Probabilistic Vector Machines

A. Pedro Duarte Silva
Católica Porto Business School & CEGE
Universidade Católica Portuguesa

1 Introduction

Kernel based Support Vector Machines (SVMs) were originally designed to handle two-class supervised classification problems, and quickly established themselves as one of the most accurate machine learning algorithms for class prediction. However, this success did not translate to the related task of deriving reliable probability estimates of class membership. In fact, Lin (2002) has shown that, by targeting directly classification boundaries, standard SVMs do not carry much information about class probabilities other than the predicted class by itself. Nevertheless, Lin, Lee and Wahba (Lin et al., 2002) showed that, by appropriately modifying (weighting) the loss function used in standard SVMs, nonstandard SVMs can estimate consistently a theoretical Bayes rule for any arbitrary setting of class probabilities. Based on this property, Wang, Shen and Liu (Wang et al., 2008) proposed to solve sequences of nonstandard SVMs with varying weight specifications, and to recover class probabilities from the frontiers between regions of the weights domain that lead to different predictions.

The first proposal to extend this idea to the general k-class problems, is an *all-in-one* approach due to Wu, Zhang and Liu (Wu et al., 2010) (WZL). However, in this proposal the number of base weighted SVMs increases exponentially with the number of classes, and their training requires the optimisation of non-convex problems, making the method impractical for big, or even moderate, data problems. Multiclass probability estimation based on pairwise *one-against-the-rest* weighted SVMs were proposed in Xu and Wang (2013) and Wang et al. (2019).

This work addresses the computational difficulties associated with the WZL *all-in-one* approach, and compares its statistical performance against competing alternatives. In particular, on the one hand, we will propose an improved method for recovering class probability estimates from weighted SVM predictions. In our approach, these estimates will be based on the solutions of linear programming models that optimize an l_1-norm measure of the agreement between the predictions implied by probability estimates, and those made by weighted SVMs. One important advantage of this strategy is that, unlike in the original WZL method, the different weight specifications do not have to be uniformly distributed over a k-dimensional simplex, which allows for the creation of grids with satisfactory resolution, while ensuring that the number of required weighted SVMs only grows linearly with the number of different classes. On the other hand, we propose to employ an universal kernel without bias terms, using the weighted loss proposed by Lin, Lee and Wahba

(Lee et al., 2004) (LLW). We note that the LLW loss leads to convex optimization problems and, as noted in Dogan et al. (2016), for multiclass SVMs based on universal kernels, dropping bias terms is of minor importance in terms of statistical properties, and allows the use of computationally efficient decomposition algorithms for SVM training. Based on these strategies, we were able the find reliable class probability estimates for problems with hundreds of examples, and more than a dozen different classes.

Simulation results suggest that class probability estimation based on weighted SVMs are usually more accurate than competing distribution free machine learning approaches, and more reliable than model based statistical methodologies when their assumptions fail. Amongst the SVM based methods, no alternative is universally superior to the others, and the best method seems to depend on the particular data conditions at hand.

The remainder of this paper is organized as follows. Section 2 introduces notation, and reviews weighted multiclass SVM formulations. Section 3 formalizes the method of estimating class estimates from sequences of weighted SVMs predictions. Section 4 describes the SVM training algorithm proposed. Section 5 presents controlled simulation experiments comparing our proposal with the most important alternatives, and Section 6 concludes the paper.

2 Multiclass Probability SVMs

Let $\mathcal{T} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ be a training set of n examples, where x_i is an attribute descriptor belonging to some domain, $\mathcal{X}$, the label y_i is an integer belonging to the set $\mathcal{Y} = \{1, \ldots, k\}$, and all pairs (x_i, y_i) were independently generated from some unknown, but common, probability distribution, $P(\mathbf{X}, Y)$. Based on $\mathcal{T}$, we are interested in developing estimator functions, $\mathbf{p}_c(\mathbf{x})\ ;\ c \in \mathcal{Y}$, for the *posterior* class probabilities:

$$\mathbf{p}_c(\mathbf{x}) = P(Y = c|\mathbf{X} = \mathbf{x}) = \frac{P(\mathbf{x}, c)}{\sum_{c' \in \mathcal{Y}} P(\mathbf{x}, \mathbf{y}_{c'})} \tag{1}$$

These estimators are to be recovered from a sequence of nonstandard (weighted) multiclass SVMs yielding decisions rules with general form

$$\hat{y} = argmax_c\ \mathbf{f_c}(\mathbf{x}) \tag{2}$$

where $\mathbf{f_c}$ is the c^{th} element of the vector function $\mathbf{f} : \mathcal{X} \mapsto \mathbb{R}^k$ that solves the optimization problem:

$$min_{\mathbf{f} \in \mathcal{F}^k} \quad n^{-1} \sum_{i=1}^{n} \boldsymbol{\pi}_{y_i} L(\mathbf{f}(\mathbf{x}_i), y_i) + \lambda\, J(\mathbf{f}) \tag{3}$$

$$subject\ to \qquad \sum_{c \in \mathcal{Y}} \mathbf{f}_c = \mathbf{0} \tag{4}$$

Here, $\mathcal{F}^k$ is a cartesian product of some known functional space $\mathcal{F}$, the penalty operator $J : \mathcal{F}^k \mapsto \mathbb{R}_0^+$ measures model complexity, $\lambda \in \mathbb{R}^+$ is a regularization parameter that controls the trade-off between the smoothness of $\mathbf{f}$ and the multi-class large margin loss $L : \mathbb{R}^k \times \mathcal{Y} \mapsto \mathbb{R}_{\nearrow}^+$, and the weighting vector $\boldsymbol{\pi}$ belongs to the k-dimensional simplex, $A_k = \boldsymbol{\pi} \in \mathbb{R}^k : \sum_{c=1}^{k} \boldsymbol{\pi}_c = 1\ ,\ \forall c\ \boldsymbol{\pi}_c \geq 0\}$.

In this paper we will study SVMs based on universel kernels, where $\mathcal{F}$ is a strictly positive definite Reproducing Kernel Hilbert Space (RKHS), $\mathcal{H}_{\mathbf{K}}$, induced by some known kernel function $K : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$, and endowed by the norm $||.||_{\mathcal{H}_{\mathbf{K}}}$ (Wahba (1998), Cristianini and Shawe-Taylor (2000), Poggio et al. (2002)). Then, the representer theorem (Kimeldorf and Wahba (1971)) implies that for all $\mathbf{x} \in \mathcal{X}$ and $c \in \mathcal{Y}$, $\mathbf{f_c}(\mathbf{x})$ and $||f_c||^2_{\mathcal{H}_{\mathbf{K}}}$ can be expressed as $\mathbf{f_c}(\mathbf{x}) = \sum_{i=1}^{n} \theta_i^c \ K(\mathbf{x_i}, \mathbf{x}),\ ||f_c||^2_{\mathcal{H}_{\mathbf{K}}} = \sum_{i=1}^{n} \sum_{j=1}^{n} \theta_i^c \ \theta_j^c \ K(\mathbf{x_i}, \mathbf{x_j})$, $\theta^c \in \mathbb{R}^n$, and the penalty $J(.)$ is typically chosen as the sum of the squared norms of the $\mathbf{f}$ components, *i.e.*, $J(\mathbf{f}) = \sum_{c=1}^{k} ||\mathbf{f}_c||^2_{\mathcal{H}_{\mathbf{K}}}$.

We note that this framework differs from the traditional one assumed in most SVMs, in that our classification functions $\mathbf{f}$ do not include bias terms. Poggio et al. (2002)) show that the general approximation properties of strictly positive definite Reproducing Kernel Hilbert Spaces do not require such terms, and Dogan et al. (2016) recommend dropping them from standard multiclass SVMs, since this can lead to substantial computational gains without affecting the main statistical properties of the resulting classifiers. We followed Dogan's recommendation and our numerical comparisons, to be described in Section 5, suggest that the resulting weighted SVMs are not adversely affected by this choice.

Different weighted versions of known multi-class SVMs can be specified as particular cases of decision rule (2) and optimization model (3) (4), by choosing particular loss functions. In this paper, we will estimate class probabilities from the class predictions given by the weighted LLW loss proposed in Lee et al. (2004):

$$L_{LLW}(\mathbf{f}, y) \ = \sum_{c \in \mathcal{Y} \setminus \{y\}} (\mathbf{f}_c + (k-1)^{-1})_+ \tag{5}$$

where $(u)_+ := max(0, u)$.

Advantages of using the LLW loss include the facts that this loss is weighted Fisher consistent (see Wu et al. (2010), and Lee et al. (2004)), and that the training of the resulting SVMs leads to convex optimization models. In Section 4 we will describe an efficient algorithm to train weighted LLW SVMs.

3 Recovering Class Probabilities from Class Predictions

In order to estimate class probabilities from the predictions given by weighted SVMs, we need first to train several weighted SVMs with different weight specifications. Let $\mathcal{G}$ be the grid of different specifications for the weighting vector $\boldsymbol{\pi}$. In Wu et al. (2010), $\mathcal{G}$ is defined as an uniformly distributed set of points over A_k. A consequence of this choice is that the number of grid points, $\#\mathcal{G}$, grows exponentially with k.

In this paper we propose a novel method to recover class probabilities from class predictions, where $\#\mathcal{G}$ only increases linearly with k. Our approach is based on optimizing an heuristic l_1-norm measure of the agreement between the weighted SVMs predictions and the optimal classification rules implied by the $\mathbf{p}_c(\mathbf{x})$ estimates. In particular, consider the weight specification $\boldsymbol{\pi}^g$ and the corresponding weighted SVM prediction $\hat{y}_g$. Assuming the theoretical cost weighted classification problem,

$min_{\mathbf{f}}\, E[\boldsymbol{\pi}^g_Y L(\mathbf{f}(\mathbf{X}), Y)|\mathbf{X} = \mathbf{x})]$ $s.t.\sum_{c\in\mathcal{Y}} \mathbf{f}_c = \mathbf{0}$, the SVM prediction agrees with the optimal Bayes rule for a vector of *a-posteriori* probabilities, $\mathbf{p}_c(\mathbf{x})$, iff

$$\boldsymbol{\pi}^g_{\hat{y}_g}\mathbf{p}_{\hat{y}_g}(\mathbf{x}) \geq \boldsymbol{\pi}^g_c\,\mathbf{p}_c(\mathbf{x}) \qquad \forall c \in \mathcal{Y}\backslash\{\hat{y}_g\} \tag{6}$$

When it is possible to find $\mathbf{p}(\mathbf{x})$ vectors such that (6) is satisfied for all $\boldsymbol{\pi}^g \in \mathcal{G}$, a reasonable estimation criterion is to choose amongst such vectors, the one that maximizes the desirable sum of the l_1-norm margin deviations $\boldsymbol{\pi}^g_{\hat{y}_g}\mathbf{p}_{\hat{y}_g}(\mathbf{x}) - \boldsymbol{\pi}^g_c\mathbf{p}_c(\mathbf{x})$. On the other hand, when it is not possible to find a vector estimate that always satisfies (6), one may search for one the minimizes the undesirable l_1-norm deviations $\boldsymbol{\pi}^g_c\mathbf{p}_c(\mathbf{x}) - \boldsymbol{\pi}^g_{\hat{y}_g}\mathbf{p}_{\hat{y}_g}(\mathbf{x})$. Putting these two goals together, we propose to search for the probability estimate that solves

$$\begin{aligned} min_{\mathbf{p}(\mathbf{x})\in A_k} \sum_{\boldsymbol{\pi}^g\in\mathcal{G}} \sum_{c\in\mathcal{Y}\backslash\{\hat{y}_g\}} \quad & \eta\ (\boldsymbol{\pi}^g_c\,\mathbf{p}_c(\mathbf{x}) - \boldsymbol{\pi}^g_{\hat{y}_g}\mathbf{p}_{\hat{y}_g}(\mathbf{x}))_+ \ - \\ & - (\boldsymbol{\pi}^g_{\hat{y}_g}\mathbf{p}_{\hat{y}_g}(\mathbf{x}) - \boldsymbol{\pi}^g_c\mathbf{p}_c(\mathbf{x}))_+ \qquad (7) \\ \textit{subject to} \quad & \mathbf{p}_c(\mathbf{x}) \geq\ \epsilon \qquad \forall c \in \mathcal{Y} \qquad (8) \end{aligned}$$

where the constraint (8) enforces that $\mathbf{p}(\mathbf{x})$ is always strictly positive, η is an hyper-parameter that controls the trade-off between the desirable and undesirable deviations, and ϵ is a small positive constant.

The optimization of l_1-norm measures similar to the one used in (7)-(8) has been widely studied in the Operations Research literature on supervised classification (see Duarte Silva (2017)), where it has been shown that the resulting problems can be solved by straightforward linear programming models. In the current context, one important advantage of this method is the fact that it does not require $\boldsymbol{\pi}^g$ to be strictly uniformly distributed over A_k, which allows for alternative ways of defining representative $\mathcal{G}$ sets with a considerable smaller number of grid points. We use one such alternative, where we look at one component of $\boldsymbol{\pi}$, say $\boldsymbol{\pi}_c$, at the time, and ensure that a resulting set of $\boldsymbol{\pi}^g$ specifications gives an adequate representation of the $(0, 1)$ interval, while the remaining components of $\boldsymbol{\pi}$ are assigned at random. The representation of the $(0, 1)$ line is satisfied by setting $\boldsymbol{\pi}_c$ in turn to each of the $d_{\boldsymbol{\pi}}^{-1}$-1 uniformly distributed points of the $(0, 1)$ line, with a distance of $d_{\boldsymbol{\pi}}$ between each point. The resulting grid size equals $\#\mathcal{G} = k\, d_{\boldsymbol{\pi}}^{-1} - k$, a number that only increases linearly with k.

4 Learning Algorithms

The state of art algorithms for training SVMs are based on decomposition strategies to solve dual formulations of the associated optimization problems. In the case of 2-class problems, a standard reference is Platt's Sequential Minimal Optimization (SMO) algorithm (Platt, 1999) which decomposes a large convex quadratic optimization problem into a sequence of two-dimensional quadratic problems that can be solved analytically. This algorithm was adapted by Dogan et al. (2011) to solve the most common k-class SVMs, including those based on the unweighted LLW loss. It is straightforward to show that this approach also

applies to SVMs using the weighted LLW loss. In particular, the dual of optimization problem (3) - (4) with loss (5) can be expressed as the following as a convex quadratic optimization problem with box constrains.

$$max_{\boldsymbol{\alpha}\in\mathbb{R}^{n\times k}} \quad \frac{1}{k-1}\sum_{i\in\mathcal{T}}\ \sum_{c\in\mathcal{Y}\backslash\{y_i\}}\boldsymbol{\alpha}_{ic} - \\ -\frac{1}{2\lambda}\ \sum_{i,i'\in\mathcal{T}} K(\mathbf{x}_i,\mathbf{x}_{i'})\sum_{c,c'\in\mathcal{Y}}\ (\delta_{cc'}-\tfrac{1}{k})\ \boldsymbol{\alpha}_{ic}\ \boldsymbol{\alpha}_{i'c'} \tag{9}$$

$$subject\ to \qquad 0\ \leq\ \boldsymbol{\alpha}_{ic} \leq\ \tfrac{\boldsymbol{\pi}_{y_i}}{n} \qquad i\in\mathcal{T},\ c\in\mathcal{Y}\backslash\{y_i\} \tag{10}$$

$$\boldsymbol{\alpha}_{iy_i}\ =\ 0 \qquad i\in\mathcal{T} \tag{11}$$

where $\delta_{cc'} = I(c=c')$ is the kronecker delta.

Problem (9)-(11) can be solved efficiently by the following algorithm: (i) initialize the $\boldsymbol{\alpha}$ vectors at $\mathbf{0}$; (ii) choose the pair of $\boldsymbol{\alpha}$ components, $\boldsymbol{\alpha}_i$ and $\boldsymbol{\alpha}_{i'}$, that lead to the maximal increase in (9) subject to (10)-(11); (iii) find the analytical solution of problem (9)-(11) restricted to $\boldsymbol{\alpha}_i$ and $\boldsymbol{\alpha}_{i'}$; (iv) repeat (ii) and (iii) until convergence.

The details can be found in Dogan et al. (2011), while in Dogan et al. (2016) it is argued that, under reasonable assumptions, the asymptotic time complexity of this algorithm equals the one required to solve Crammer and Singer (2001) (CS) SVM. We note that the popularity of the CS SVM is mostly due to the fact that this algorithm is believed to be the fastest to train amongst all *all-in-one* multiclass SVMs.

5 Simulation Experiments

In this section we illustrate the performance of this proposal, comparing it with seven alternative methods for 4 simulation scenarios.

The methods under comparison are: (i) three model based statistical methods, namely Multinomial Logistic Regression (MLR), Multiple Linear Discriminant Analyis (MLDA) and Multinomial Generalized Additive Models (MGAM) (Yee and Wild (1996)); (ii) three SVM based metods, namely our Probabilistic Vector Machines (PVM) proposal, and the WZW (Wu et al., 2010) and XW (Xu and Wang, 2013) pairwise methods. (iii) two standard machine learning methods, namely classification trees (TREE) (Breiman et al., 1984) and Random Forests (RF) (Breiman, 2001).

We estimated the MLR and MLDA models using, respectively, the *multinom* and the *lda* functions of the *nnet* and *MASS* R packages (Venables and Ripley (2002)). In the MGAM models we employed cubic splines for all continuous attributes, linear functions for the discrete attributes, and relied on the *vgam* function of the *VGAM* package (Yee (2010)), with all remaining arguments at their default values. In the SVM based methods we used R code gently ceded by Xu and Wang, for XW method, and our own implementations for the PVM and WZW methods. In these three methods we always employed the Gaussian kernel, $K(\mathbf{x}_i,\mathbf{x}_{i'}) = e^{||\mathbf{x}_i,-\mathbf{x}_{i'}||_2^2/\sigma^2}$, with the hyperparameter σ^2 chosen as the median between all pairwise distances $||\mathbf{x}_i,-\mathbf{x}_{i'}||_2^2 \quad i,i'\in\mathcal{T} \quad i\neq i'$, in the training sample (see Caputo et al. (2002)). The regularization parameter λ in (3), was found by a two step search that tried

to minimize the log-likelihood, $\ln L = \sum_i \ln \hat{\mathbf{p}}_{y_i}(\mathbf{x}_i)$, in an independently generated tuning set with the same size as the training set. In the first step we searched for the λ_0 value that minimizes $\ln L$ over the set $\{2^{5j} | \, j = -3, -2, \ldots, 3\}$, and in the second step we refined the search by looking for the the minimizer of $\ln L$ over $\{2^j \lambda_0 | \, j = -2, -1, 0, 1, 2\}$. For the PVM method, the grid step size was always set at $d_{\boldsymbol{\pi}} = 0.2/\sqrt{n}$, the ϵ constant in (8) to $\epsilon = d_{\boldsymbol{\pi}}/2$, and the η hyper-parameter at $\eta = 15$. For the Tree and Random Forest methods we relied on the *TREE* and *RF* functions of the *rpart* (Therneau and Atkinson, 2018) and *randomForest* (Liaw and Wiener, 2002) R packages, with all arguments set at their default values.

We considered four different experiments with different data conditions. The first two experiments use setups initially considered in Wu et al. (2010). Both these experiments are 3-class problems, and the first one illustrates conditions in which the assumptions of MLR are satisfied, while in the second one the data was generated by highly non-linear functions that lead to a gross violation of these assumptions. The third and fourth experiments use setups initially considered in Xu and Wang (2013), in which the data was generated by heavy-tailed distribtions that also violate MLR assumptions. These two experiments consider, respectivelly a 5-class (3rd experiment) and a 10-class (4th experiment) problem. The details of the data generation are described in the paragraphs below.

Experiment 1. The first experiment uses the data conditions described in Example 1 of Wu et al. (2010), namely training samples of 400 observations with the Y class labels generated uniformly from $\mathcal{Y} = \{1, 2, 3\}$, and 2-dimensional predictors generated conditionally on Y, from a Gaussian distribution with mean vector $\boldsymbol{\mu}(y) = [cos(2y\pi/3), sin(2y\pi/3)]^T$ and covariance matrix $\Sigma = 0.7^2 \, \mathbf{I}_2$, with $\mathbf{I}_2$ being the 2-dimensional identity matrix.

Experiment 2. The second experiment uses the data conditions described in Example 3 of Wu et al. (2010), namely training samples of 600 observations with 2-dimensional predictors generated uniformly over the disk $\{\mathbf{x} : x_1^2 + x_2^2 \leq 100\}$, and class probabilities generated conditionally on $\mathbf{x}$ from $p_c(\mathbf{x}) = \exp(g_c(\mathbf{x}))/\sum_{c'=1}^{3} \exp(g_{c'}(\mathbf{x}))$, $c \in \mathcal{Y} = \{1, 2, 3\}$, where $g_1(\mathbf{x}) = \Phi^{-1}(T_2(-5\mathbf{x}_1\sqrt{3} + 5\mathbf{x}_2))$, $g_2(\mathbf{x}) = \Phi^{-1}(T_2(-5\mathbf{x}_1\sqrt{3} - 5\mathbf{x}_2))$, $g_3(\mathbf{x}) = 0$, and $\Phi(.)$ $T_2(.)$ denote the univariate cumulative standard normal and student t with 2 degrees of freedom (t_2) distributions.

Experiments 3 and 4. The third and fourth experiments use the data conditions described in Examples 1 and 2 of Xu and Wang (2013), namely training samples of 400 observations with the Y class labels generated uniformly from $\mathcal{Y} = \{1, 2, \ldots, k\}$, and 2-dimensional predictors generated conditionally on Y, from a t_2 distribution with mean vector $\boldsymbol{\mu}(y) = [cos(2y\pi/k), sin(2y\pi/k)]^T$ and covariance matrix $\Sigma = diag(1, 2)$. In experiment 3, $k = 5$ while in experiment 4, $k = 10$.

In all fourth experiments, we trained the eight estimation methods on 50 different, independently generated, training samples, and evaluated them by computing the l_2 norm error, $l_2\, err = \frac{1}{\#\mathcal{E}} \sum_{i \in \mathcal{E}} \sum_{c \in \mathcal{Y}} (\hat{\mathbf{p}}_c(\mathbf{x}_i) - \mathbf{p}_c(\mathbf{x}_i))^2$, and Empirical Generalized Kullback-Leiber (EGKL) measure, $EGKL = \frac{1}{\#\mathcal{E}} \sum_{i \in \mathcal{E}} \sum_{c \in \mathcal{Y}} \mathbf{p}_c(\mathbf{x}_i) \ln \frac{\mathbf{p}_c(\mathbf{x}_i)}{\hat{\mathbf{p}}_c(\mathbf{x}_i)}$, in an independently generated validation data set ($\mathcal{E}$) with 1000 examples.

Tables 1 through 4 present the means, mean standard errors, and medians of the l_2-norm and EGKL errors for all simulation experiments. Tables 1 and 2 also show, under the WZL acrynom, the reported average of these measures (see Wu et al. (2010)) for the

Table 1: Error rates for Experiment 1

	MLDA	MLR	MGAM	PVM	WZW	XW	TREE	RF	WZL
l2err									
mean	0.29	0.36	0.53	0.57	1.62	6.26	8.23	5.18	0.90
(stderr)	(0.03)	(0.03)	(0.04)	(0.04)	(0.09)	(0.13)	(0.15)	(0.11)	–
median	0.21	0.30	0.40	0.52	1.59	6.26	8.26	5.13	–
EGKL									
mean	0.59	0.77	1.12	1.53	3.47	∞	∞	∞	2.56
(stderr)	(0.05)	(0.07)	(0.09)	(0.09)	(0.15)	–	–	–	–
median	0.46	0.64	0.91	1.44	3.38	∞	∞	∞	–

Table 2: Error rates for Experiment 2

	MLDA	MLR	MGAM	PVM	WZW	XW	TREE	RF	WZL
l2err									
mean	6.90	6.62	5.21	2.50	5.01	9.44	10.13	5.26	4.47
(stderr)	(0.05)	(0.03)	(0.06)	(0.09)	(0.31)	(0.22)	(0.27)	(0.08)	–
median	6.81	6.58	5.19	2.38	4.71	9.03	9.75	5.35	–
EGKL									
mean	12.59	12.33	10.69	5.97	8.49	∞	∞	∞	11.79
(stderr)	(0.06)	(0.06)	(0.07)	(0.11)	(0.40)	–	–	–	–
median	12.51	12.18	10.59	5.81	7.70	∞	∞	∞	–

original Wu, Zhang and Liu proposal. We note that, unlike in our experiments, Wu et al. (2010) did not use an universal kernel but a linear one instead, taking advantage of the known form of the optimal classification boundaries for these conditions. However, in real data problem the true form of these boundaries is always unknown.

For some combinations of experiments, replications, and classes, the XL, TREE or RF methods estimate class probabilities exactly by 0, which leads to an infinite value of the EGKL measure. This problem does not occur for the l_2-norm error measure, nor for the remaining three methods, including the PVM which always enforce strictly positive probability estimates. Nevertheless, we tend to prefer the EGKL loss as an global measure of estimation performance because, among other reasons, unlike the l_2 norm error, it does not treat equal absolute errors as equally important, regardless of being associated with probabilities close to 0.5 or to the 0 and 1 boundaries of their domain. However, we note that, in these experiments, when results for the EGKL loss are available the resulting rankings of the six estimation methods tends to agree with the rankings given by the l_2 error. In Experiment 4 (10-classes with heavy-tailed distributions) we were not able to fit MGAM models in many replications because of numerical difficulties that were not resolved by the usual remedies of trying different starting conditions and data scalings. Therefore, we not present results for the MGAM method under this condition.

Overall these results confirmed previous findings in the literature, gave new evidence

Table 3: Error rates for Experiment 3

	MLDA	MLR	MGAM	PVM	WZW	XW	TREE	RF
l2err								
mean	5.60	5.01	3.59	4.17	2.47	2.82	4.73	13.19
(stderr)	(0.10)	(0.10)	(0.07)	(0.08)	(0.05)	(0.04)	(0.17)	(0.11)
median	5.56	4.91	3.61	4.13	2.39	2.81	4.57	13.23
EGKL								
mean	15.30	15.80	12.17	11.28	6.00	∞	∞	∞
(stderr)	(0.15)	(0.20)	(0.28)	(0.18)	(0.12)	–	–	–
median	15.08	15.66	11.53	11.26	5.81	6.36	∞	∞

Table 4: Error rates for Experiment 4

	MLDA	MLR	MGAM	PVM	WZW	XW	TREE	RF
l2err								
mean	3.34	3.09	–	2.63	1.81	1.81	3.93	14.89
(stderr)	(0.05)	(0.05)	–	(0.06)	(0.03)	(0.03)	(0.13)	(0.10)
median	3.34	3.07	–	2.53	1.81	1.78	3.82	14.81
EGKL								
mean	17.33	18.47	–	11.83	8.64	8.03	∞	∞
(stderr)	(0.20)	(0.29)	–	(0.19)	(0.16)	(0.15)	–	–
median	17.36	18.14	–	11.42	8.54	7.76	∞	∞

on the competitivety of the SVM aproach to probability estimation, and demonstrated the utility of our proposal. In particular, the MLR and MLDA methods gave the best results when the assumptions of MLR were met (Experiment 1), but when they were grossly violated one of the SVM based methods always performed the best. The MGAM performed worse than MLR and MLDA under MLR assumptions, and better than these two methods otherwise. However, in the conditions where MGAM beat the two classical statistical methods, it was always inferior to at least one of the SVM based methods. On the other hand, the performance of the Tree and Random Forest methods was disappointing, suggesting that in spite of their merits for pure supervised classification problems, they are not as reliable at providing estimates of class probabilities.

Among the three SVM methods we did not find a clear overall winner. In these experiments, the pairwise *one-against-all* methods performed the best for the conditions characterized by heavy-tailed distributions (Experiments 3 and 4), while the *all-in-one* PVM gave the best results for the non-lineary transformed data (Experiment 2). Furthermore, both the PVM and MGAM methods seemed relatively stable across different data conditions, coming often as second best, when they were not the ideal methods and, in particular, performing better than the pairwise SVM methods, when the assumptions of MLR are satisfied (Experiment 1), and better than MLR and MLDA when the data has severe outliers (Experiments 3 and 4). Among the parwise *one-against-all* SVM methods

the WZW performed better than the XW, confirming previous findings reported in Wang et al. (2019).

Remarkably, in the experiments for which results for the original *all-in-one* Wu et al. (2010) proposal were available, those results were improved by our PVM proposal, even though, contrary to Wu et al. (2010), we relied on agnostic universal kernel, and used only 297 (Experiment 1) and 364 (Experiment 2) grid points, while Wu et al. used a total of 1326 grid points in each of these two experiments. We believe that this surprising result might be explained by the fact the recovery of probabilities by optimizing (7) uses more information, than the matching of observed with expected frequencies employed in the original *all-in-one* SVM method. Finally the comparison of the results of experiments 3 and 4 suggests that the number of different classes might not a strong influence on the relative standing of alternative estimation methods. Naturally, additional studies are necessary to verify if these results hold for other data conditions.

6 Conclusions

Kernel based methods are an important set of tools for any modern statistician. However, most kernel methods focus on pure prediction problems, and do not pay enough attention to the related, and critical, problem of providing reliable confidence measures for these predictions. In the particular case of kernel based classification SVMs, this problem has being tackled by taking advantage of information provided by sequences of weighted SVMs. However, up to now, for a moderate or large number of different classes this approach was only computationally feasible based on pairwise pairwise *one-against-all* strategies, and not by a theoretically more sound *all-in-one* approach.

In this paper we have filled this gap, and developed a computationally efficient estimation method based on sequences of weighted SVMs that consider all k classes, simultaneously. Furthermore, we have provided further statistical evidence that SVM based probability estimators are among the most reliable distribution free estimators for class probabilities. In line with similar results for the pure classification problem, we have found that no particular method of extending 2-class to general k-class SVM methodologies dominates the alternatives, and different data conditions may favor different approaches.

References

Breiman, L. (2001). Random forests. *Machine Learning 45*(1), 5–32.

Breiman, L., J. Friedman, R. Olshen, and C. Stone (1984). *Classification and Regression Trees*. Wadsworth Publishing Company.

Caputo, B., K. Sim, F. Furesjo, and A. Smola (2002). Appearance-based object recognition using svms: which kernel should i use? In *Proc of NIPS workshop on Statistical methods for computational experiments in visual processing and computer vision, Whistler*.

Crammer, C. and Y. Singer (2001). On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research 2*, 265–292.

Cristianini, N. and J. Shawe-Taylor (2000). *An introduction to Support Vector Machines*. Cambridge University Press.

Dogan, U., T. Glasmachers, and C. Igel (2011). Fast training of multi-class support vector machines. Technical report, Department of Computer Science, University of Copenhagen.

Dogan, U., T. Glasmachers, and C. Igel (2016). A unified view on multi-class support vector classification. *Journal of Machine Learning Research 17*(45), 1–32.

Duarte Silva, A. P. (2017). Optimization approaches to supervised classification. *European Journal of Operational Research 261*(2), 772–788.

Kimeldorf, G. and G. Wahba (1971). Somes results on tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications 33*, 82–95.

Lee, Y., Y. Lin, and G. Wahba (2004). Multicategory support vector machines: Theory and application to the classification of microarray data and satellite radiance data. *Journal of the American Statistical Association 99*(465), 67–81.

Liaw, A. and M. Wiener (2002). Classification and regression by randomforest. *R News 2*(3), 18–22.

Lin, Y. (2002). Support vector machines and the bayes rule in classification. *Data Mining and Knowledge Discovery 6*(3), 988–994.

Lin, Y., L. Y., and G. Wahba (2002). Support vector machines for classification in non-standard situations. *Machine Learning 46*, 191–202.

Platt, J. (1999). Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, J. Burges, C, and A. Somola (Eds.), *Advances in kernel methods—support vector learning*, pp. 185–208. MIT Press.

Poggio, T., S. Mukherjee, R. Rifkin, A. Raklin, and A. Verri (2002). B. In J. Winkler and M. Niranjan (Eds.), *Uncertainty in Geometric Computations*. Springer US.

Therneau, T. and B. Atkinson (2018). *rpart: Recursive Partitioning and Regression Trees*. R package version 4.1-13.

Venables, W. N. and B. D. Ripley (2002). *Modern Applied Statistics with S* (Fourth ed.). New York: Springer.

Wahba, G. (1998). Support vector machines, reproducing kernel hilbert spaces, and randomized gacv. In B. Schölkopf, C. Burges, and A. Somola (Eds.), *Advances in Kernel Methods: Support Vector Learning*, pp. 69–87. MIT Press.

Wang, J., X. Shen, and Y. Liu (2008). Probability estimation for large-margin classifiers. *Biometrika 95*(1), 149–167.

Wang, X., H. Zhang, and Y. Wu (2019). Multiclass probability estimation with support vector machines. *Journal of Computational and Graphical Statistics 28*(3), 586–595.

Wu, Y., H. Zhang, and Y. Liu (2010). Robust model-free multiclass probavility estimation. *Journal of the American Statistical Association 105*(489), 424–436.

Xu, T. and J. Wang (2013). An efficient model-free estimation of multiclass conditional probability. *Journal of Statistical Planning and Inference 143*, 2079–2088.

Yee, T. W. (2010). The vgam package for categorical data analysis. *Journal of Statistical Software 32*, 1–34.

Yee, T. W. and C. Wild (1996). Vector generalized additive models. *Journal of the Royal Statistical Society: Series B 58*(3), 481–493.