Multiclass classification machines with the complexity of a single binary classifier

Paul Honeine[a,*], Zineb Noumir[a], Cédric Richard[b]

[a] *Institut Charles Delaunay (UMR CNRS 6279), LM2S, Université de technologie de Troyes, France*
[b] *Laboratoire Lagrange (UMR CNRS 7293), Observatoire de la Côte d'Azur, Université de Nice Sophia-Antipolis, France*

Abstract

In this paper, we study the multiclass classification problem. We derive a framework to solve this problem by providing algorithms with the complexity of a single binary classifier. The resulting multiclass machines can be decomposed into two categories. The first category corresponds to vector-output machines, where we develop several algorithms. In the second category, we show that the least-squares classifier can be easily cast into a multiclass one-versus-all scheme, without the need to train multiple binary classifiers. The proposed framework shows that, while keeping the classification accuracy essentially unchanged, the computational complexity is orders of magnitude lower than those previously reported in the literature. This makes our approach extremely powerful and conceptually simple. Moreover, we study the coding of the multiclass labels, and demonstrate that several celebrated approaches are equivalent. These arguments are illustrated with experimentations on well-known benchmarks.

Keywords: Multiclass classification, machine learning, SVM, one-versus-all, least-squares classification

1. Introduction

The multiclass classification problem has been widely investigated in machine learning and data mining, in many fields including signal and image processing. Based on a set of observations from several classes, the aim is to learn a decision rule that accurately classifies new observations. Most classification techniques were initially derived for binary task, i.e., solving two-class classification problems. This is the case of many kernel-based classifiers, including the large-margin technique in support vector machines (SVM) [1] and the least-squares machines (LSM)[1] [2, 3]. These classification techniques have been thoroughly investigated, and their performance well studied and understood. One seeks to generalize these approaches for multiclass tasks [4, 5], which find numerous applications in signal processing [6, 7, 8, 9].

[☆] This work was partly supported by ANR-08-SECU-013-02 VigiRes'Eau.

[*] Corresponding author

Email addresses: paul.honeine@utt.fr (Paul Honeine), zineb.noumir@utt.fr (Zineb Noumir), cedric.richard@unice.fr (Cédric Richard)

[1] The LSM (and several minor variants) has been rederived under several names, often with different methodological frameworks, i.e., functional analysis with estimation in a reproducing kernel Hilbert space versus Lagrange duality. This includes the regularized least-squares classification (RLS), least-squares SVM (LSSVM) and proximal SVM. The name LSM highlights the shared property of these machines, while distinguishing them from (sparse) SVM.

Preprint submitted to Signal Processing. **Signal Processing 93 (2013) 1013–1026** *January 6, 2013*

Multiclass machines can be roughly divided into two categories. The first category attempts to incorporate simultaneously all the classification constraints within a single-machine scheme, in a similar way to binary classifiers (for instance, by maximizing the margin between classes). However, such a scheme significantly increases the problem complexity and thus requires advanced optimization techniques [10, 11, 12]. The second category consists of a divide-to-conquer strategy, such as the one-versus-all[2] (OvA) [13], the one-versus-one (OvO) [14], and the decision directed-acyclic-graph (DAG) [15] schemes. In this case, the multiclass problem is decomposed into multiple binary tasks. All these subproblems are solved separately, and their results combined to infer the multiclass solution. For a survey, see [11, 16, 17, 18].

In [13], Rifkin and Klautau conducted a comparative study of multiclass machines, including those presented in all the aforementioned papers. Since these results are well-known in the literature, we give here a small review: multiclass schemes such as OvO, OvA, and DAG, have essentially the same accuracy as single-machine schemes. The LSM performs just as well as SVM, as illustrated in many studies [3, 19] including [20] with binary and multiclass classification. Therefore, a simple scheme such as OvA (for SVM or LSM) is preferable to a more complicated single-machine or error-correcting coding scheme. It is often argued that the main drawback of an OvA scheme is its computational complexity. For an ℓ class classification task, it requires to train ℓ binary classifiers, each one involving all the training data. This should be compared with the OvO and DAG schemes, which both need to train $\ell(\ell-1)/2$ binary classifiers, each individual subproblem being much smaller. See [14] for more details.

In this paper, we study the multiclass classification problem by providing algorithms with the complexity of a single binary classifier. To this end, we derive a framework for multiclass problems by considering two possible forms of the solution, which are equivalent in the binary case [21, 22]. The first form includes explicitly the class labels in the coefficients of the decision function to be estimated. The second form includes them implicitly. This leads to two classes of machines.

On the one hand, when labels are explicitly included in the coefficients of the expansion, we explore a new trend of deriving low computational algorithms based on learning vector-valued functions. Initially introduced for multi-task learning [23, 24], this idea has been naturally applied for multiclass classification problems, by using a SVM-like formulation [25, 26]. More recently, we derived a least-squares formulation in [27, 28]. In this paper, we provide a framework for deriving several algorithms based on binary classification algorithms, including LSSVM, RLS, and SVM.

On the other hand, we consider a model where the labels do not appear explicitly. We show that the nature of the LSM with the OvA scheme provides an algorithm, with essentially the same computational complexity as a single binary classifier. The proposed method is designated by oneLSM in this paper. We conduct a study on the model coefficients and we establish connections between the labels in multiclass problems. A single-machine formulation is proposed, and connections with other related work are presented [29].

We also conduct a comprehensive study on the class coding in multiclass classification. Several techniques have been proposed in the literature, as extensions of the binary problem. In this paper, we show that they are equivalent, including the ±1 coding [16, 18], the standard basis coding (or indicators) inspired from

[2] Short of one class versus all the rest, to be more specific.

artificial neural networks (ANN) [30], the alignment-based coding [31, 27], the consistency-based coding used with the inductive principle in multiclass SVM [32], and the minimum-correlation coding proposed in [26]. We study these label codings and investigate their equivalence. Experimental results corroborate these findings. A nonlinear extension is also proposed by applying a kernel on the labels.

The rest of the paper is organized as follows. Sec. 2 provides a succinct presentation of the classification problem in kernel-based machine learning. We describe the proposed framework in Sec. 3, and provide analogy with the binary classification case. This leads to two classes of machines, the vector-output machines defined in Sec. 4, and the oneLSM given in Sec. 5 where we revisit the least-squares machines. In Sec. 6, we study the label coding for multiclass classification. Sec. 7 is devoted for further discussions. Sec. 8 gives comparative results in terms of accuracy and computational time, as well as a statistical test. Conclusions and further directions are given in Sec. 9.

Notations: In this paper, we use the superscript index to designate an entry related to a subproblem, such as $f^{(k)}(\cdot)$ and $\alpha^{(k)}$ associated to the k-th subproblem classification. The subscript index is related to a given sample, such as y_i and α_i corresponding to the i-th sample $\boldsymbol{x}_i$.

2. From binary to multiclass classification: a primer

Binary classification with kernel machines

In a two-class classification problem, one seeks a decision rule that predicts well the class membership of a given sample, based on a set of training data with available class membership. Such decision rule, for any new sample $\boldsymbol{x}$, takes the form

$$\begin{cases} \text{if } f(\boldsymbol{x}) < 0, & \text{then } y = -1 \\ \text{if } f(\boldsymbol{x}) > 0, & \text{then } y = +1 \end{cases} \tag{1}$$

namely, comparing the sign of $f(\boldsymbol{x})$ to determine if $\boldsymbol{x}$ belongs to the (-1)-class or to the $(+1)$-class[3]. This function $f(\cdot)$ is estimated using a training set of labeled data, $\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_n, y_n)\}$, with label $y_i = \pm 1$ depending on which class sample $\boldsymbol{x}_i$ belongs to. This decision rule is equivalent to

$$\arg\max_{y=\pm 1} y f(\boldsymbol{x}), \tag{2}$$

as well as $\arg\min_{y=\pm1} |f(\boldsymbol{x}) - y|^2$.

A straightforward solution for the binary classification problem is obtained when one seeks a function from the space $\mathcal{H}$ of linear functions,

$$f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x}. \tag{3}$$

Consider the least-squares classification problem (also known as Ridge Regression) defined by

$$\min_{f \in \mathcal{H}} \sum_{j=1}^{n} |\boldsymbol{w}^\top \boldsymbol{x}_j - y_j|^2 + \gamma \|\boldsymbol{w}\|^2, \tag{4}$$

[3] One may also consider the Fisher coding, with $y = -(n/n_-)$ or $y = n/n_+$, where n_- and n_+ are the number of samples in each class.

where γ is a tunable parameter that controls the trade-off between errors on the training data and regularity of the solution. Taking the derivative of the above function with respect to $\boldsymbol{w}$, namely $2\sum_{j=1}^{n}(\boldsymbol{w}^\top\boldsymbol{x}_j - y_j)\,\boldsymbol{x}_j + 2\gamma\,\boldsymbol{w}$, and setting it to zero at the optimum, we observe that $\boldsymbol{w}$ can be written in the form[4]

$$\boldsymbol{w}^\top = \sum_{i=1}^{n} \alpha_i\,\boldsymbol{x}_i^\top, \tag{5}$$

with $\alpha_i = (y_i - \boldsymbol{w}^\top\boldsymbol{x}_i)/\gamma$. Substituting the above form of $\boldsymbol{w}$ in the previous one, we obtain the linear system of n equations with n unknowns

$$\sum_{j=1}^{n} \alpha_j\,\boldsymbol{x}_j^\top\boldsymbol{x}_i + \gamma\,\alpha_i = y_i, \qquad \text{for } i = 1, 2, \ldots, n.$$

In matrix form, we have $\boldsymbol{w} = \boldsymbol{X}\,\boldsymbol{\alpha}$ with $\boldsymbol{X} = [\boldsymbol{x}_1\ \boldsymbol{x}_2\ \cdots\ \boldsymbol{x}_n]$ and $\boldsymbol{\alpha}$ a column vector with entries α_j for $j = 1, 2, \ldots, n$. Therefore, the above linear system can be written as

$$(\boldsymbol{K} + \gamma\boldsymbol{I})\,\boldsymbol{\alpha} = \boldsymbol{y},$$

where $\boldsymbol{K}$ is a n-by-n matrix with entries $\boldsymbol{x}_i^\top\boldsymbol{x}_j$, for $i, j = 1, 2, \ldots, n$, i.e., $\boldsymbol{K} = \boldsymbol{X}^\top\boldsymbol{X}$, $\boldsymbol{y}$ is a column vector with entries y_j, for $j = 1, 2, \ldots, n$, and $\boldsymbol{I}$ is the identity matrix of appropriate size (n-by-n here).

The resulting coefficients are considered in the decision rule for any $\boldsymbol{x}$, with $f(\boldsymbol{x}) = \boldsymbol{\alpha}^\top\boldsymbol{\kappa}_{\boldsymbol{x}}$, where $\boldsymbol{\kappa}_{\boldsymbol{x}}$ is a column vector whose j-th entry is $\boldsymbol{x}_j^\top\boldsymbol{x}$. The decision rule is given by comparing the evaluation of this function $f(\boldsymbol{x})$ to a threshold according to (1). This threshold (called bias) is often set to 0, as advised in many studies [33].

There exists another formulation of the Ridge Regression problem, when the labels $y_i = \pm 1$ are applied. In this case, the least-squares problem (4) is equivalent to

$$\min_{f\in\mathcal{H}} \sum_{j=1}^{n} |y_j f(\boldsymbol{x}_j) - 1|^2 + \gamma\|f\|^2. \tag{6}$$

By considering the space of linear functions, of the form $f(\boldsymbol{x}) = \boldsymbol{w}^\top\boldsymbol{x}$, we obtain the following solution

$$\boldsymbol{w}^\top = \sum_{i=1}^{n} \beta_i\,y_i\,\boldsymbol{x}_i^\top. \tag{7}$$

The connection between (7) and (5) is obvious, with $\alpha_i = \beta_i\,y_i$. Both criteria, $f(\boldsymbol{x}_j) - y_j$ and $y_j f(\boldsymbol{x}_j)$, are considered in [19] for regression, as opposed to [2] where only the latter is used. It is worth noting that the equivalence between both criteria is true only when the (±1)-label coding is considered in classification.

Finally, thanks to the kernel trick, one can easily extend this linear model into a nonlinear one, by substituting some kernel function $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$ for $\boldsymbol{x}_i^\top\boldsymbol{x}_j$. In fact, inner products are involved, both in the estimation of the optimal coefficients $\boldsymbol{\alpha}$ (with $\boldsymbol{K}$) and in the decision function $f(\boldsymbol{x}) = \boldsymbol{\alpha}^\top\boldsymbol{\kappa}_{\boldsymbol{x}}$ (with $\boldsymbol{\kappa}_{\boldsymbol{x}}$). The most used kernels are the polynomial kernel, of the form $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i^\top\boldsymbol{x}_j + c)^p$, and the Gaussian kernel, with $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/2\sigma^2)$, where σ is the bandwidth parameter.

[4] This is the Representer Theorem [21], well known in regularized optimization problems, which states that the solution is a linear combination of the input training data. It also applies to the kernel-based formulation [22], where $\mathcal{H}$ is a reproducing kernel Hilbert space associated to some kernel κ.

Multiclass classification

Consider an ℓ-class classification task. Using a divide-to-conquer strategy, the problem can be solved by working on a collection of binary classification subproblems, then combining each of the resulting real-valued decision functions. Thus, each function $f^{(k)}(\cdot)$ is defined from a binary classification subproblem by using the same training data, $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$, and assigning its proper labels, $y_1^{(k)}, y_2^{(k)}, \ldots, y_n^{(k)}$.

In a one-versus-all (OvA) scheme, ℓ binary classifiers are trained, where each subproblem confronts a class with all the rest. In this case, we have $y_j^{(k)} \in \{-1; +1\}$ depending on the k-th class membership of $\boldsymbol{x}_j$. This requires estimating $\ell \times n$ coefficients, $\alpha_j^{(k)}$, for $j = 1, 2, \ldots, n$ and $k = 1, 2, \ldots, \ell$. In a one-versus-one (OvO) scheme, $\ell(\ell-1)/2$ binary classifiers are trained by taking data from each pair of classes. For this purpose, $y_j^{(k)} \in \{-1; 0; 1\}$ where the value zero corresponds to samples that do not belong to the classes under examination. Although $\ell(\ell-1)/2$ binary classifiers are trained, the individual problems are much smaller. Note that each sample is used by $n-1$ decision functions. It is well-known that these schemes (as well as directed-acyclic-graph, complete codes, ...) perform essentially identically (see Sec. 1 for references). Therefore, the simplest scheme is preferable to more complex ones, making OvA and OvO good candidates. Still, one needs to solve several binary classification subproblems.

Any multiclass machine, defined by a set of ℓ functions $f^{(k)}(\cdot)$, is completely described by the coefficients to be determined, either $\beta_j^{(k)}$ or $\alpha_j^{(k)}$ for $j = 1, 2, \ldots, n$ and $k = 1, 2, \ldots, \ell$. This yields an estimation problem with $n \times \ell$ unknown parameters. In this paper, we show that, without sacrificing performance, one can naturally reduce significantly the number of unknowns, by imposing some relation between the functions $f^{(k)}(\cdot)$'s. This yields multiclass classification machines with essentially the same computational complexity as a single binary classifier.

3. The proposed framework

Let $f^{(1)}(\cdot), f^{(2)}(\cdot), \ldots, f^{(\ell)}(\cdot)$ be the functions to be learnt[5], in the same spirit as the OvA scheme while not limiting ourselves to it. In what follows, we regroup the ℓ functions in the vector-of-functions[6]

$$\boldsymbol{f}(\cdot) = [f^{(1)}(\cdot) \;\; f^{(2)}(\cdot) \;\; \cdots \;\; f^{(\ell)}(\cdot)]^\top$$

For the k-th classification subproblem, defined by the function $f^{(k)}(\cdot)$, we assign the n-entry label vector $\boldsymbol{y}^{(k)}$, namely the (±1)-label coding as defined in Table 1 (See Sec. 6 for a study of several codings of the labels). Let $\boldsymbol{Y}$ be the ℓ-by-n matrix with each row corresponding to the labels associated to a classification subproblem, namely $\boldsymbol{Y}^\top = [\boldsymbol{y}^{(1)} \;\; \boldsymbol{y}^{(2)} \;\; \cdots \;\; \boldsymbol{y}^{(\ell)}]$. In other words, each column of $\boldsymbol{Y}$ contains $\boldsymbol{y}_i$, the label associated to $\boldsymbol{x}_i$ in all the classification subproblems. Thus, the i-th entry of $\boldsymbol{y}^{(k)}$ is the k-th entry of $\boldsymbol{y}_i$.

We explore the differences, between (4) and (6), and between the forms (5) and (7), now written in a multiclass classification formulation. By coupling these diversities, we investigate several classification methods. The following table summarized these differences, in terms of the optimized criterion, the parameters to be estimated, β_i's or α_i's, as well as the sparsity of the solution:

[5] In principle, ℓ functions can encode up to 2^ℓ different classes. However, the design of an optimal coding matrix for such task for the maximum number of classes requires prior information on the samples, often unavailable in practice. See for instance [34] for an information-theoretic point of view. This study is out of scope of this paper, where we consider ℓ functions to encode ℓ classes.

[6] To be more precise, $\boldsymbol{f}(\cdot)$ is an ℓ-tuple of the discriminant functions.

Table 1: Expressions of well-known labelbooks.

Labelbook	$[\boldsymbol{y}^{(k)}]_i = [\boldsymbol{y}_i]_k$	$\boldsymbol{y}_i^\top \boldsymbol{y}_j$
(± 1)-label	$\begin{cases} +1 & \text{if } \boldsymbol{x}_i \text{ belongs to class } k; \\ -1 & \text{otherwise} \end{cases}$	$\begin{cases} \ell & \text{if } \boldsymbol{y}_i = \boldsymbol{y}_j; \\ \ell - 4 & \text{otherwise} \end{cases}$
Standard basis (or indicators)	$\begin{cases} 1 & \text{if } \boldsymbol{x}_i \text{ belongs to class } k; \\ 0 & \text{otherwise} \end{cases}$	$\begin{cases} 1 & \text{if } \boldsymbol{y}_i = \boldsymbol{y}_j; \\ 0 & \text{otherwise} \end{cases}$
Alignment-based	$\begin{cases} \sqrt{\frac{\ell-1}{\ell}} & \text{if } \boldsymbol{x}_i \text{ belongs to class } k; \\ \frac{-1}{\sqrt{\ell(\ell-1)}} & \text{otherwise} \end{cases}$	$\begin{cases} 1 & \text{if } \boldsymbol{y}_i = \boldsymbol{y}_j; \\ \frac{-1}{\ell-1} & \text{otherwise} \end{cases}$
Consistency-based	$\begin{cases} 1 & \text{if } \boldsymbol{x}_i \text{ belongs to class } k; \\ \frac{-1}{\ell-1} & \text{otherwise} \end{cases}$	$\begin{cases} \frac{\ell}{\ell-1} & \text{if } \boldsymbol{y}_i = \boldsymbol{y}_j; \\ \frac{-\ell}{(\ell-1)^2} & \text{otherwise} \end{cases}$

Table 2: Analogy between the binary and the multiclass formulations.

	binary formulation	multiclass formulation
models	$f(\cdot)$	$\boldsymbol{f}(\boldsymbol{x}) = [f^{(1)}(\cdot) \ \cdots \ f^{(\ell)}(\cdot)]^\top$
	$\boldsymbol{w}$	$\boldsymbol{W} = [\boldsymbol{w}^{(1)} \ \cdots \ \boldsymbol{w}^{(\ell)}]$
	$f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x}$	$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{W}^\top \boldsymbol{x}$
	$\boldsymbol{w} = \boldsymbol{X}\boldsymbol{\alpha}$	$\boldsymbol{W} = \boldsymbol{X}[\boldsymbol{\alpha}^{(1)} \ \cdots \ \boldsymbol{\alpha}^{(\ell)}]$
	$\boldsymbol{w}^\top = \sum_i \beta_i \, y_i \, \boldsymbol{x}_i^\top$	$\boldsymbol{W}^\top = \sum_i \beta_i \, \boldsymbol{y}_i \, \boldsymbol{x}_i^\top$
criterion	$y_j f(\boldsymbol{x}_j)$	$\boldsymbol{y}_j \boldsymbol{f}(\boldsymbol{x}_j)$
	$\lvert f(\boldsymbol{x}_j) - y_j \rvert^2$	$\lVert \boldsymbol{f}(\boldsymbol{x}_j) - \boldsymbol{y}_j \rVert^2$
	$\lVert \boldsymbol{w} \rVert^2$	$\lVert \boldsymbol{W} \rVert_F^2$
decision	$\max\limits_{y=\pm 1} y f(\boldsymbol{x})$	$\max\limits_{\boldsymbol{y} \in \mathcal{Y}} \boldsymbol{y}^\top \boldsymbol{f}(\boldsymbol{x})$
	$\min\limits_{y=\pm 1} \lvert f(\boldsymbol{x}) - y \rvert^2$	$\min\limits_{\boldsymbol{y} \in \mathcal{Y}} \lVert \boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{y} \rVert^2$

	vector-output machines (see Section 4)			OvA (see Section 5)
	RLS	LSSVM	SVM	LSM
criterion	$\boldsymbol{f}(\boldsymbol{x}_j) - \boldsymbol{y}_j$	$\boldsymbol{y}_j \boldsymbol{f}(\boldsymbol{x}_j)$	$\boldsymbol{f}(\boldsymbol{x}_j) - \boldsymbol{y}_j$	$\boldsymbol{f}(\boldsymbol{x}_j) - \boldsymbol{y}_j$
β_i's or α_i's	β_i	β_i	β_i	α_i
full or sparse	full	full	sparse	full

Finally, once the model parameters estimated, the decision rule is given by analogy with (2), for a given $\boldsymbol{x}$:

$$\arg\max_{\boldsymbol{y}} \boldsymbol{y}^\top \boldsymbol{f}(\boldsymbol{x}) \tag{8}$$

The decision rule is studied in detail in Sec. 7.1.

The analogy between the binary formulation and the multiclass formulation is given in Table 2, where the vector $\boldsymbol{w}$ is used in the former and the matrix $\boldsymbol{W}$ in the latter. Thus, the Euclidean vector norm $\|\boldsymbol{w}\|$ is replaced by the matrix norm $\|\boldsymbol{W}\|_F$, where $\|\cdot\|_F$ denotes Frobenius norm defined as

$$\|\boldsymbol{W}\|_F^2 = \sum_{i,j} \big([\boldsymbol{W}]_{i,j}\big)^2 = \text{trace}(\boldsymbol{W}^\top \boldsymbol{W}),$$

We associate these methods with several label codes, beyond the conventional (±1)-label. See Table 1. The alignment-based label coding can be viewed as a natural extension of the binary ±1 label coding. This is illustrated in Fig. 1. We make a thorough study of the label coding in Sec. 6. Independent of the choice of the label coding, we derive next algorithms for multiclass classification.

4. Vector-output machines for multiclass classification

In this section, we show that one can easily derive multiclass classification algorithms, in the same spirit as binary classification algorithms and specifically RLS, SVM and LSSVM.

Following [26] and more recently [27, 28], we propose the following expression for $\boldsymbol{f}(\boldsymbol{x})$

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{W}^\top \boldsymbol{x}, \tag{9}$$

with $\boldsymbol{W}$ a d-by-ℓ matrix defined, by analogy with (7), as

$$\boldsymbol{W}^\top = \sum_{j=1}^{n} \beta_j \, \boldsymbol{y}_j \, \boldsymbol{x}_j^\top.$$

In this case, we have

$$\boldsymbol{f}(\boldsymbol{x}) = \sum_{j=1}^{n} \beta_j \, \boldsymbol{y}_j \, \boldsymbol{x}_j^\top \boldsymbol{x}, \tag{10}$$

which means that the vector $\boldsymbol{f}(\boldsymbol{x})$ belongs to the span of the training label vectors $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n$.

An interesting property of this model is that the functions $f(\cdot)$'s in $\boldsymbol{f}(\cdot)$ share the same coefficients, namely

$$\beta_j = \beta_j^{(k)} \qquad \text{for all } k = 1, 2, \ldots, \ell$$

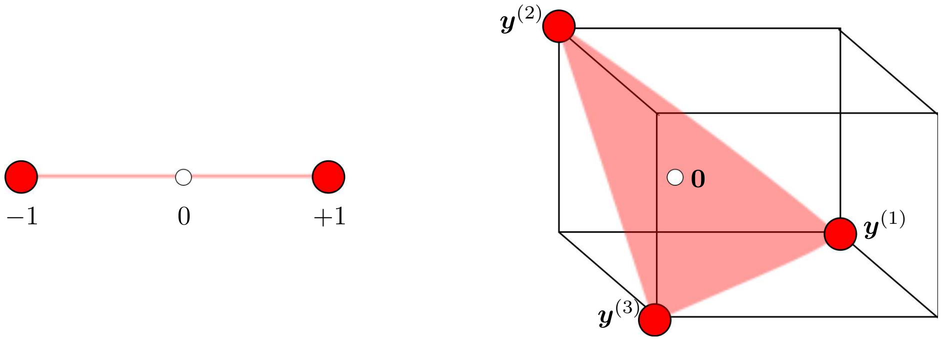

Figure 1: Illustration of the class-membership encoding. Left: In the binary case, $y = \pm 1$ are unit-norm and defined by a segment centered at the origin. Right: In the multiclass case, the vector labels are unit-norm vertices and having the origin as centroid, as defined in Table 1 (here $\ell = 3$) with $\boldsymbol{y}^{(1)} = \left[\frac{\sqrt{2}}{\sqrt{3}} \;\; \frac{-1}{\sqrt{6}} \;\; \frac{-1}{\sqrt{6}}\right]$, $\boldsymbol{y}^{(2)} = \left[\frac{-1}{\sqrt{6}} \;\; \frac{\sqrt{2}}{\sqrt{3}} \;\; \frac{-1}{\sqrt{6}}\right]$, $\boldsymbol{y}^{(3)} = \left[\frac{-1}{\sqrt{6}} \;\; \frac{-1}{\sqrt{6}} \;\; \frac{\sqrt{2}}{\sqrt{3}}\right]$.

In (10), these coefficients are weighted by the label associated to $\boldsymbol{x}_j$ in the k-th class membership, i.e, $[\boldsymbol{y}^{(k)}]_j = \pm 1$, which leads to the only difference between the functions in $\boldsymbol{f}(\cdot)$. This allows us to have only n unknowns, in the same way as a single binary classifier. Next, we show how to estimate these coefficients, by adapting several classical classification algorithms to operate for vector output:

- multiclass vo-RLS
- multiclass vo-LSSVM
- multiclass vo-SVM which provides a sparse solution to the latter

It is worth noting that both RLS and LSSVM are equivalent in the scalar-output case, when $y_j = \pm 1$.

Finally, once the model parameters β_i's estimated, the decision rule is defined by combining (8) with (10), for any given $\boldsymbol{x}$, with

$$\arg\max_{\boldsymbol{y}} \sum_{j=1}^{n} \beta_j \, \boldsymbol{y}^\top \boldsymbol{y}_j \, \boldsymbol{x}_j^\top \boldsymbol{x}. \tag{11}$$

4.1. Multiclass vo-RLS

In the least-squares sense, we consider the following optimization problem

$$\min_{\boldsymbol{f}} \sum_{j=1}^{n} \|\boldsymbol{f}(\boldsymbol{x}_j) - \boldsymbol{y}_j\|^2 + \gamma \, \|\boldsymbol{\beta}\|^2, \tag{12}$$

where $\boldsymbol{\beta} = [\beta_1 \;\; \beta_2 \;\; \cdots \;\; \beta_n]^\top$. By substituting the expression of $\boldsymbol{f}(\boldsymbol{x})$ in this optimization problem, we get the following cost function to be minimized with respect to the β_i's:

$$\sum_{i=1}^{n} \boldsymbol{y}_i^\top \boldsymbol{y}_i + \sum_{i=1}^{n} \sum_{j,k=1}^{n} \beta_j \beta_k \boldsymbol{x}_i^\top \boldsymbol{x}_j \, \boldsymbol{x}_i^\top \boldsymbol{x}_k \, \boldsymbol{y}_j^\top \boldsymbol{y}_k + \gamma \sum_{i=1}^{n} \beta_i^2 - 2 \sum_{i=1}^{n} \boldsymbol{y}_i^\top \sum_{j=1}^{n} \beta_j \boldsymbol{x}_i^\top \boldsymbol{x}_j \, \boldsymbol{y}_j.$$

This optimization problem can be written in matrix form as,

$$\min_{\boldsymbol{\beta}} \boldsymbol{Y}^\top \boldsymbol{Y} - 2 \, \boldsymbol{d} \boldsymbol{\beta} + \boldsymbol{\beta}^\top \boldsymbol{G} \boldsymbol{\beta} + \gamma \, \boldsymbol{\beta}^\top \boldsymbol{\beta}, \tag{13}$$

where $\boldsymbol{G}$ is a matrix whose (j,k)-th entry is

$$[\boldsymbol{G}]_{j,k} = \boldsymbol{y}_j^\top \boldsymbol{y}_k \big(\boldsymbol{x}_j^\top \boldsymbol{x}_k\big)^2,$$

and $\boldsymbol{d}$ is a vector whose j-th entry is

$$[\boldsymbol{d}]_j = \sum_{i=1}^{n} \boldsymbol{y}_i^\top \boldsymbol{y}_j \, \boldsymbol{x}_i^\top \boldsymbol{x}_j.$$

By taking the gradient of the objective function in (13) with respect to $\boldsymbol{\beta}$, namely $-\boldsymbol{d} + \boldsymbol{G}\boldsymbol{\beta} + \gamma\,\boldsymbol{\beta}$, and setting it to zero, we obtain the final solution

$$(\boldsymbol{G} + \gamma\,\boldsymbol{I})\,\boldsymbol{\beta} = \boldsymbol{d}. \tag{14}$$

One can also consider other regularization terms in the above optimization problem. For instance, we consider the following optimization problem

$$\min_{\boldsymbol{f}} \sum_{j=1}^{n} \|\boldsymbol{f}(\boldsymbol{x}_j) - \boldsymbol{y}_j\|^2 + \gamma\, R(\boldsymbol{f}), \tag{15}$$

where the regularization term is given by

$$R(\boldsymbol{f}) = \sum_{i,j=1}^{n} \beta_i \beta_j \, \boldsymbol{y}_i^\top \boldsymbol{y}_j \, \boldsymbol{x}_i^\top \boldsymbol{x}_j. \tag{16}$$

From above, it is easy to see that we obtain the final solution

$$(\boldsymbol{G} + \gamma\,\boldsymbol{H})\,\boldsymbol{\beta} = \boldsymbol{d}, \tag{17}$$

where $\boldsymbol{H}$ is a matrix whose (i,j)-th entry is

$$[\boldsymbol{H}]_{i,j} = \boldsymbol{y}_i^\top \boldsymbol{y}_j \, \boldsymbol{x}_i^\top \boldsymbol{x}_j.$$

In this paper, we study the performance of both optimization problems. The vo-RLS problem defined by (12)-(14) is denoted by vo-RLS($\boldsymbol{\beta}$), while the use of the regularization term (16), establishing the problem (12)-(14), is denoted by vo-RLS($\boldsymbol{f}$).

We conclude that a multiclass regularized least squares classifier can be obtained by solving a single system of n linear equations with n unknowns. This requires the inversion of a n-by-n matrix. The computational complexity of this algorithm is cubic in the number of training data, n, but essentially independent of the number of classes ℓ. This is made possible here thanks to the fact that the label vectors only appear in terms of inner products, with $\boldsymbol{y}_i^\top \boldsymbol{y}_j$ in $\boldsymbol{H}$, $\boldsymbol{G}$ and $\boldsymbol{d}$. This reflects an analogy with the kernel trick in kernel machines, where data are involved only in terms of inner products, namely $\boldsymbol{x}_i^\top \boldsymbol{x}_j$. See Sec. 7.2 for a discussion on applying nonlinear kernels on the labels.

4.2. Multiclass vo-LSSVM

Another way to tackle the problem, is to solve an optimization problem with equality constraints, such as

$$\min_{\boldsymbol{W},\boldsymbol{\xi}} \frac{1}{2}\|\boldsymbol{W}\|_F^2 + \gamma \frac{1}{2} \sum_{i=1}^{n} \xi_i^2,$$

subject to

$$\boldsymbol{y}_i^\top(\boldsymbol{W}^\top\boldsymbol{x}_i) = 1-\xi_i, \qquad \text{for } i=1,2,\ldots,n.$$

The corresponding Lagrangian is defined by

$$\mathcal{L}(\boldsymbol{W},\boldsymbol{\xi},\boldsymbol{\beta})=\frac{1}{2}\|\boldsymbol{W}\|_F^2+\gamma\frac{1}{2}\sum_{i=1}^n\xi_i^2-\sum_{i=1}^n\beta_i(\boldsymbol{y}_i^\top(\boldsymbol{W}^\top\boldsymbol{x}_i)-1+\xi_i),$$

where β_i's are the Lagrangian multipliers. The optimality conditions are given by

$$\begin{cases} \frac{\partial\mathcal{L}}{\partial\boldsymbol{W}}=0 & \Rightarrow \quad \boldsymbol{W}^\top=\sum_i\beta_i\,\boldsymbol{y}_i\,\boldsymbol{x}_i^\top \\ \\ \frac{\partial\mathcal{L}}{\partial\xi_i}=0 & \Rightarrow \quad \beta_i=\gamma\,\xi_i \\ \\ \frac{\partial\mathcal{L}}{\partial\beta_i}=0 & \Rightarrow \quad \boldsymbol{y}_i^\top(\boldsymbol{W}^\top\boldsymbol{x}_i)-1+\xi_i=0 \end{cases}$$

This leads to the following linear system

$$\left(\boldsymbol{H}+\gamma^{-1}\boldsymbol{I}\right)\boldsymbol{\beta}=\mathbf{1}_n$$

where $\mathbf{1}_n$ is a n-entry column vector of ones. Solving this linear system, i.e., identifying $\boldsymbol{\beta}$, requires the inversion of a n-by-n matrix. The computational cost of such inversion is about $\mathcal{O}(n^3)$. This should be compared with Suykens and Vanderwalle's multiclass LSSVM algorithm in [35] which requires the inversion of an $(\ell n+\ell)$-by-$(\ell n+\ell)$ matrix, with a computational complexity that scales cubically with the number of classes, namely $\mathcal{O}((\ell n)^3)$.

4.3. Multiclass vo-SVM

In the same spirit as the support vector machines [36], we consider the following optimization problem:

$$\min_{\boldsymbol{W},\boldsymbol{\xi}}\frac{1}{2}\|\boldsymbol{W}\|_F^2+\gamma\sum_{i=1}^n\xi_i \tag{18}$$

subject to

$$\boldsymbol{y}_i^\top(\boldsymbol{W}^\top\boldsymbol{x}_i)\geq 1-\xi_i, \qquad \text{for } i=1,2,\ldots,n,$$

and

$$\xi_i\geq 0, \qquad \text{for } i=1,2,\ldots,n.$$

One can solve this constrained optimization problem using Lagrangian, $\mathcal{L}(\boldsymbol{W},\boldsymbol{\xi},\boldsymbol{\beta},\boldsymbol{\delta})$ with $\boldsymbol{\beta}$ and $\boldsymbol{\delta}$ the Lagrangian multipliers, namely

$$\min_{\boldsymbol{W},\boldsymbol{b},\boldsymbol{\xi}}\max_{\boldsymbol{\beta},\boldsymbol{\delta}}\frac{1}{2}\text{trace}(\boldsymbol{W}^\top\boldsymbol{W})+\gamma\sum_{i=1}^n\xi_i-\sum_{i=1}^n\delta_i\,\xi_i-\sum_{i=1}^n\beta_i\left(\boldsymbol{y}_i^\top(\boldsymbol{W}^\top\boldsymbol{x}_i)-1+\xi_i\right).$$

The minimum of the Lagrangian with respect to $\boldsymbol{W}$ and $\boldsymbol{\xi}$ is given by

$$\begin{cases} \frac{\partial\mathcal{L}}{\partial\boldsymbol{W}}=0 & \Rightarrow \quad \boldsymbol{W}^\top=\sum_i\beta_i\,\boldsymbol{y}_i\,\boldsymbol{x}_i^\top \\ \\ \frac{\partial\mathcal{L}}{\partial\xi_i}=0 & \Rightarrow \quad \beta_i+\delta_i=\gamma \end{cases}$$

This leads to the following dual form

$$\max_{\boldsymbol{\beta}} \sum_{i=1}^{n} \beta_i - \frac{1}{2} \sum_{i,j=1}^{n} \beta_i \beta_j \, \boldsymbol{y}_i^\top \boldsymbol{y}_j \, \boldsymbol{x}_i^\top \boldsymbol{x}_j \tag{19}$$

subject to

$$0 \leq \beta_1, \beta_2, \ldots, \beta_n \leq \gamma. \tag{20}$$

In matrix form, we can write

$$\max_{\boldsymbol{\beta}} \mathbf{1}_n^\top \boldsymbol{\beta} - \frac{1}{2} \boldsymbol{\beta}^\top \boldsymbol{H} \boldsymbol{\beta}$$

subject to

$$\mathbf{0}_n \leq \boldsymbol{\beta} \leq \gamma \mathbf{1}_n,$$

where $\mathbf{0}_n$ is a n-entry column vector of zeros. This is a quadratic programming problem, which can be solved using any off-the-shelf optimization technique. The is essentially similar to the SVM binary-classifier, thus the same optimization routines can be used for both binary and multiclass classification tasks[7].

4.4. Model with a bias

In this paper, we study the bias-free model (9). One may also consider an offset in the model, with

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{b},$$

where $\boldsymbol{b}$ is the bias parameter. It is easy to see that our approach extends naturally to this model, where the constraint $\boldsymbol{y}_i^\top (\boldsymbol{W}^\top \boldsymbol{x}_i) = 1 - \xi_i$ is substituted with

$$\boldsymbol{y}_i^\top (\boldsymbol{W}^\top \boldsymbol{x}_i + \boldsymbol{b}) = 1 - \xi_i,$$

in the vo-LSSVM and vo-SVM algorithms. In their dual forms, we get the following additional constraint:

$$\boldsymbol{Y} \boldsymbol{\beta} = \mathbf{0}_\ell. \tag{21}$$

This constraint can be easily incorporated within the proposed algorithms, as illustrated here.

Consider the vo-LSSVM, this leads to the following augmented linear system

$$\left[\begin{array}{c|c} \mathbf{0} & \boldsymbol{Y} \\ \hline \boldsymbol{Y}^\top & \boldsymbol{H} + \gamma^{-1} \boldsymbol{I} \end{array}\right] \begin{bmatrix} \boldsymbol{b} \\ \boldsymbol{\beta} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_\ell \\ \mathbf{1}_n \end{bmatrix}.$$

where $\mathbf{0}$ is the square matrix of zeros of appropriate size (ℓ-by-ℓ here). Solving this linear system requires the inversion of a $(n+\ell)$-by-$(n+\ell)$ matrix. Since the number of classes is significantly smaller than the number of available data, the computational cost of such inversion is about $\mathcal{O}(n^3)$. This still outperforms the multiclass LSSVM algorithm proposed by Suykens and Vanderwalle's in [35] with $\mathcal{O}((\ell n)^3)$.

In the case of the vo-SVM, we get

$$\max_{\boldsymbol{\beta}} \mathbf{1}_n^\top \boldsymbol{\beta} - \frac{1}{2} \boldsymbol{\beta}^\top \boldsymbol{H} \boldsymbol{\beta}$$

[7] For instance, one can use the same matlab function `quadprog` for both problems.

subject to

$$\boldsymbol{Y}\boldsymbol{\beta} = \mathbf{0}_\ell \quad \text{and} \quad \mathbf{0}_n \leq \boldsymbol{\beta} \leq \gamma \mathbf{1}_n.$$

This quadratic programming problem is similar to the SVM binary-classifier. The only difference is the linear constraint, here $\sum_{i=1}^{n} \beta_i \, \boldsymbol{y}_i = \mathbf{0}_\ell$ as opposed to $\sum_{i=1}^{n} \beta_i \, y_i = 0$ for the binary case. This is essentially the multiclass SVM algorithm proposed Szedmak *et al.* in [26] (see also [25]).

Discussion: biased versus unbiased model

The use of a biased or an unbiased model is still an open question in machine learning, in the binary classification case [33] as well as in the multiclass case [37]. For instance in SVM, and by analogy with the binary case, one can define the optimal value of the bias by averaging, over all support vectors $\boldsymbol{x}_i$, the expression

$$\boldsymbol{b}_i = \boldsymbol{y}_i - \sum_{j=1}^{n} \beta_j \, \boldsymbol{y}_j \, \boldsymbol{x}_j^\top \boldsymbol{x}_i.$$

In [38, page 203], it is advised not to use an optimal value, but to change it in order to adjust the number of false positives and false negatives. Many authors disallow the bias term completely, i.e., $\boldsymbol{b} = \mathbf{0}_\ell$, see for instance [39, 33]. It is worth noting that in SVM implementations that disallow the bias term, the linear constraint $\sum_{i=1}^{n} \beta_i \, \boldsymbol{y}_i = \mathbf{0}_\ell$ is removed from the SVM problem. For the least-squares approaches, many studies motivate the use of the unbiased version, see for instance [33, 40].

In practice, it turns out that the unbiased model often outperforms the biased one. This is illustrated in Sec. 8, *e.g.*, by comparing our vo-SVM with the one proposed by Szedmak *et al.* in [26]. A statistical test is also conducted on the performance of the latter, showing that it highly depends on the choice of the label coding. Still the classification accuracy remains poor in general, as shown in our experimentations.

5. The multiclass least-squares machines : oneLSM

In this section, we revisit the least-squares solution with

$$\boldsymbol{w}^\top = \sum_{i=1}^{n} \alpha_i \, \boldsymbol{x}_i^\top,$$

and show how our framework is a natural choice for the multiclass problem. To this end, we recall the least-squares problem for the binary case, as defined in Sec. 2. The solution is given as $(\boldsymbol{K} + \gamma \boldsymbol{I}) \, \boldsymbol{\alpha} = \boldsymbol{y}$, which leads to $f(\boldsymbol{x}) = \boldsymbol{y}^\top (\boldsymbol{K} + \gamma \boldsymbol{I})^{-1} \, \boldsymbol{\kappa}_{\boldsymbol{x}}$.

5.1. Multiclass LSM in one shot

Consider the OvA scheme for an ℓ-class classification task. This corresponds to estimate ℓ decision functions, each taking the form

$$f^{(k)}(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i^{(k)} \, \boldsymbol{x}_i^\top \boldsymbol{x}, \tag{22}$$

for $k = 1, 2, \ldots, \ell$. The ℓ vectors of unknowns are denoted by $\boldsymbol{\alpha}^{(k)}$, with entries $\alpha_i^{(k)}$ for $i = 1, 2, \ldots, n$. It is obvious that all these functions share the same input data, $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$. The only difference between the binary classifiers is the assigned labels in the subproblems. For the k-th binary classification subproblem, we assign the label vector $\boldsymbol{y}^{(k)}$ as given in Table 1. See Sec. 6 for a study of the label coding and its impact on the solution.

Then the multiclass problem is defined by the expressions

$$\begin{aligned}
\boldsymbol{\alpha}^{(1)} &= (\boldsymbol{K}+\gamma\boldsymbol{I})^{-1}\,\boldsymbol{y}^{(1)},\\
\boldsymbol{\alpha}^{(2)} &= (\boldsymbol{K}+\gamma\boldsymbol{I})^{-1}\,\boldsymbol{y}^{(2)},\\
&\vdots\\
\boldsymbol{\alpha}^{(\ell)} &= (\boldsymbol{K}+\gamma\boldsymbol{I})^{-1}\,\boldsymbol{y}^{(\ell)}.
\end{aligned}\tag{23}$$

It is obvious that one only needs to inverse a single matrix, $(\boldsymbol{K}+\gamma\boldsymbol{I})$, for all the binary classification subproblems. Moreover, the above ℓ equations (each implicitly involving n equalities) can be easily written in matrix form with

$$[\boldsymbol{\alpha}^{(1)}\ \boldsymbol{\alpha}^{(2)}\ \cdots\ \boldsymbol{\alpha}^{(\ell)}] = (\boldsymbol{K}+\gamma\boldsymbol{I})^{-1}\,\boldsymbol{Y}^\top,$$

and consequently we obtain

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{Y}(\boldsymbol{K}+\gamma\boldsymbol{I})^{-1}\,\boldsymbol{\kappa}_{\boldsymbol{x}}.\tag{24}$$

It is obvious that this approach, denoted oneLSM in this paper, is fundamentally an OvA scheme of LSM, without naively computing ℓ binary classifiers. Finally, the label of any given $\boldsymbol{x}$ is determined by

$$\arg\max_{\boldsymbol{y}} \boldsymbol{y}^\top\boldsymbol{Y}(\boldsymbol{K}+\gamma\boldsymbol{I})^{-1}\,\boldsymbol{\kappa}_{\boldsymbol{x}}.\tag{25}$$

By studying the computational requirements, the matrix inversion is the most demanding operation. The proposed trick allows us to compute a single matrix inversion of an n-by-n matrix, rather than ℓ inversions of the same matrix for the naive implementation.

5.2. Relationship between the parameters

While expression (22) shows that there are $n\times\ell$ unknowns, the $\alpha_i^{(k)}$'s, there exists a connection between these parameters. The following proposition can be easily derived from (23) :

Proposition 1. *In the oneLSM multiclass problem (as well as in LSM with the OvA scheme), the coefficients of binary classifiers are connected to each others with the following relation :*

$$\boldsymbol{\alpha}^{(k)^\top}\boldsymbol{y}^{(k')} = \boldsymbol{\alpha}^{(k')^\top}\boldsymbol{y}^{(k)},$$

for all pairs $k,k' = 1,2,\ldots,\ell$.

This result leads to other expressions, such as $[\boldsymbol{\alpha}^{(1)}\ \boldsymbol{\alpha}^{(2)}\ \cdots\ \boldsymbol{\alpha}^{(\ell)}]^\top\boldsymbol{y}^{(k)} = [\boldsymbol{y}^{(1)}\ \boldsymbol{y}^{(2)}\ \cdots\ \boldsymbol{y}^{(\ell)}]^\top\boldsymbol{\alpha}^{(k)}$. These results give very useful insights, and can be illustrated on specific types of labels, such as the standard basis (or indicators) given in Table 1, namely $[\boldsymbol{y}^{(k)}]_j = 1$ if $\boldsymbol{x}_j$ belongs to class k and 0 otherwise. In this case, we have

$$\sum_{\boldsymbol{x}_j\in\mathcal{C}^{(k')}} \alpha_j^{(k)} = \sum_{\boldsymbol{x}_i\in\mathcal{C}^{(k)}} \alpha_i^{(k')}, \qquad \text{for any } k,k' = 1,2,\ldots,\ell,$$

where $\mathcal{C}^{(k)}$ denotes the k-th class. For a given binary classifier k, we consider the $\ell - 1$ above equations obtained for $\mathcal{C}^{(k')}$ of the remaining classes, which sums to

$$\sum_{k' \neq k} \sum_{\boldsymbol{x}_j \in \mathcal{C}^{(k')}} \alpha_j^{(k)} = \sum_{\boldsymbol{x}_i \in \mathcal{C}^{(k)}} \sum_{k' \neq k} \alpha_i^{(k')}, \text{ for a given } k = 1, \ldots, \ell.$$

The above left-hand side can be written as $\sum_{\boldsymbol{x}_i \notin \mathcal{C}^{(k)}} \alpha_j^{(k)}$. Therefore, the above equation can be read as follows: For a given binary classifier k-against-all the rest, the contributions (in terms of α's) in classifier k of the data belonging to all the remaining classes is equal to the contribution in all the remaining classifiers of the data belonging to the k-th class.

To the best of our knowledge, these connections between the parameters and the labels were never stated or proved before. These results are to be compared with the equality constraints (21) in the SVM and LSSVM machines, as given in Sec. 4.4. In the well known binary classification case with (± 1)-label coding, we have $\sum_{i=1}^{n} \beta_i \, y_i = 0$, thus

$$\sum_{\boldsymbol{x}_j \in \mathcal{C}} \beta_j = \sum_{\boldsymbol{x}_i \notin \mathcal{C}} \beta_i.$$

5.3. Multiclass oneLSM as a single-machine problem

The proposed approach can be derived using a single-machine problem as illustrated in next proposition. To this end, we write (24) as

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{W}^\top \boldsymbol{x},$$

where $\boldsymbol{W} = [\boldsymbol{w}^{(1)} \; \boldsymbol{w}^{(2)} \; \cdots \; \boldsymbol{w}^{(\ell)}]$ with $\boldsymbol{w}^{(k)}$ associated to the k-th binary classifier.

Proposition 2. *The multiclass LSM problem is equivalent to the optimization problem*

$$\min_{\boldsymbol{W}} \sum_{i=1}^{n} \|\boldsymbol{W}^\top \boldsymbol{x}_i - \boldsymbol{y}_i\|^2 + \gamma \|\boldsymbol{W}\|_F^2,$$

whose solution $\boldsymbol{W} = \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X} + \gamma \boldsymbol{I})^{-1} \boldsymbol{Y}^\top$ *defines the decision function* $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{W}^\top \boldsymbol{x}$.

Proof. To see this, consider the derivative with respect to $\boldsymbol{W}^\top$ of the above cost function, namely $\sum_{i=1}^{n} \left(\boldsymbol{W}^\top \boldsymbol{x}_i^\top \boldsymbol{x}_i - \boldsymbol{y}_i \boldsymbol{x}_i^\top \right) + \gamma \boldsymbol{W}^\top$, and set it to zero:

$$\boldsymbol{W}^\top (\boldsymbol{X}\boldsymbol{X}^\top + \gamma \boldsymbol{I}) = \boldsymbol{Y}\boldsymbol{X}^\top.$$

This leads to $\boldsymbol{W}^\top = \boldsymbol{Y}(\boldsymbol{X}^\top \boldsymbol{X} + \gamma \boldsymbol{I})^{-1} \boldsymbol{X}^\top$, where the matrix identity $\boldsymbol{X}^\top (\boldsymbol{X}\boldsymbol{X}^\top + \gamma \boldsymbol{I})^{-1} = (\boldsymbol{X}^\top \boldsymbol{X} + \gamma \boldsymbol{I})^{-1} \boldsymbol{X}^\top$ is considered. □

6. Labels

For the sake of simplicity, we will use label to name a label vector, and labelbook $\mathcal{Y}$ to denote the set of ℓ label vectors, each one being associated to a class. Thus, for any $\boldsymbol{x}_i$ we associate $\boldsymbol{y}_i \in \mathcal{Y}$, with $\boldsymbol{y}_i \neq \boldsymbol{y}_j$ if and only if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ belong to different classes. The choice of a labelbook for a given classification task is studied in this section.

Several labelbooks can be used as extensions of the binary case to the multiclass case. Expressions of the most used ones are given in Table 1:

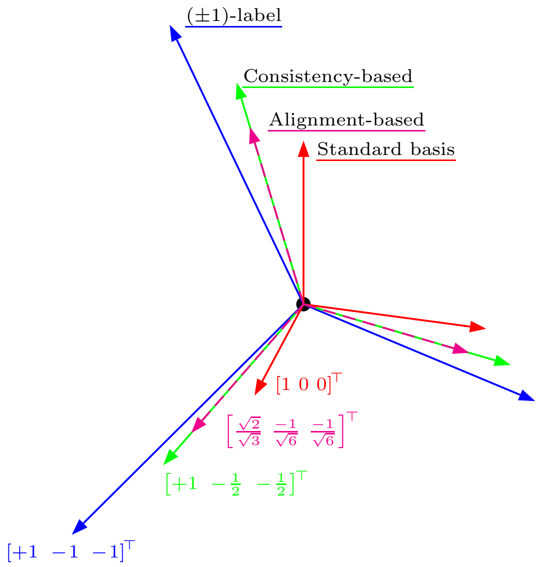
(±1)-label
Consistency-based
Alignment-based
Standard basis
$[1\ 0\ 0]^\top$
$\left[\frac{\sqrt{2}}{\sqrt{3}}\ \frac{-1}{\sqrt{6}}\ \frac{-1}{\sqrt{6}}\right]^\top$
$[+1\ -\frac{1}{2}\ -\frac{1}{2}]^\top$
$[+1\ -1\ -1]^\top$

Figure 2: Illustration of the (vector) labels in $\ell = 3$ dimensions, for well-known labelbooks.

- A straightforward generalization of the binary case to the multiclass case is to keep all the entries with -1 except a single $+1$ to identify the class. This labelbook is often suggested in the literature. See for instance [16, 18].
- The standard basis (also known as indicators) is used in artificial neural networks [30, 41] (e.g., Boltzmann networks and extreme learning machines), and is called dummy variables by statisticians. In this case, the labelbook is defined by the columns of the identity matrix.
- The alignment-based labelbook is optimal in the sense of the kernel-target alignment criterion [31], and has been systematically applied in many multiclass machines [27, 26].
- In [32], the authors study the consistency of the inductive principle in multiclass SVM. This desirable property is satisfied by the consistency-based labelbook.

See Fig. 2 for a three-dimensional illustration, i.e., a 3-class classification problem. For any of these labelbooks, we see that all the entries of a given label $\boldsymbol{y}_i$ are identical except for a single entry, which corresponds to the class membership. In what follows, we call such category of labelbooks the *one-per-class* labelbooks (short for one entry that identifies the class). This is similar to the notion of class-symmetric matrix as defined in [13], where the authors consider only $\{-1, +1\}$ entries. Still, in our definition, the values may be arbitrary. Table 1 presents several well-known labelbooks. We also present the corresponding inner products, since this is the essential information of any labelbook, as illustrated in the parameters estimation and the decision rule, in (11) and (25), namely

$$\arg\max_{\boldsymbol{y}} \boldsymbol{y}^\top \boldsymbol{Y} (\boldsymbol{K} + \gamma \boldsymbol{I})^{-1} \boldsymbol{\kappa}_{\boldsymbol{x}}.$$

where only inner products between labels is required.

Each one-per-class labelbook defines ℓ distinct (vector) labels of ℓ entries. Thus, the ℓ vectors lie in a ℓ-dimensional space. One may also consider a labelbook of ℓ vectors in an $\ell - 1$ dimensional space. For instance, when three classes are in competition, this corresponds to three vectors spanning an equilateral triangle. In [26], the authors generalize this result to any number of classes, by imposing that correlation between any distinct vectors is the same, and minimized. The resulting vectors are given as the solution of an eigen-problem.

Proposition 3. *The minimum-correlation labelbook is equivalent to the alignment labelbook.*

Proof. From [26, Proposition 2], it is shown that the minimum-correlation labelbook is defined by ℓ vectors spanning a $\ell - 1$ dimensional subspace and the inner-products of distinct vectors equal to $-1/(\ell-1)$. It turns out that this is exactly the inner-products of the alignment-based labelbook (see third column in Table 1). Since both labelbooks have the same $\boldsymbol{y}_i^\top \boldsymbol{y}_j$, they are equivalent for both vector-output machines and the oneLSM machines. □

The consistency-based and the alignment-based labelbooks are identical up to the multiplicative constant $\sqrt{(\ell-1)/\ell}$ (see Fig. 2). Thus, they give comparable results in multiclass classification, since the scale factor will be absorbed in optimization. In what follows, we study the one-per-class labelbooks (e.g., those given in Table 1) for the oneLSM machines.

Labels in oneLSM machines

Next, show that the one-per-class labelbooks are equivalent for the oneLSM optimization problem (24)-(25). But before, we show the link between these labelbooks.

Lemma 4. *The one-per-class labelbooks can be generated from each others using the linear transformation*

$$a\boldsymbol{y}_i + b\mathbf{1}, \tag{26}$$

from some arbitrary a and b, where $\boldsymbol{y}_i$ spanning some given labelbook and $\mathbf{1}$ is the column vector with ℓ ones.

The proof is straightforward, by generating any one-per-class labelbook from the standard basis labelbook, and vice versa. Therefore, we can define any of the one-per-class labelbook with elements $\boldsymbol{y}_i$ such that

$$[\boldsymbol{y}_i]_k = \begin{cases} a+b & \text{if } \boldsymbol{x}_i \text{ belongs to class } k; \\ b & \text{otherwise} \end{cases}$$

where the standard basis is implicitly used.

Theorem 5. *The one-per-class labelbooks are equivalent for the oneLSM.*

Proof. Consider the transformation according to (26), when all the labels, $\boldsymbol{y}^\top$ and $\boldsymbol{Y}$, are transformed into $a\boldsymbol{y} + b\mathbf{1}$ and $a\boldsymbol{Y} + b\mathbf{1}\mathbf{1}^\top$, respectively. Next, we show that the decision rule (25) is invariant to such transformation:

$$\begin{aligned} \arg\max_{\boldsymbol{y}\in\mathcal{Y}} \; (a\boldsymbol{y}+b\mathbf{1})^\top(a\boldsymbol{Y}+b\mathbf{1}\mathbf{1}^\top)(\boldsymbol{K}+\gamma\boldsymbol{I})^{-1}\boldsymbol{\kappa}_{\boldsymbol{x}} &= \arg\max_{\boldsymbol{y}\in\mathcal{Y}} \left(a^2\,\boldsymbol{y}^\top\boldsymbol{Y} + ab\,\boldsymbol{y}^\top\mathbf{1}\mathbf{1}^\top\right)(\boldsymbol{K}+\gamma\boldsymbol{I})^{-1}\boldsymbol{\kappa}_{\boldsymbol{x}} \\ &= \arg\max_{\boldsymbol{y}\in\mathcal{Y}} a^2\,\boldsymbol{y}^\top\boldsymbol{Y}(\boldsymbol{K}+\gamma\boldsymbol{I})^{-1}\boldsymbol{\kappa}_{\boldsymbol{x}} \\ &= \arg\max_{\boldsymbol{y}\in\mathcal{Y}} \boldsymbol{y}^\top\boldsymbol{Y}(\boldsymbol{K}+\gamma\boldsymbol{I})^{-1}\boldsymbol{\kappa}_{\boldsymbol{x}} \end{aligned}$$

where the first equality follows from removing terms independent of $\boldsymbol{y}$, and the second equality is due to the fact that $\boldsymbol{y}^\top\mathbf{1}$ is constant for all $\boldsymbol{y} \in \mathcal{Y}$ in a given labelbook. □

Corollary 6. *The one-per-class labelbooks are equivalent to the minimum-correlation labelbook.*

This corollary follows directly from combining Proposition 3 and Theorem 5.

Beyond establishing the equivalence between several labelbooks, Theorem 5 also provides the equivalence between the decision rule (27) and the maximum-value rule used in the classical OvA scheme. The proof is straightforward for the standard basis labelbook, since $\arg\max_{\boldsymbol{y}\in\mathcal{Y}} \boldsymbol{y}^\top \boldsymbol{f}(\boldsymbol{x})$ yields $\arg\max_{1\leq k\leq \ell} f^{(k)}(\boldsymbol{x})$. The extension of this result to other labelbooks is due to Theorem 5.

7. Discussions

7.1. The decision rule

Several decision rules can be implemented. In classical OvA scheme, the multiclass decision for any new sample $\boldsymbol{x}$ is obtained by selecting the class whose corresponding classifier $f_k(\boldsymbol{x})$ has maximum value, i.e., the winner-takes-all strategy. This can be easily applied in our approach, by simply inspecting the resulting vector-of-functions $\boldsymbol{f}(\boldsymbol{x})$. Other decision rules are the classical Hamming distance, or the less-common Bayesian distance measure which gives an estimation of the posterior probability; see [16].

By analogy with (2), we consider the following decision rule for any given observation $\boldsymbol{x}$:

$$\arg\max_{\boldsymbol{y}\in\mathcal{Y}} \boldsymbol{y}^\top \boldsymbol{f}(\boldsymbol{x}), \tag{27}$$

where the identified label determines the membership class. By substituting $\boldsymbol{f}(\boldsymbol{x})$ with its expression in (24), it is easy to see that the labels are only involved in terms of inner products, with $\boldsymbol{y}^\top \boldsymbol{Y}$. This decision rule is motivated in [26] for maximum margin in vector-output SVM. Since by construction one may set the norm of the label to be constant for a given labelbook[8], this decision criterion is equivalent to

$$\arg\min_{\boldsymbol{y}\in\mathcal{Y}} \|\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{y}\|^2.$$

It is worth noting that the proposed decision rule allows the use of any labelbook. However, the decision rule (27) shows a dominant influence of the label coding on the result. In the next subsection, we show that the considered inner-product form provides a non-linear kernel-like expansion.

7.2. Kernels on the labels

It is well known that linear classification techniques can be easily extended for nonlinear classification, i.e., where data are not linearly separable, thanks to the kernel trick. Moreover, the same rule can be applied for the labels. This is true since the proposed framework leads to algorithms that depend on the labels in terms of inner products $\boldsymbol{y}_i^\top \boldsymbol{y}_j$, as illustrated in this paper with vo-RLS, vo-LSSVM, vo-SVM, and oneLSM (see for instance (25)). Thus one may also propose a kernel function for the labels, namely

$$\kappa_y(\boldsymbol{y}_i, \boldsymbol{y}_j) = \Psi(\boldsymbol{y}_i)^\top \Psi(\boldsymbol{y}_j),$$

where the nonlinear function $\Psi(\cdot)$ maps the output space to some higher dimensional space. Nevertheless, the labels are by construction separable, as illustrated in Fig. 1. Therefore, we do not think that a kernel

[8] The squared norm of any $\boldsymbol{y}\in\mathcal{Y}$ is $\|\boldsymbol{y}\|^2 = \boldsymbol{y}^\top \boldsymbol{y}$, for instance ℓ for the (±1)-label. See the last column in Table 1 for other labelbooks.

Table 3: Databases under investigation, where n denotes the number of available samples with d attributes (dimensions) and ℓ classes.

	n	ℓ	d
iris	150	3	4
wine	178	3	13
glass	214	6	13
vowel	528	11	10
yeast	1 484	10	8
letters	up to 2 000	26	16
USPS	1 000	10	up to 64

function for labels is necessary in multiclass classification. Moreover, the choice of the optimal kernel as well as the optimal value of its parameter(s) is an open issue. We have conducted some preliminary experiments on applying a Gaussian kernel to the labels, and found no significant difference in performance when compared to the linear kernel on the labels.

Output kernels may be interesting as they allow us to include prior knowledge, e.g., incorporating the probability priors on the classes to deal with cases when the sizes of the classes are highly unbalanced by applying uneven weights. Another application of the label kernels provides an approach to tackle with the complex label structures, e.g. hierarchical relations. For example in [42], the authors propose a SVM-like algorithm by applying joint kernels between inputs and outputs. Joint kernels can be easily implemented within the framework proposed in paper, as illustrated by the different forms of vo-RLS, vo-LSSVM, SVM, and oneLSM.

In [29], the authors study transduction learning by solving a problem similar to the one given in Proposition 2 but with label kernels, i.e., by including implicitly some nonlinear transformation $\Psi(\cdot)$ on the labels. Predicting structured outputs, such as in graphs, the resulting problem requires to find the inverse transformation of $\Psi(\cdot)$. This is an ill-posed problem, where function $\Psi(\cdot)$ is not explicited thanks to the kernel trick. This inverse problem, known as the pre-image problem, is nonconvex and nonlinear. See [43] for a definition of the pre-image problem with techniques to solve this problem.

8. Experimentations

To illustrate the relevance of the proposed approach, we have tested all the algorithms on several well-known datasets (available at the UCI Repository): iris, wine, glass, vowel, and yeast. We also considered two datasets with large size and/or large dimensions and/or large number of classes: letters and USPS handwritten digit data [45]. See Table 3 for some statistics, including the number of samples n, the dimension d, and the number of classes ℓ. In order to provide a study that can be comparable with previous work, we considered a configuration given in [44], summarized as follows. Training data were normalized into the range $[-1, 1]$ with the appropriate scaling and bias factors, which were applied to the unlabelled test data.

[9] In [41], the Vowel dataset contains 528 + 462 instances for training+test, as opposed to the dataset studied in several studies [44, 13], including this paper.

[9] In [41], the Vowel dataset contains 528 + 462 instances for training+test, as opposed to the dataset studied in several studies [44, 13], including this paper.

Table 4: A comparative study of the classification error of several vector-output algorithms, using the same configuration as in Table 4 (the errors are given in the format $mean\vert^{worst}_{best}$).

			iris	wine	glass	vowel	yeast	
this paper	oneLSM		$2.80\vert^{3.33}_{2.00}$	$0.33\vert^{1.11}_{0.00}$	$27.39\vert^{28.7}_{24.9}$	$0.60\vert^{0.93}_{0.37}$	$38.91\vert^{39.4}_{38.1}$	
	vo-LSSVM	(±1)-label	$2.80\vert^{3.33}_{2.66}$	$2.06\vert^{2.77}_{1.11}$	$28.40\vert^{31.6}_{25.4}$	$0.62\vert^{0.93}_{0.37}$	$40.78\vert^{41.9}_{39.8}$	
		indicators	$2.80\vert^{3.33}_{2.00}$	$1.95\vert^{2.81}_{1.11}$	$27.67\vert^{30.2}_{25.5}$	$0.60\vert^{0.93}_{0.37}$	$39.82\vert^{40.3}_{39.5}$	
		alignment	$2.66\vert^{4.00}_{2.66}$	$1.35\vert^{1.69}_{1.11}$	$26.70\vert^{29.3}_{24.1}$	$0.60\vert^{0.93}_{0.37}$	$39.81\vert^{40.1}_{39.4}$	
		consistency	$3.00\vert^{4.00}_{2.66}$	$1.62\vert^{2.25}_{1.11}$	$26.61\vert^{29.3}_{23.6}$	$0.60\vert^{0.93}_{0.37}$	$39.86\vert^{40.3}_{39.4}$	
		min-corr.	$3.00\vert^{4.00}_{2.66}$	$1.62\vert^{2.25}_{1.11}$	$26.61\vert^{29.3}_{23.6}$	$0.60\vert^{0.93}_{0.37}$	$39.86\vert^{40.3}_{39.4}$	
	vo-RLS($\boldsymbol{\beta}$)	(±1)-label	$3.13\vert^{4.00}_{2.66}$	$1.45\vert^{2.25}_{1.11}$	$27.65\vert^{29.0}_{25.5}$	$0.60\vert^{0.93}_{0.37}$	$38.79\vert^{39.2}_{38.4}$	
		indicators	$3.13\vert^{4.00}_{2.66}$	$1.68\vert^{2.25}_{1.11}$	$27.80\vert^{29.2}_{25.9}$	$0.62\vert^{0.93}_{0.37}$	$38.94\vert^{39.1}_{38.2}$	
		alignment	$3.20\vert^{4.00}_{2.66}$	$1.28\vert^{1.69}_{0.58}$	$27.85\vert^{29.2}_{25.6}$	$0.62\vert^{0.93}_{0.37}$	$38.59\vert^{39.1}_{38.0}$	
		consistency	$3.13\vert^{4.00}_{2.66}$	$1.34\vert^{2.25}_{0.58}$	$27.71\vert^{28.8}_{25.6}$	$0.62\vert^{0.93}_{0.37}$	$38.62\vert^{39.0}_{38.0}$	
		min-corr.	$3.20\vert^{4.00}_{2.66}$	$1.28\vert^{1.69}_{0.58}$	$27.85\vert^{29.2}_{25.6}$	$0.62\vert^{0.93}_{0.37}$	$38.59\vert^{39.1}_{38.0}$	
	vo-RLS($\boldsymbol{f}$)	(±1)-label	$3.40\vert^{4.00}_{2.66}$	$1.62\vert^{2.25}_{1.11}$	$27.85\vert^{29.0}_{25.9}$	$0.60\vert^{0.93}_{0.37}$	$38.76\vert^{39.2}_{38.2}$	
		indicators	$3.40\vert^{4.00}_{2.66}$	$1.40\vert^{2.25}_{1.11}$	$27.93\vert^{29.2}_{25.9}$	$0.62\vert^{0.93}_{0.37}$	$38.72\vert^{39.2}_{38.3}$	
		alignment	$3.33\vert^{4.00}_{2.66}$	$1.28\vert^{1.69}_{0.58}$	$27.84\vert^{29.2}_{25.6}$	$0.62\vert^{0.93}_{0.37}$	$38.63\vert^{39.1}_{38.1}$	
		consistency	$3.40\vert^{4.00}_{2.66}$	$1.28\vert^{2.25}_{0.58}$	$27.83\vert^{29.2}_{25.6}$	$0.62\vert^{0.93}_{0.37}$	$38.67\vert^{39.2}_{38.2}$	
		min-corr.	$3.40\vert^{4.00}_{2.66}$	$1.28\vert^{2.25}_{0.58}$	$27.83\vert^{29.2}_{25.6}$	$0.62\vert^{0.93}_{0.37}$	$38.67\vert^{39.2}_{38.2}$	
	vo-SVM	(±1)-label	$3.13\vert^{4.00}_{2.66}$	$1.84\vert^{2.29}_{1.11}$	$27.68\vert^{29.7}_{25.0}$	$0.62\vert^{0.93}_{0.37}$	$39.43\vert^{40.0}_{38.9}$	
		indicators	$3.13\vert^{4.00}_{2.66}$	$1.84\vert^{2.71}_{1.11}$	$26.20\vert^{28.2}_{24.5}$	$0.60\vert^{0.93}_{0.37}$	$39.91\vert^{40.3}_{39.6}$	
		alignment	$2.86\vert^{3.33}_{2.00}$	$1.68\vert^{2.25}_{1.11}$	$27.08\vert^{28.9}_{25.1}$	$0.60\vert^{0.93}_{0.37}$	$39.74\vert^{40.5}_{39.3}$	
		consistency	$2.86\vert^{3.33}_{2.00}$	$1.68\vert^{2.25}_{1.11}$	$27.08\vert^{28.9}_{25.1}$	$0.60\vert^{0.93}_{0.37}$	$39.74\vert^{40.5}_{39.3}$	
		min-corr.	$2.86\vert^{3.33}_{2.00}$	$1.68\vert^{2.25}_{1.11}$	$27.08\vert^{28.9}_{25.1}$	$0.60\vert^{0.93}_{0.37}$	$39.74\vert^{40.5}_{39.3}$	
	Szedmak *et al.* [26]	(±1)-label	$3.73\vert^{4.66}_{2.66}$	$2.45\vert^{3.36}_{1.11}$	$34.34\vert^{35.9}_{32.0}$	$0.83\vert^{1.32}_{0.37}$	$43.12\vert^{43.9}_{42.4}$	
		indicators	$3.40\vert^{4.00}_{2.66}$	$2.51\vert^{2.84}_{2.19}$	$30.09\vert^{31.2}_{28.2}$	$0.60\vert^{0.93}_{0.37}$	$42.37\vert^{42.9}_{41.7}$	
		alignment	$3.40\vert^{4.00}_{2.66}$	$1.67\vert^{2.22}_{1.11}$	$32.23\vert^{33.4}_{31.1}$	$0.75\vert^{1.12}_{0.37}$	$45.02\vert^{45.4}_{44.4}$	
		consistency	$3.53\vert^{4.00}_{2.66}$	$1.67\vert^{2.22}_{1.11}$	$32.18\vert^{34.8}_{29.7}$	$0.75\vert^{1.12}_{0.37}$	$45.00\vert^{45.4}_{44.4}$	
		min-corr.	$3.33\vert^{4.00}_{2.66}$	$3.19\vert^{3.92}_{2.77}$	$33.04\vert^{34.9}_{31.5}$	$0.73\vert^{1.12}_{0.37}$	$45.00\vert^{45.7}_{44.2}$	
	LS	OvA	$2.80\vert^{3.33}_{2.00}$	$0.33\vert^{1.11}_{0.00}$	$27.39\vert^{28.7}_{24.9}$	$0.60\vert^{0.93}_{0.37}$	$38.91\vert^{39.4}_{38.1}$	
		OvO	$2.80\vert^{3.33}_{2.00}$	$0.33\vert^{1.11}_{0.00}$	$27.49\vert^{29.0}_{24.1}$	$0.58\vert^{0.93}_{0.37}$	$38.92\vert^{39.3}_{38.3}$	
	SVM	OvA	3.33	1.12	28.03	1.51	—	[44]
		OvO	2.66	0.56	28.50	0.94	—	
		DAG	3.33	1.12	26.16	1.32	—	
	single	Weston's [10]	2.66	1.12	28.97	1.51	—	
		Crammer's [11]	2.66	1.12	28.03	1.32	—	
		ANN ELM	3.96	1.52	31.59	41.33^{9}	—	[41]

results of the last 6 rows are borrowed from ↑

Table 5: A comparative study of the estimated computation time for several multiclass algorithms, in *hh:mm:ss*.

	least-squares machines			vector-output machines		
	OvA	OvO	**oneLSM**	**vo-LSSVM**	**vo-RLS**	**vo-SVM**
iris	*1:42*	*1:44*	*38*	*43*	*1:06*	*1:00*
wine	*3:41*	*3:39*	*1:14*	*1:19*	*2:31*	*9:32*
glass	*7:10*	*18:22*	*1:22*	*1:27*	*2:45*	*12:12*
vowel	*1:48:28*	*8:56:00*	*10:32*	*11:48*	*40:51*	*45:35*
yeast	*8:26:45*	*37:36:24*	*54:29*	*1:07:09*	*3:21:06*	*2:40:40*
			see Sec. 5	see Sec. 4		

Table 6: A comparative study of the classification error and computation time (in *hh:mm:ss*) between several multiclass algorithms for the letters ($\ell = 26$ classes) and USPS ($\ell = 10$ classes) datasets.

		USPS ($\ell = 10$)		letters ($\ell = 26$)
	$n =$	1 000	2 000	2 000
	$d =$	8×8	8×8	16
this paper	oneLSM	6.50 (7:42)	4.59 (46:32)	10.69 (25:27)
	vo-LSSVM	8.41 (7:54)	6.19 (46:02)	14.08 (29:08)
	vo-SVM	8.80 (10:02)	6.54 (41:23)	14.08 (27:15)
	LS OvA	6.50 (1:06:36)	4.59 (7:04:15)	10.69 (10:29:03)
	LS OvO	6.50 (4:56:37)	4.59 (31:06:11)	10.69 (133:10:22)
	[26]	16.61 (9:42)	12.69 (41:52)	15.04 (28:23)

We also applied the same kernel function, which is the Gaussian kernel. To estimate the classification error, a ten-fold cross-validation was used, with parameters optimized by grid search over $\gamma \in \{2^{-4}; 2^{-3}; \cdots; 2^3; 2^4\}$ and $\sigma \in \{2^{-4}; 2^{-3}; \cdots; 2^3; 2^4\}$. This configuration is very similar to the one given in [13], where the authors opted for optimizing separately both parameters, thus sub-optimal as opposed to our joint optimization over both γ and σ.

In Table 4, we give the accuracy rate for each machine, where the values were averaged over 10 Monte Carlo simulations. For a comparative study to other machines, the same partitions for cross-validation were applied to conventional OvA and OvO schemes with LSM machines, as well as vector-output SVM as defined in [26]. We borrowed the last column of the Table 4 from [41], where ANN extreme learning machines were applied, the optimal parameters being estimated on a wider grid search $\{2^{-24}; 2^{-23}; \cdots; 2^{24}; 2^{25}\}$. We also borrowed 5 columns of the Table 4 from [44], with essentially the same configuration while requiring more sophisticated algorithms to train these SVM machines. The proposed single matrix inversion scheme is as competitive as these machines in terms of accuracy.

The computation time was estimated over the ten-fold cross-validation with the grid search for the optimal parameters identification, as illustrated in Table 5 where values were averaged over 10 Monte Carlo simulations. The computational time was estimated on a Matlab 64-bit running on a Macbook Pro laptop[10]. As illustrated with the average running time, it is obvious that the proposed approach highly boosts the speed of the multiclass machine.

We also studied the proposed approach on two large scale datasets: letters (from the UCI Repository) and USPS handwritten digit data [45]. The USPS dataset consists of a large number of scans of the $\ell = 10$ digits, in a 16×16 grey level images. We considered the same settings as in [46], with pixel intensities normalized within $[-1, 1]$. As stated in [41], both the letters and the USPS datasets require high-performance computers for classification. In order to give a comparative study with different multiclass classification methods, we considered several reduced datasets, with $n = 1\,000$ and $n = 2\,000$ samples, where the USPS images were resampled into $d = 8 \times 8$ images. The classification error was estimated using a ten-fold cross-validation, with parameters optimized by the same aforementioned grid search. Table 6 shows the estimated classification error and the computational time. It is easy to see that the classical schemes, applying OvA or OVO, are inappropriate for such large scale datasets. The multiclass SVM algorithm proposed by Szedmak *et al.* in [26] is more suitable for this task, still it is outperformed by all the methods proposed in this paper.

Statistical test

Several statistical tests have been derived in the literature to address the problem of comparing two classifiers. In [47], five statistical tests are analyzed, including a new one, the 5×2 cv t-test based on 5 iterations of 2-fold cross-validation. The latter test has low type I error and is more powerful than other tests including the McNemar's test and the classical two-tailed t test. This study is extended further in [48], where a combined 5×2 cv F-test is proposed.

The combined 5×2 cv F-test for comparing any two classification algorithms is described as follows. Five replications of two-fold cross-validations are performed, where each replication partitions the data into two halves: one half is used in training[11] and the other one in testing, and vice-versa. Let $p_i^{(j)}$ be the difference in the error rates of the two classifiers in fold j of the i-th iteration, for $j = 1, 2$ and $i = 1, 2, \ldots, 5$. By denoting $\overline{p}_i = \frac{1}{2}(p_i^{(1)} + p_i^{(2)})$ the mean, and $s_i^2 = (p_i^{(1)} - \overline{p}_i)^2 + (p_i^{(2)} - \overline{p}_i)^2$ the variance, then

$$f = \frac{\sum_{i=1}^{5} \sum_{j=1}^{2} \left(p_i^{(j)}\right)^2}{2 \sum_{i=1}^{5} s_i^2}$$

is approximately F distributed with 10 and 5 degrees of freedom. The null hypothesis that the two algorithms have the same performances is rejected, at statistical significance level of 0.95, if $f > 4.74$.

We analyzed the influence of the label on the performance, and compared all the machines proposed in this paper. We considered the application of the above F-test, and found that there is no significative difference (at 95% level). We also compared the multiclass SVM as derived by Szedmak *et al.*, where a biased model is considered [26]. We found that the performance of their method highly depends on the label coding, a property confirmed by the above F-test. Table 4 shows that their method has, in almost all cases, worst performance than all the methods proposed in this paper.

[10] Mackbook Pro with a 2.53 GHz Intel Core 2 Duo processor and 4 GB RAM.

[11] As recommended in [49], the training set is also used for estimating the optimal parameters. To this end, we used a ten-fold cross-validation on the training set, with parameters optimized by grid search over $\gamma \in \{2^{-4}; 2^{-3}; \cdots; 2^3; 2^4\}$ and $\sigma \in \{2^{-4}; 2^{-3}; \cdots; 2^3; 2^4\}$. The testing set is not considered at all by the algorithms in the training stage.

9. Conclusion

In this paper, our main thesis was that simple multiclass classification machines can be designed with the same complexity as a single binary classifier. Several algorithms were developed for this purpose. We also provided a theoretical study on the coding of the labels, and showed that several well-known labelbooks are equivalent. Experiments conducted on well-known datasets show that the resulting machines are faster than classical ones, and performs just as well as other machines in the literature.

As for future work, we plan to extend the proposed approach to the one-versus-one scheme. While this is less straightforward, one can still take advantage of classical linear algebra to reduce its computational complexity. Also, it would be desirable to derive an optimal labelbook for a given classification task, for instance by optimizing a kernel function applied on the labels. We also plan to consider an optimization on symmetric positive-definite matrices manifold, in the same spirit of [50]. The sparsity of the solution is also of great interest, in the same spirit as the framework of compressed sensing, with ℓ_1 or even ℓ_0 norms as opposed to the ℓ_2 norm given in this paper.

References

[1] V. N. Vapnik, *Statistical Learning Theory.* Wiley-Interscience, 1998.

[2] J. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Processing Letters*, vol. 9, pp. 293–300, 1999. [Online]. Available: http://dx.doi.org/10.1023/A:1018628609742

[3] R. Rifkin, G. Yeo, and T. Poggio, "Regularized least squares classification," in *Advances in Learning Theory: Methods, Model and Applications*, vol. 190, Amsterdam, 2003, pp. 131–172.

[4] R. El-Yaniv, D. Pechyony, and E. Yom-Tov, "Better multiclass classification via a margin-optimized single binary problem," *Pattern Recognition Letters*, vol. 29, pp. 1954–1959, October 2008.

[5] Y. Liu, Z. You, and L. Cao, "A novel and quick SVM-based multi-class classifier," *Pattern Recognition*, vol. 39, no. 11, pp. 2258–2264, 2006.

[6] L. Galluccio, O. Michel, P. Comon, and A. O. Hero, "Graph based k-means clustering," *Signal Processing*, vol. 92, no. 9, pp. 1970 – 1984, 2012.

[7] W.-J. Zeng, X.-L. Li, X.-D. Zhang, and E. Cheng, "Kernel-based nonlinear discriminant analysis using minimum squared errors criterion for multiclass and undersampled problems," *Signal Processing*, vol. 90, no. 8, pp. 2333 – 2343, 2010.

[8] P. Wahlberg and G. Salomonsson, "Methods for alignment of multi-class signal sets," *Signal Processing*, vol. 83, no. 5, pp. 983 – 1000, 2003.

[9] M. E. Aladjem, "Multiclass discriminant mappings," *Signal Processing*, vol. 35, no. 1, pp. 1–18, Jan. 1994.

[10] J. Weston and C. Watkins, *Support Vector Machines for Multi-Class Pattern Recognition.* Proc. Seventh European Symposium On Artificial Neural Networks, 1999, vol. 4, no. 6, pp. 219–224.

[11] K. Crammer and Y. Singer, "On the algorithmic implementation of multiclass kernel-based vector machines," *Journal of Machine Learning Research*, vol. 1, pp. 143–160, 2002.

[12] E. J. Bredensteiner and K. P. Bennett, "Multicategory classification by support vector machines," *Comput. Optim. Appl.*, vol. 12, pp. 53–79, January 1999.

[13] R. Rifkin and A. Klautau, "In defense of one-vs-all classification," *Journal of Machine Learning Research*, vol. 5, pp. 101–141, 2004.

[14] J. Fürnkranz, "Round robin classification," *Journal of Machine Learning Research*, vol. 2, pp. 721–747, 2002.

[15] J. C. Platt, N. Cristianini, and J. Shawe-Taylor, "Large margin DAGs for multiclass classification," in *Proc. of Neural Information Processing Systems, NIPS'99.* MIT Press, 2000, pp. 547–553.

[16] T. G. Dietterich and G. Bakiri, "Solving multiclass learning problems via error-correcting output codes," *Journal of Artificial Intelligence Research*, vol. 2, pp. 263–286, January 1995.

[17] G. Ou and Y. Murphey, "Multi-class pattern classification using neural networks," *Pattern Recognition*, vol. 40, no. 1, pp. 4–18, 2007.

[18] E. L. Allwein, R. E. Schapire, and Y. Singer, "Reducing multiclass to binary: a unifying approach for margin classifiers," *Journal of Machine Learning Research*, vol. 1, pp. 113–141, September 2001.

[19] J. A. K. Suykens, T. V. Gestel, J. D. Brabanter, B. D. Moor, and J. Vandewalle, *Least Squares Support Vector Machines.* Singapore: World Scientific Pub. Co., 2002.

[20] T. Van Gestel, J. A. K. Suykens, B. Baesens, S. Viaene, J. Vanthienen, G. Dedene, B. De Moor, and J. Vandewalle, "Benchmarking least squares support vector machine classifiers," *Mach. Learn.*, vol. 54, no. 1, pp. 5–32, Jan. 2004.

[21] G. Kimeldorf and G. Wahba, "Some results on tchebycheffian spline functions," *Journal of Mathematical Analysis and Applications*, vol. 33, pp. 82–95, 1971.

[22] B. Schölkopf, R. Herbrich, and A. J. Smola, "A generalized representer theorem," in *Proc. 14th Annual Conference on Computational Learning Theory and 5th European Conference on Computational Learning Theory.* Springer-Verlag, 2001.

[23] C. A. Micchelli and M. A. Pontil, "On learning vector-valued functions," *Neural Comput.*, vol. 17, pp. 177–204, January 2005.

[24] T. Evgeniou, C. A. Micchelli, and M. Pontil, "Learning multiple tasks with kernel methods," *Journal of Machine Learning Research*, vol. 6, pp. 615–637, 2005.

[25] S. Szedmak and J. Shawe-Taylor, "Multiclass learning at one-class complexity," 2005, technical Report, ISIS Group, Electronics and Computer Science.(Unpublished).

[26] S. Szedmak, J. Shawe-Taylor, and E. Parado-Hernandez, "Learning via linear operators: Maximum margin regression; multiclass and multiview learning at one-class complexity," University of Southampton, Tech. Rep., 2006.

[27] Z. Noumir, P. Honeine, and C. Richard, "Multi-class least squares classification at binary-classification complexity," in *Proc. IEEE workshop on Statistical Signal Processing*, Nice, France, 28–30 June 2011.

[28] ——, "Classification multi-classes au prix d'un classifieur binaire," in *Actes du 23-me Colloque GRETSI sur le Traitement du Signal et des Images*, Bordeaux, France, September 2011.

[29] C. Cortes, M. Mohri, and J. Weston, "A general regression framework for learning string-to-string mappings," in *Predicting Structured Data.* MIT Press, 2007.

[30] C. Bishop, *Neural networks for pattern recognition.* Oxford University Press, 1995.

[31] Y. Guermeur, "SVM multiclasses, théorie et applications," Habilitation à diriger des recherches, Université Nancy I, Jan. 2008.

[32] Y. Lee, Y. Lin, and G. Wahba, "Multicategory support vector machines, theory, and application to the classification of microarray data and satellite radiance data," *Journal of the American Statistical Association*, vol. 99, pp. 67–81, 2004.

[33] T. Poggio, S. Mukherjee, R. Rifkin, A. Rakhlin, and A. Verri., "b," *Uncertainty in Geometric Computations*, pp. 131–141, 2002.

[34] S. Voloshynovskiy, O. Koval, F. Beekhof, and T. Holotyak, "Information—theoretic multiclass classification based on binary classifiers," *J. Signal Process. Syst.*, vol. 65, no. 3, pp. 413–430, Dec. 2011.

[35] J. Suykens and J. Vandewalle, "Multiclass least squares support vector machines," in *Proc. International Joint Conference on Neural Networks.* World Scientific, 1999.

[36] V. N. Vapnik, *The nature of statistical learning theory.* New York, NY, USA: Springer-Verlag New York, Inc., 1995.

[37] L. G. Abril, C. Angulo, F. Velasco, and J. A. Ortega, "A note on the bias in svms for multiclassification," *IEEE Transactions on Neural Networks*, vol. 19, no. 4, pp. 723–725, 2008.

[38] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* Cambridge, MA, USA: MIT Press, 2001.

[39] Ş. Ertekin, L. Bottou, and C. L. Giles, "Nonconvex online support vector machines," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, pp. 368–381, 2011.

[40] R. Rifkin, "Everything old is new again: A fresh look at historical approaches in machines learning," in *PhD thesis, MIT*, 2002.

[41] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 2012, in press.

[42] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, "Large margin methods for structured and interdependent output variables," *Journal of Machine Learning Research*, vol. 6, pp. 1453–1484, 2005.

[43] P. Honeine and C. Richard, "Preimage problem in kernel-based machine learning," *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 77–88, 2011.

[44] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multi-class support vector machines," *IEEE Transactions on Neural Networks*, vol. 13, pp. 415–425, 2002.

[45] J. J. Hull, "A database for handwritten text recognition research," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, pp. 550–554, May 1994.

[46] C. E. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning.* MIT Press, 2006. [Online]. Available: http://www.gaussianprocess.org/gpml/

[47] T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," *Neural Computation*, vol. 10, pp. 1895–1923, 1998.

[48] E. Alpaydin, "Combined 5 x 2 cv f test for comparing supervised classification learning algorithms," *Neural Comput.*, vol. 11, no. 8, pp. 1885–1892, Nov. 1999.

[49] E. Cantú-Paz and C. Kamath, "An empirical comparison of combinations of evolutionary algorithms and neural networks for classification problems." *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, pp. 915–927, 2005.

[50] H. Laanaya, F. Abdallah, H. Snoussi, and C. Richard, "Learning general gaussian kernel hyperparameters of svms using optimization on symmetric positive-definite matrices manifold," *Pattern Recogn. Lett.*, vol. 32, no. 13, pp. 1511–1515, Oct. 2011.