes. Behav Brain Res, 46(1):95–98

Garey MR, Johnson DS (1979) Computers and intractability: a guide to the theory of *NP*-completeness. W.H. Freeman, New York

Horel JA, Misantone LJ (1976) Visual discrimination impaired by cutting temporal lobe connections. Science, 193(4250):336–338

Huang Z, Chen H, Hsu CJ, Chenb WH, Wuc S (2004) Credit rating analysis with support vector machines and neural networks: a market comparative study. Decis Support Syst, 37:543–558

Joachims T (1998) Text categorization with support vector machines: Learning with many relevant features. In: Nédellec C, Rouveirol C (eds.), Proceedings of the European conference on machine learning. Springer, Berlin, pp. 137–142

Lal TN, Schroeder M, Hinterberger T, Weston J, Bogdan M, Birbaumer N, Schölkopf B (2004) Support vector channel selection in BCI. IEEE Trans Biomed Eng, 51(6):1003–1010

Ledberg A, Bressler SL, Ding M, Coppola R, Nakamura R (2007) Large-scale visuomotor integration in the cerebral cortex. Cereb Cortex, 17:44–62

Lee S, Verri A (2002) Pattern recognition with support vector machines. In: SVM 2002. Springer, Niagara Falls

Mendola JD, Corkin S (1999) Visual discrimination and attention after bilateral temporal-lobe lesions: a case study. Neuropsychologia, 37(1):91–102

Noble WS (2004) Kernel methods in computational biology. In: Support vector machine applications in computational biology. MIT Press, Cambridge, pp. 71–92

Shawe-Taylor J, Cristianini N (2004) Kernel methods for pattern analysis. Cambridge University Press, Cambridge

Trafalis TB, Ince H (2002) Support vector machine for regression and applications to financial forecasting. In: International joint conference on neural networks (IJCNN'02), Como, Italy. IEEE-INNS-ENNS

Vapnik V (1995) The nature of statistical learning theory. Springer, New York