Electronic Journal of Statistics
Vol. 9 (2015) 2255–2292
ISSN: 1935-7524
DOI: 10.1214/15-EJS1073

A Fisher consistent multiclass loss function with variable margin on positive examples

Irene Rodriguez-Lujan

BioCircuits Institute, University of California, San Diego,
La Jolla, CA 92093-0328, USA

Machine Learning Group, Escuela Politécnica Superior,
Universidad Autónoma de Madrid,
28049 Madrid, Spain
e-mail: irene.rodriguez@uam.es

and

Ramon Huerta

Rady School of Management & BioCircuits Institute,
University of California, San Diego,
La Jolla, CA 92093-0328, USA
e-mail: rhuerta@ucsd.edu

Abstract: The concept of pointwise Fisher consistency (or classification calibration) states necessary and sufficient conditions to have Bayes consistency when a classifier minimizes a surrogate loss function instead of the 0-1 loss. We present a family of multiclass hinge loss functions defined by a continuous control parameter λ representing the margin of the positive points of a given class. The parameter λ allows shifting from classification uncalibrated to classification calibrated loss functions. Though previous results suggest that increasing the margin of positive points has positive effects on the classification model, other approaches have failed to give increasing weight to the positive examples without losing the classification calibration property. Our λ-based loss function can give unlimited weight to the positive examples without breaking the classification calibration property. Moreover, when embedding these loss functions into the Support Vector Machine's framework (λ-SVM), the parameter λ defines different regions for the Karush—Kuhn—Tucker conditions. A large margin on positive points also facilitates faster convergence of the Sequential Minimal Optimization algorithm, leading to lower training times than other classification calibrated methods. λ-SVM allows easy implementation, and its practical use in different datasets not only supports our theoretical analysis, but also provides good classification performance and fast training times.

MSC 2010 subject classifications: 68T10.
Keywords and phrases: Multiclass classification, classification calibration, Bayes consistency, Fisher consistency, hinge loss functions, Support Vector Machine.

Received November 2014.

Contents

1. Introduction

Many of the most used classification algorithms are based on the minimization of a surrogate convex loss function since the direct minimization of the 0-1 loss is computationally intractable. Some examples of these algorithms include Support Vector Machines (SVMs) [8, 10, 34], boosting [13, 7, 22], and logistic regression [14]. Conditions such as convexity, continuity, and differentiability of these surrogate loss functions are easy to analyze; however, the statistical implications of using these surrogate loss functions are not so evident [2]. The notion of classification calibration was initially defined by Bartlett et al. [2, 3] as a pointwise form of Fisher consistency for classification. It was shown to be a necessary and sufficient condition for a binary classifier to be Bayes consistent when the empirical risk Ψ of a surrogate loss function converges to the minimal possible Ψ-risk. Tewari and Bartlett [37, Theo. 2] extended this classification calibration concept to multiclass problems. However, the extension of binary loss functions to multiclass classification settings is non-trivial, leading to a large body of research to better understand classification calibration in multiclass scenarios [23, 26, 37, 41, 42, 43, 28, 40]. The main contribution of this work is the formulation of a pointwise Fisher consistent (classification calibrated) multiclass loss function that can give arbitrary high weight to the margin of the positive points, which is shown to beneficial in terms of classification accuracy and training times. Our loss function overcomes some limitations of previous approaches since (i) it allows overweighting the margin of positive points while maintaining classification calibration, (ii) it yields consistent classification accuracies with respect to the classification calibration domain, and (iii) it can be efficiently trained when embedded in the Support Vector Machine's framework (λ-SVM).

There exist two main strategies to extend binary learning algorithms to a multiclass setting. The first approach consists of formulating the multiclass clas-

sification problem as a combination of several binary classification tasks. It includes strategies such as one-versus-rest, one-versus-one, and pairwise coupling [1]. Though these strategies are easy to implement, having optimal solutions of those binary classifiers does not guarantee having a global optimal solution for the multiclass problem. Additionally, the multiclass loss function does not necessarily inherit the classification calibration properties of its binary counterpart [37, 26, 42]. For example, the hinge loss function commonly used in Support Vector Machines has shown to be classification calibrated for binary problems [25], but the one-versus-rest strategy may be inconsistent when there is no a dominating class [26]. The second approach is based on the formulation of multiclass surrogate loss functions to use the same global optimization procedure as in the binary case. Though several multiclass hinge loss functions can be found in the literature [39, 9, 23, 28, 19], only two of them have been shown to be classification calibrated for every multiclass problem: Lee et al.'s loss function [23] and Liu and Yuan's loss function[1] (reinforced multicategory hinge loss) [28]. However, these approaches present some limitations. On the one hand, Lee et al.'s loss function does not consider the slack of the positive points of a given class, so it overlooks valuable information for the classification algorithm as it will be shown in our experiments (Section 6). On the other hand, the reinforced multicategory hinge loss considers both the margin of the positive and negative points of a given class, but experimental results in [28] on two synthetic datasets show that best classification performances are obtained for values of γ that overweight the margin of the positive points, and make the loss function classification uncalibrated. As pointed out by the authors, this is a surprising result. However, it points out the importance of paying attention to the error of positive examples. Additionally, the reinforced multicategory hinge loss assigns a margin of $(L-1)$ to the positive points, with L the number of classes, which is justified as a natural choice to have sum-to-zero loss functions. Unfortunately, using this margin in the context of Support Vector Machines is not beneficial for the optimization algorithm as it represents a boundary between different Karush—Kuhn—Tucker (KKT) conditions as we show in Proposition 3 (Section 5). A more detailed comparison between the different multiclass loss functions proposed in the literature can be found in Section 3.1.

The key contributions of this paper are:

- Formulation of a new family of multiclass hinge loss functions with a single control parameter $\lambda \in \mathbb{R}$ that represents the margin of the positive points of a given class. Our family of loss functions takes into account both the error of the positive and negative points of a given class, and it allows us to freely overweight the error associated to the positive points without losing classification calibration. A property that was attempted by Liu and Yuan [28] and Huerta et al. [19], but was not fully completed.
- Characterization of the classification calibration domain of this family of hinge loss functions. We show that its classification calibration proper-

[1] Liu and Yuan's loss function is classification calibrated for certain values of a meta–parameter γ.

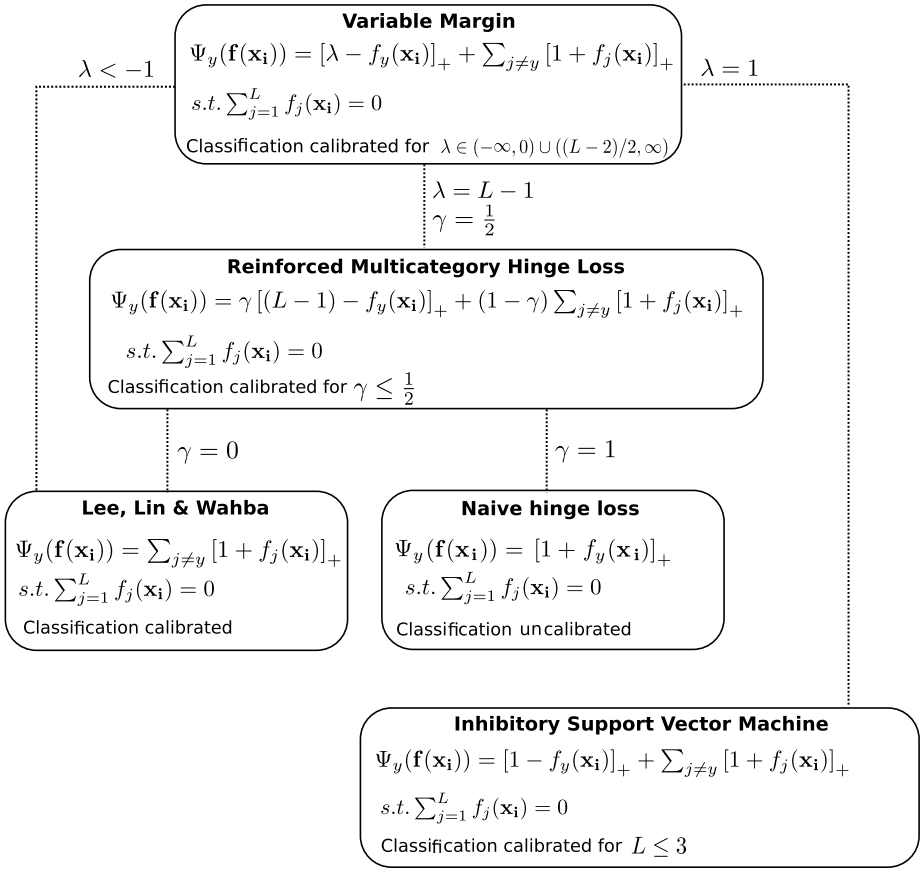


FIG 1. *Connections between different multiclass loss functions proposed in the literature and our family of variable margin loss functions.*

ties can be fully controlled by λ. This analysis reveals that our family of loss functions is classification calibrated provided that the margin of the positive points is larger or equal than $(L-2)/2$ with L the number of classes in the problem. This interesting property makes it possible to define a classification calibrated hinge loss function for every multiclass classification problem while overweighting the error of positive points. In other words, as long as one chooses $\lambda \notin [0, (L-2)/2]$ one can optimize the λ meta–parameter guaranteeing the classification calibration property.

- Formulation of a common framework that allows connecting the new family of loss functions with other classification calibrated multiclass hinge loss functions studied in the literature (Figure 1). Certain values of λ recover the hinge loss functions proposed by Lee et al. [23], Liu and Yuan [28] ($\gamma = 1/2$), and Huerta et al. [19], but appropriate values for λ allow overcoming some limitations of previous approaches. Lee et al.'s loss

function does not take into account the margin of the positive points of a given class, Huerta et al.'s loss function is not classification calibrated for classification problems with more than three classes, and the reinforced multicategory hinge loss cannot give large weight to positive classes without losing the classification calibration property.

- New multiclass SVM algorithm, named λ-SVM, formulated under the Inhibitory Support Vector Machine's formalism to guarantee sum-to-zero decision functions [19]. λ-SVM implementation is based on Sequential Minimal Optimization (SMO) [30, 20], the *de facto* standard in non-linear SVM training software [6]. Our C++ and Matlab implementations of λ-SVMs are provided as Supplementary Material.
- Theoretical and empirical analysis of the λ-SVM solutions (Karush–Kuhn–Tucker conditions) as a function of λ to show that choosing λ in $[(L-2)/2, L-1]$ slows down training times given the presence of different KKT conditions in the vicinity of λ.
- Empirical proof in real-world datasets of the advantage of (i) using classification calibrated loss functions in terms of classification accuracy, and (ii) overweighting the error of the positive points in terms of computational speed.

The paper is organized as follows. Section 2 defines classification calibration for multiclass problems and establishes its relationship with Bayes consistency. Section 3 presents our family of multiclass hinge loss functions with variable margin λ and characterizes the relationships between this family of loss functions and other multiclass losses existing in the literature. Section 4 formulates Theorem 2 that states the range of values of λ that makes our family of loss functions classification calibrated (classification calibration domain). Section 5 integrates our family of loss functions into the Support Vector Machines' framework to give rise to a new multiclass SVM model with variable margin λ (λ-SVM). Section 5 also analyzes λ-SVM solutions and KKT conditions to define a range of values for λ with good convergence properties. Section 6 provides results on four publicly available datasets in terms of classification accuracy and training times as a function of the margin of positive points λ. This section also provides a comparison with MSVMpack [21], a well-known package for multiclass Support Vector Machines. Finally, Section 7 formulates the conclusions derived from this work. A detailed proof of Theorem 2 can be found in Appendix A, and C++ and Matlab codes for λ-SVMs are provided as Supplementary Material [33].

2. Classification calibration for multiclass loss functions

Given an L-class classification problem ($L \geq 2$), the goal of a multiclass classification algorithm is to find a classifier $\phi : \mathcal{X} \to \mathcal{Y}$ such that the class label of every input pattern $\boldsymbol{x} \in \mathcal{X}$ is correctly estimated. In other words, our goal is to find a classifier ϕ such that $\phi(\boldsymbol{x}) = y$ for all $(\boldsymbol{x}, y) \in \mathcal{X} \times \mathcal{Y}$. However, this goal is fully achievable only when the classification problem is separable; oth-

erwise, the objective is to correctly classify the maximum number of samples. Without loss of generality, let's assume that $\boldsymbol{x_i} \in \mathcal{X} \subseteq \mathbb{R}^M$ is an input vector, and $y_i \in \mathcal{Y} = \{1, 2, \ldots, L\}$ is its class label. We are interested in minimizing the expected misclassification risk that is expressed as $R(\boldsymbol{f}) = \mathbb{E}_{\mathcal{XY}}\left[\mathbb{I}_{[\phi(\boldsymbol{x})\neq y]}\right]$, where $\mathbb{E}_{\mathcal{XY}}$ is the expectation with respect to the distribution of $\mathcal{X} \times \mathcal{Y}$, and $\mathbb{I}_A$ is the indicator function taking the value 1 if A is true, and 0 otherwise. The misclassification risk yields the probability that $\phi(\boldsymbol{x})$ provides an incorrect prediction for $\boldsymbol{x} \in \mathcal{X}$. The least possible $R(\boldsymbol{f})$, $\mathcal{R}^*$, defines the Bayes risk. This is the risk associated with the Bayes rule, which is the optimal classification strategy consisting of predicting the majority class for $\boldsymbol{x}$. The Bayes risk $\mathcal{R}^*$ is defined as $\mathcal{R}^* = \mathbb{E}_{\mathcal{X}}\left[1 - \max_{y\in\mathcal{Y}} P_y(\boldsymbol{x})\right]$, where $P_y(\boldsymbol{x}) = P(Y = y|\boldsymbol{X} = \boldsymbol{x})$ is the probability of class y given the point $\boldsymbol{x}$. However, in practice we do not have a whole representation of $\mathcal{X} \times \mathcal{Y}$, but we have a set of N training pairs $\{(\boldsymbol{x_i}, y_i)\}_{i=1}^N$. In this case, our goal is to minimize the empirical error on the training data, which is given by

$$\epsilon = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}_{[\phi(\boldsymbol{x_i})\neq y_i]} \ . \tag{1}$$

Therefore, the minimum possible value of the empirical error is zero, and it corresponds to the case when all the training points are correctly classified.

In what follows, we assume that the classifier ϕ is expressed as the combination of functions $\boldsymbol{f}$ and pred: $\phi(\boldsymbol{x}) = \text{pred}(\boldsymbol{f}(\boldsymbol{x}))$; that is, $\phi : \mathcal{X} \xrightarrow{\boldsymbol{f}} \mathbb{R}^L \xrightarrow{\text{pred}} \mathcal{Y}$. Here, $\boldsymbol{f}$ is an L-vector that belongs to $\mathcal{F}$, a class of vector functions $\boldsymbol{f} : \mathcal{X} \mapsto \mathbb{R}^L$. We refer to $\boldsymbol{f(x)} = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_L(\boldsymbol{x}))$ as the *decision function vector* or the *decision functions* of point $\boldsymbol{x}$. Each coordinate of $\boldsymbol{f}$ corresponds to the evaluation in $\boldsymbol{x}$ of the decision function associated with each class. The function pred discretizes $\boldsymbol{f}(\boldsymbol{x})$, and it is defined as $\text{pred}(\boldsymbol{x}) = \arg\max_j\{\boldsymbol{f}_j(\boldsymbol{x})\}$. Given that maximizing argument of $\boldsymbol{f}$ is invariant with respect to the addition of a constant to all entries in $\boldsymbol{f}$, it is advisable to impose a sum-to-zero constraint in order to simplify the analysis. Then, the class of vector functions $\mathcal{F}$ is defined as $\mathcal{F} = \left\{(f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_L(\boldsymbol{x})) \mid \sum_{j=1}^{L} f_j(\boldsymbol{x}) = 0 \ \forall \boldsymbol{x} \in \mathcal{X}\right\}$, and vectors $\boldsymbol{f} \in \mathcal{F}$ are known as *multicategory margin vectors* [44].

According to this mathematical framework, the classification function ϕ is unequivocally defined by the decision function $\boldsymbol{f}$ and, thus, the goal of the classifier is to minimize Eq. (1) with respect to $\boldsymbol{f}$. However, the direct minimization of Eq. (1) is known to be NP-hard [11, 4], so it is common to minimize instead *surrogate loss functions* $\Psi_y(\boldsymbol{f}(\boldsymbol{x}))$ that approximate the 0-1 loss function and have good computational guarantees such as differentiability and convexity. More precisely, $\Psi_y(\boldsymbol{f(x)})$ is defined as a continuous function from $\mathbb{R}^L$ to $\mathbb{R}^+$, and it can be understood as the loss associated with predicting the label of $\boldsymbol{x}$ using $\boldsymbol{f(x)}$ when the true label is y. Therefore, the expected risk associated with Ψ_y ($\boldsymbol{\Psi}$**-risk**) is defined as $R_\Psi(\boldsymbol{f}) = \mathbb{E}_{\mathcal{XY}}\left[\Psi_y(\boldsymbol{f}(\boldsymbol{x}))\right]$, and the *empirical* Ψ*-risk* corresponding to a training set of N pairs $\{(\boldsymbol{x_i}, y_i)\}_{i=1}^N$ is given by $\tilde{R}_\Psi(\boldsymbol{f}) = (1/N)\sum_{i=1}^{N} \Psi_{y_i}(\boldsymbol{f}(\boldsymbol{x}_i))$. Then, in practice, the classifier is inferred

from the decision function $\tilde{\boldsymbol{f}}_N$ that minimizes the empirical Ψ-risk as

$$\tilde{\boldsymbol{f}}_N = \arg\min_{\boldsymbol{f}\in\mathcal{F}} \tilde{R}_\Psi(\boldsymbol{f}) = \arg\min_{\boldsymbol{f}\in\mathcal{F}} \frac{1}{N}\sum_{i=1}^{N} \Psi_{y_i}(\boldsymbol{f}(\boldsymbol{x}_i)). \tag{2}$$

In this framework, Bartlett et al. formulate the concept of classification calibration as a necessary and sufficient condition to have Bayes consistency when the empirical risk of a binary loss function Ψ_y converges to the minimal possible Ψ-risk [3]. Tewari and Bartlett extend this classification calibration concept to multiclass problems [37, Theo. 2]. They show that multiclass classification calibration is equivalent to Bayes consistency assuming convergence of the empirical Ψ-risk to the minimal possible Ψ-risk, and they characterize classification calibration in terms of geometric properties of the loss function. Interestingly, Tewari and Bartlett also show that Bayes consistency of binary classifiers does not automatically imply Bayes consistency of the multiclass loss function and, thus, the classification calibration problem is more interesting in multiclass settings. The classification calibration definition derives from the minimization of the Ψ-risk. Writing the Ψ-risk as follows

$$\begin{aligned} R_\Psi(\boldsymbol{f}) &= \mathbb{E}_{\mathcal{X}\mathcal{Y}}\left[\Psi_y(\boldsymbol{f}(\boldsymbol{x}))\right] = \mathbb{E}_{\mathcal{X}}\left[\mathbb{E}_{\mathcal{Y}|\boldsymbol{x}}\left[\Psi_y(\boldsymbol{f}(\boldsymbol{x}))\right]\right] \\ &= \mathbb{E}_{\mathcal{X}}\left[\sum_{y\in\mathcal{Y}} P_y(\boldsymbol{x})\Psi_y(\boldsymbol{f}(\boldsymbol{x}))\right], \end{aligned} \tag{3}$$

the minimization of Eq. (3) is equivalent to the minimization of the inner conditional expectation for each $\boldsymbol{x}$. Initially proposed by Tewari and Bartlett [37, Definition 1], the classification calibration property can be defined as follows

Definition 1. *[37, 42] A surrogate function $\Psi_y(\boldsymbol{f}(\boldsymbol{x}))$ is said to be classification calibrated w.r.t. a margin vector $\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_L(\boldsymbol{x}))^T$ if for all $\{P_y(\boldsymbol{x})\}_{y\in\mathcal{Y}} \in \Delta_L$, where $\Delta_L = \{\mathbf{P}\in\mathbb{R}^L : P_j \geq 0\ \forall i = 1,\ldots,L$ and $\sum_{i=1}^{L} P_i = 1\}$ is the probability simplex in $\mathbb{R}^L$, the following conditions are satisfied:*

1. *The risk minimization problem $\hat{\boldsymbol{f}}(\boldsymbol{x}) = \arg\min_{\boldsymbol{f}(\boldsymbol{x})\in\mathcal{F}} \sum_{y\in\mathcal{Y}} P_y(\boldsymbol{x})\Psi_y(\boldsymbol{f}(\boldsymbol{x}))$ has a unique solution $\hat{\boldsymbol{f}}(\boldsymbol{x}) = (\hat{f}_1(\boldsymbol{x}), \hat{f}_2(\boldsymbol{x}), \ldots, \hat{f}_L(\boldsymbol{x}))^T$ for all $\boldsymbol{x}\in\mathcal{X}$; and*
2. *$\arg\max_{y\in\mathcal{Y}} \hat{f}_y(\boldsymbol{x}) = \arg\max_{y\in\mathcal{Y}} P_y(\boldsymbol{x})$ for all $\boldsymbol{x}\in\mathcal{X}$.*

Intuitively, Definition 1 states that the loss function Ψ_y is classification calibrated if its minimum allows recovering the index of the maximum probability for all $\boldsymbol{x}\in\mathcal{X}$.

Finally, it is worth noting that classification calibration is closely related to the concept of proper loss functions. However, classification calibration is a weaker condition as it only focuses on classification rather than estimating probabilities as in the case of properness [31]. For a more detailed explanation of the classification calibration framework and the consequent Bayes consistency properties, the reader is referred to [3, 37] and references therein.

3. Family of loss functions with variable margin λ

The analysis of Bayes consistency and classification calibration of several multiclass hinge loss functions has been extensively addressed in the literature [23, 26, 37, 41]. However, many existing multiclass loss functions are not classification calibrated. In order to provide a classification calibrated multiclass hinge loss function for every multiclass classification problem, we propose to use a family of loss functions regulated by a control parameter λ. Our set of loss functions for a data point $\boldsymbol{x_i}$ can be expressed as

$$\Psi_y(\boldsymbol{f}(\boldsymbol{x_i})) = [\lambda - f_y(\boldsymbol{x_i})]_+ + \sum_{j \neq y} [1 + f_j(\boldsymbol{x_i})]_+ \tag{4}$$

$$\text{s.t.} \quad \sum_{j=1}^{L} f_j(\boldsymbol{x_i}) = 0 \,, \tag{5}$$

where $[\rho]_+$ takes the value ρ for $\rho \geq 0$, and 0 otherwise. Intuitively, the above equation imposes variable margin λ for points in class y_i and margin 1 for points belonging to other classes. Eq. (4)–(5) are indeed a continuum of loss functions parametrized by λ. Finally, note that $\Psi_y(\boldsymbol{f}(\cdot))$ satisfies $\arg\min_j\{\Psi_j(\boldsymbol{f}(\boldsymbol{x_i}))\} = \arg\max_j\{f_j(\boldsymbol{x_i})\} = \text{pred}(\boldsymbol{x_i})$.

3.1. Connection with other multiclass loss functions

The connection between our family of loss functions and some other multiclass loss functions proposed in the literature is shown in Figure 1. Certain values of the parameter λ allow us to recover some existing classification calibrated loss functions. The equivalence to Lee et al.'s loss function [23] is obtained with $\lambda < -1$, but our family of loss functions is able to consider the slack of the positive points of a given class, which is beneficial to efficient learning (Section 6). The equivalence to the reinforced multicategory hinge loss [28] is obtained for $\gamma = 1/2$ and $\lambda = L - 1$, where L is the number of classes. In fact, the authors suggest to use the reinforced multicategory hinge loss with $\gamma = 1/2$ as a good trade-off between classification accuracy and classification calibration. However, the best performance is generally obtained in classification uncalibrated scenarios ($\gamma > 1/2$) in which the margin of the positive points dominates in the loss function. On the other hand, the reinforced multicategory hinge loss sets the margin of positive points λ equal to $(L - 1)$ to have sum-to-zero decision functions. Beyond the mathematical convenience, this decision restricts the classification calibration domain of the loss function to $\gamma \leq 1/2$. In Section 4, we show that our family of loss functions not only provides optimal decision functions different from those of the reinforced multicategory hinge loss, but it also allows us to have a classification calibrated loss function while giving arbitrarily high weight to the margin of the positive points of a given class. Furthermore, setting the

margin of positive points equal to $(L-1)$ has negative effects on the optimizer since three different KKT solutions are obtained for any interval containing $\lambda = L-1$ (Section 5). Our loss function also becomes equivalent to that of Inhibitory Support Vector Machines (ISVMs) proposed by Huerta et al. when $\lambda = 1$ [19]. Huerta et al. show that their loss function is classification uncalibrated for problems with more than three classes. This result matches with that obtained in Section 4. The introduction of the variable margin in our loss functions makes it possible to define a classification calibrated loss function for every classification problem regardless of its number of classes.

Besides the multiclass loss functions that can be treated as special cases of our multiclass loss function, other multicategory loss functions can be also found in the literature. Guermeur and Monfrini propose a new multiclass loss function with quadratic loss instead of hinge loss (MSVM2 loss function) [18]. As stated by Guermeur and Monfrini, the main advantage of using the 2-norm loss is that the training algorithm can be expressed, after an appropriate change of kernel, as the training algorithm of a hard margin machine. Guermeur and Monfrini established a generalized radius–margin bound on the leave–one–out error of the hard margin version of their loss function. This provides them with a differentiable objective function to perform model selection for the MSVM2 loss. However, hinge loss is usually preferred for classification tasks. Additionally, though Guermeur and Monfrini state that their MSVM2 loss function can be seen as a quadratic loss variant of the multiclass SVM of Lee et al. [23], the MSVM2 consistency properties are not discussed. In Section 6, we included a comparison in terms of classification accuracy and training times between the MSVM2 loss function implemented in the MSVMpack package [21] and our loss function.

Liu and Shen's multiclass loss function [27] is an extension of the binary ψ-learning originally proposed by Shen et al. [35]. ψ-learning is as another margin-based technique that replaces the convex SVM loss function by a non-convex ψ-loss function. Shen et al. show that ψ-learning can achieve good classification rates while maintaining the margin interpretation. They also show that their loss function converges to the Bayes decision rule. In contrast, our loss function extends the hinge loss function traditionally used in SVMs while ensuring consistency for certain values of λ. As an extension of traditional SVMs, the λ-SVM problem is convex and solvers commonly used for SVMs can be applied. However, these solvers are not suitable for the ψ-loss function; a method based on a difference convex (dc) decomposition is used instead to solve the multiclass ψ-learning optimization problem.

Finally, the L1MSVM approach is another multiclass Support Vector Machine model that is based on the L1-norm [38]. L1MSVM simultaneously performs feature selection and classification through an L1-norm penalized sparse representation. L1MSVM is formulated to use several loss functions that can be expressed in a unified fashion. Wang and Shen conduct a detailed analysis of L1MSVM considering Lee et al.'s loss function [23], which is known to be

classification calibrated. Unlike L1MSVM, in this work we embed our loss functions into the Inhibitory Support Vector Machine framework with an L2-norm regularization term.

4. Classification calibration domain

According to the framework described in Section 2, the analysis of classification calibration requires to minimize the inner conditional expectation for each $\boldsymbol{x}$ in Eq. (3). In what follows, we fix $\boldsymbol{x}$ and omit dependencies on $\boldsymbol{x}$ to simplify the notation. Replacing $\Psi_y(\boldsymbol{f})$ by our set of loss functions in Eq. (3) we obtain $\hat{\boldsymbol{f}} = \arg\min_{\boldsymbol{f}} \sum_{l=1}^{L} P_l \left([\lambda - f_l]_+ + \sum_{j \neq l} [1 + f_j]_+ \right)$. Equivalently,

$$\hat{\boldsymbol{f}} = \arg\min_{\boldsymbol{f}} \sum_{l=1}^{L} P_l [\lambda - f_l]_+ + (1 - P_l)[1 + f_l]_+ \,. \tag{6}$$

Now, we are ready to formulate the following theorem that characterizes the classification calibration domain of our family of loss functions.

Theorem 2. *Given a multiclass classification problem with L classes, the family of loss functions defined in Eq. (4)–(5) is classification calibrated for $\lambda \in (-\infty, 0) \cup ((L-2)/2, \infty)$.*

Proof. A detailed proof can be found in Appendix A. The sketch of the proof can be outlined as follows: for $\lambda \leq L - 1$, it is shown that the optimal decision functions are lower bounded by -1, while for $\lambda > L - 1$ the decision functions are upper bounded by λ. Taking into account these bounds together with the sum-to-zero constraint, the minimization problem in Eq. (6) is formulated as an optimization problem with equality and inequality constraints. Then, the relationships between decision functions and class probabilities, which allow us to determine the classification calibration properties of our loss functions, are stated by the Karush—Kuhn—Tucker (KKT) conditions [5]. □

According to Theorem 2, we can define classification calibrated multiclass hinge loss functions for any multiclass classification problem by means of the scalar parameter λ. Certain values of λ enable not only to have classification calibrated loss functions, but also, unlike the other classification calibrated loss function [23], to take into account the margin of positive points. Figure 2 shows the ratio of classification uncalibrated solutions obtained by Monte Carlo simulations for different values of the control parameter λ and the number of classes L. These results were obtained by counting the number of classification uncalibrated cases when minimizing the empirical Ψ-risk in Eq. (6) for 10,000 random probability simplex in $\mathbb{R}^L$. Monte Carlo simulations give evidence of the classification calibration domain presented in Theorem 2: $\lambda \in (-\infty, 0) \cup ((L-2)/2, +\infty)$.

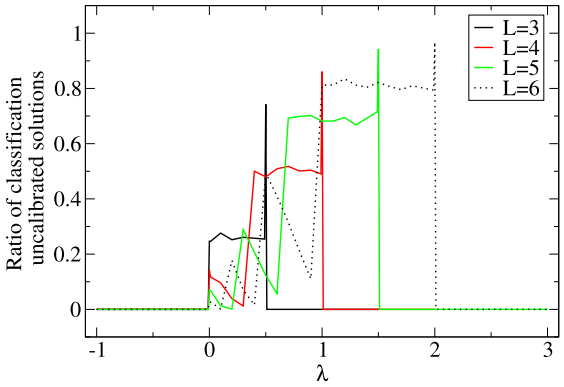


FIG 2. *Monte Carlo simulation results on the minimization of the empirical Ψ-risk associated with our multiclass loss functions with variable margin λ. Figure shows the ratio of classification uncalibrated cases as a function of the control parameter λ and for different number of classes L. The number of simulations was set to 10,000.*

5. Classification calibration for Support Vector Machines

Large-margin classifiers make tractable the minimization of the 0-1 loss by using convex surrogate loss functions. Examples of this approach are Support Vector Machines [8] and boosting [13]. The general formulation of a large-margin classification algorithm with regularization is $\min_{\boldsymbol{f}\in\mathcal{F}}\ (1/N)\sum_{i=1}^{N}\Psi_{y_i}(\boldsymbol{f}(\boldsymbol{x_i}))+\rho J(\boldsymbol{f})$, where $J(\boldsymbol{f})$ is a regularization term to penalize the model complexity, and ρ is the regularization parameter. Our proposed loss functions can be used in any standard regularized empirical risk minimizer. We used the Sequential Minimal Optimization (SMO) implementation of the Inhibitory Support Vector Machines (ISVMs) [19] since, as described further down in this section, they implicitly produce sum-to-zero decision functions for any example, while standard SVMs do not. For example, Lee et al.'s implementation of SVMs needs to explicitly add a sum-to-zero constraint not necessary in the ISVM implementation [23]. The best feature of the ISVM is the easiness of the implementation that allows a quick adaptation to any variable margin framework.

ISVM is an extension of SVM to provide a simple algorithm for multiclass classification by directly integrating the concept of inhibition into the SVM formalism. The objective of the inhibition mechanism behind the ISVM algorithm is to find a hyperplane associated with each class, $\{\boldsymbol{w}_j\}_{j=1}^{L}$, that exerts downward pressure on the rest hyperplanes while trying to maximize its generalization capability. ISVM decision function for class j evaluated in a data point $\boldsymbol{x_i}$ has the form

$$f_j(\boldsymbol{x_i}) \quad = \quad \langle \boldsymbol{w_j}, \Phi(\boldsymbol{x_i})\rangle - \mu\sum_{k=1}^{L}\langle \boldsymbol{w_k}, \Phi(\boldsymbol{x_i})\rangle\ , \tag{7}$$

where Φ is a mapping function from the original input space to a higher-dimensional space $\mathcal{V}$ (feature space) where the optimal hyperplane is calculated. The parameter μ is a scalar number that regulates the inhibitory term, which is the key difference with respect to standard SVMs. The optimal decision vector $\boldsymbol{f}$ is determined by following the standard SVMs' framework. The ISVM primal problem is expressed as

$$\min_{\boldsymbol{w}} \quad \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{C}{NL}\sum_{i=1}^{N}\sum_{j=1}^{L}\eta_{ij} \tag{8}$$

$$\text{s.t.} \quad \eta_{ij} \geq 0 \tag{9}$$

$$y_{ij} f_j(\boldsymbol{x_i}) - 1 + \eta_{ij} \geq 0, \tag{10}$$

where $\boldsymbol{w}$ is the concatenation of the hyperplanes of each class, $\boldsymbol{w} = [\boldsymbol{w_1}, \ldots, \boldsymbol{w_L}]$, $\{\eta_{ij}\}$ are the slack variables that provide room to handle the noisy data, and y_{ij} takes the value 1 if the pattern $\boldsymbol{x}_i$ belongs to class j (i.e., $y_i = j$) and -1 otherwise. Note that now the trade-off between the regularization term and the loss function is controlled by the cost parameter C instead of the regularization parameter ρ. To simplify the notation, in what follows we assume that the cost parameter C is already normalized by the number of training points (N) and the number of classes (L). Inhibitory Support Vector Machines use an input space formed by L concatenations of the original input space $\mathcal{X}$, and they use a feature space that is the product space $\mathcal{V}^L$. Then, an input vector $\boldsymbol{\chi}_i \in \mathbb{R}^{ML}$ is formed by L concatenations of the original training pattern $\boldsymbol{x}_i \in \mathbb{R}^M$. The corresponding nonlinear transformation $\Upsilon(\boldsymbol{\chi}) \in \mathcal{V}^L$ is defined as $\Upsilon(\boldsymbol{\chi}) = (\Phi(\boldsymbol{x}), \Phi(\boldsymbol{x}), \ldots, \Phi(\boldsymbol{x}))$ (L times), and $\Upsilon_j(\boldsymbol{\chi})$ is the composition of $\Upsilon(\boldsymbol{\chi})$ with the projection operator onto the j-th coordinate subspace corresponding to the j-th class; that is, $\Upsilon_j(\boldsymbol{\chi}) = (0, 0, \ldots, \Phi(\boldsymbol{x}), \ldots, 0)$ with all coordinates except the j-th equal to zero. The transformations Υ and Υ_j inherit many properties from the mapping function $\Phi(\boldsymbol{x})$. In particular,

$$\langle \Upsilon_j(\boldsymbol{\chi}_i), \Upsilon_{j'}(\boldsymbol{\chi}_{i'}) \rangle = \mathbb{I}_{[j=j']} \langle \Phi(\boldsymbol{x}_i), \Phi(\boldsymbol{x}_{i'}) \rangle, \tag{11}$$

$$\langle \Upsilon_j(\boldsymbol{\chi}_i), \Upsilon(\boldsymbol{\chi}_{i'}) \rangle = \langle \Phi(\boldsymbol{x}_i), \Phi(\boldsymbol{x}_{i'}) \rangle, \tag{12}$$

$$\langle \Upsilon(\boldsymbol{\chi}_i), \Upsilon(\boldsymbol{\chi}_{i'}) \rangle = L \, \langle \Phi(\boldsymbol{x}_i), \Phi(\boldsymbol{x}_{i'}) \rangle. \tag{13}$$

Huerta et al. show that the optimal value for μ is $\mu = 1/L$, which can be obtained directly from the minimization of the Lagrangian of Problem (8)–(10) [19]. They also show that, in that limit, ISVMs become a tight bound to probabilistic exponential models. The inhibition term, therefore, is the average over the evaluation of the hyperplanes of each class. Interestingly, μ is dependent on the number of classes of the problem, but independent of the training points themselves. This result is especially appealing when working with multiclass margin vectors since it yields sum-to-zero decision functions without imposing additional constraints in the optimization problem. ISVM automatically embodies all the zero-sum loss functions:

$$\begin{aligned}\sum_{j=1}^{L} f_j(\boldsymbol{x_i}) &= \sum_{j=1}^{L}\left\{\langle \boldsymbol{w_j}, \Phi(\boldsymbol{x_i})\rangle - \frac{1}{L}\sum_{k=1}^{L}\langle \boldsymbol{w_k}, \Phi(\boldsymbol{x_i})\rangle\right\} \\ &= \sum_{j=1}^{L}\langle \boldsymbol{w_j}, \Phi(\boldsymbol{x_i})\rangle - L\,\frac{1}{L}\sum_{k=1}^{L}\langle \boldsymbol{w_k}, \Phi(\boldsymbol{x_i})\rangle = 0\,.\end{aligned}$$

The loss function in Eq. (10) corresponds to $\lambda = 1$. Therefore, it is straightforward to integrate the family of loss functions presented in Eq. (4)–(5) into the ISVM's framework. It can be formulated as follows

$$\min_{\boldsymbol{w}} \quad \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N}\sum_{j=1}^{L}\eta_{ij} \tag{14}$$

$$\text{s.t.} \quad \eta_{ij} \geq 0 \tag{15}$$

$$-1 - (\lambda - 1)\frac{y_{ij}+1}{2} + f_j(\mathbf{x}_i)y_{ij} + \eta_{ij} \geq 0 \text{ for } \lambda \in \mathbb{R}\,. \tag{16}$$

To obtain the solution to Problem (14)–(16), we compute its Lagrangian as follows

$$\begin{aligned}\mathcal{L}(\boldsymbol{w}, \boldsymbol{\eta}, \mu, \boldsymbol{\zeta}, \boldsymbol{\alpha}) &= \left\{\frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N}\sum_{j=1}^{L}\eta_{ij} - \sum_{i=1}^{N}\sum_{j=1}^{L}\zeta_{ij}\eta_{ij}\right. \\ &\quad - \sum_{i=1}^{N}\sum_{j=1}^{L}\alpha_{ij}\Big(y_{ij}\left[\langle \boldsymbol{w}, \Upsilon_j(\boldsymbol{\chi}_i)\rangle - \mu\langle \boldsymbol{w}, \Upsilon(\boldsymbol{\chi}_i)\rangle\right] \\ &\quad \left. -1 - (\lambda-1)\frac{y_{ij}+1}{2} + \eta_{ij}\Big)\right\}\end{aligned} \tag{17}$$

where the Lagrange multipliers are $\alpha_{ij} \geq 0$ and $\zeta_{ij} \geq 0$. The decision function associated with the j-th class for a training point $\boldsymbol{x_i}$ (Eq. 7) is now expressed as $f_j(\boldsymbol{x}_i) = \langle \boldsymbol{w}, \Upsilon_j(\boldsymbol{\chi}_i)\rangle - \mu\langle \boldsymbol{w}, \Upsilon(\boldsymbol{\chi}_i)\rangle$. We calculate the partial derivatives of $\mathcal{L}$ with respect to the primal variables $\boldsymbol{w}$, $\boldsymbol{\eta}$, and μ to make them equal to zero. It leads to

$$\alpha_{ij} = C - \zeta_{ij} \tag{18}$$

$$\boldsymbol{w} = \sum_{i=1}^{N}\sum_{j=1}^{L}\alpha_{ij}y_{ij}\left[\Upsilon_j(\boldsymbol{\chi}_i) - \mu\Upsilon(\boldsymbol{\chi}_i)\right] \tag{19}$$

$$0 = \sum_{i=1}^{N}\sum_{j=1}^{L}\alpha_{ij}y_{ij}\langle \boldsymbol{w}, \Upsilon(\boldsymbol{\chi}_i)\rangle \tag{20}$$

Then, as in [19, Appendix B], replacing Eq. (19) in Eq. (20) yields the optimal μ as $\mu = 1/L$. Since the partial derivatives of the Lagrangian w.r.t. μ and $\boldsymbol{w}$ do not depend on λ, this reasoning is valid for any $\lambda \in \mathbb{R}$, and, thus, sum-to-zero decision functions are guaranteed for all $\lambda \in \mathbb{R}$. This property makes ISVM's framework advantageous for implementing multicategory margin vectors.

Now, we obtain the ISVM dual problem by applying Eq. (18)–(19) and Properties (11)–(13) to the Lagrangian in Eq. (17) with $\mu = 1/L$. It leads to the dual cost function W that has to be maximized with respect to the Lagrange multipliers, α_{ij},

$$\begin{aligned} \max_{\boldsymbol{\alpha}} W &= \sum_{i=1}^{N}\sum_{j=1}^{L}\alpha_{ij}l_{ij} - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{L}\sum_{i'=1}^{N}\sum_{j'=1}^{L}\alpha_{ij}y_{ij}\alpha_{i'j'}y_{i'j'}K_{i'i}\left[\mathbb{I}_{[j=j']} - \frac{1}{L}\right], \\ \text{s.t.} \quad & 0 \le \alpha_{ij} \le C, \end{aligned}$$

where $l_{ij} = 1 + (\lambda - 1)\frac{y_{ij}+1}{2}$ and $K_{ii'} = K(\mathbf{x}_i, \mathbf{x}_{i'}) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_{i'}) \rangle$. Now, the decision function for the j-th class can be written in terms of the Lagrange multipliers and the kernel function as

$$f_j(\boldsymbol{x}) = \sum_{i'=1}^{N}\sum_{j'=1}^{L}\alpha_{i'j'}y_{i'j'}K(\boldsymbol{x}_{i'}, \boldsymbol{x})\mathbb{I}_{[j=j']} - \frac{1}{L}\sum_{i'=1}^{N}\sum_{j'=1}^{L}\alpha_{i'j'}y_{i'j'}K(\boldsymbol{x}_{i'}, \boldsymbol{x}) \ .$$

We can simplify the evaluation by just computing

$$f_j(\boldsymbol{x}) = \sum_{i'=1}^{N}\alpha_{i'j}y_{i'j}K(\boldsymbol{x}_{i'}, \boldsymbol{x}) \tag{21}$$

since the remaining terms simply add the same constant to all the classes. The class of the test sample $\boldsymbol{x}$ is defined as $\arg\max_j f_j(\boldsymbol{x})$. Following the notation in [19], we change the double index notation α_{ij} for a new index k running from 1 to NL. Assuming lexicographical order in the α_{ij}s, the dual cost function W can be written as

$$\max_{\boldsymbol{\alpha}} W = \sum_{k=1}^{NL}\alpha_k l_k - \frac{1}{2}\sum_{k=1}^{NL}\sum_{k'=1}^{NL}\alpha_k y_k \alpha_{k'} y_{k'} G_{kk'} \tag{22}$$

$$\text{s.t.} \quad 0 \le \alpha_k \le C \text{ for all } k = 1, \ldots, NL, \tag{23}$$

where $G_{kk'} = K_{\lfloor((k-1)/L)+1\rfloor,\lfloor((k'-1)/L)+1\rfloor}\left[\mathbb{I}_{[(k \mod L)=(k' \mod L)]} - 1/L\right]$. Then, it is easy to see that the KKT conditions for the λ-SVM training problem are

$$\begin{aligned} V_k &\ge 0 \quad \text{for } \alpha_k = 0 \ , \\ V_k &= 0 \quad \text{for } 0 < \alpha_k < C \ , \\ V_k &\le 0 \quad \text{for } \alpha_k = C \ , \end{aligned}$$

where $V_k = y_k(f_k - l_k y_k) = y_k f_l - l_k = y_k f_k - (1 + (\lambda - 1)((y_k + 1)/2)$. Huerta et al. provide a very easy and simple implementation of ISVM with $\lambda = 1$ based on Sequential Minimal Optimization (SMO) [30] that can be easily translated to the variable margin setting with minimal changes in the computer program. As originally proposed by Platt [30], the resolution of the proximity to the KKT condition in the optimization algorithm is controlled by a tolerance parameter

$T > 0$ and a numerical resolution ϵ, which depends on the machine precision. Then, the fulfillment of the KKT conditions is formulated as follows

$$V_k \geq -T \qquad \text{for } \alpha_k < \epsilon \,, \tag{24}$$

$$-T < |V_k| < T \qquad \text{for } \epsilon < \alpha_k < C - \epsilon \,, \tag{25}$$

$$V_k < T \qquad \text{for } \alpha_k > C - \epsilon \,. \tag{26}$$

An interesting point of analysis is to determine the stability of the SMO optimization algorithm by taking into account the different regions of optimal decision functions defined by λ and summarized in Figure 3 (for more details, the reader is referred to Appendix A). Note that since SMO is used as optimization algorithm, λ-SVM optimal decision functions are defined as a function of the optimal Lagrange multipliers $\hat{\alpha}_j$s according to Eq. (21), which can be easily obtained by means of the equality $V_i = y_i(\hat{f}_i - l_i y_i)$. To illustrate the negative effect of having different KKT solutions in the proximity of λ in terms of computational cost, we measured the training times in the simplest case in which SMO is applied to single point. We set class probabilities to $P_1 = 0.375$, $P_2 = 0.34$, and $P_3 = 0.28$, we created $N = 264$ training points, and we set $T = 10^{-3}$, $\epsilon = 10^{-6}$, and $C = 10^6$. The resulting training times for different λ-regions are shown in Figure 3.

The different KKT conditions derived from Figure 3 together with the KKT numerical conditions in Eq. (24)–(26) allow us to formulate the following proposition.

Proposition 3. *The optimal solution for the SVMs with variable margin has three possible KKT solutions in the domain* $\lambda \in (L-1-T, L-1+T)$ *for any resolution proximity* $T > 0$*:*

i) $\hat{V}_1 = 0$ *and* $0 < \hat{\alpha}_1 < C$ *;* $\hat{V}_2 \leq 0$ *and* $\hat{\alpha}_2 = C$*;* $\{\hat{V}_j\}_{j=3}^{L} = 0$ *and* $0 < \{\hat{\alpha}_j\}_{j=3}^{L} < C$*.*

ii) $\hat{V}_1 \geq 0$ *and* $\hat{\alpha}_1 = 0$ *;* $\{\hat{V}_j\}_{j=2}^{L} = 0$ *and* $0 < \{\hat{\alpha}_j\}_{j=2}^{L} < C$*.*

iii) $\hat{V}_1 = 0$ *and* $0 < \hat{\alpha}_1 < C$ *;* $\{\hat{V}_j\}_{j=2}^{L-1} = 0$ *and* $0 < \{\hat{\alpha}_j\}_{j=2}^{L-1} < C$ *;* $\hat{V}_L \geq 0$ *and* $\hat{\alpha}_L = 0$*.*

The resolution proximity T in the KKT conditions (Eq. (24)–(26)) implies to solve the dual problem for an effective margin $\lambda_{\text{eff}} \in (\lambda - T, \lambda + T)$. That is why the SMO algorithm shows a slow convergence for λ in the proximity of the boundary between different KKT solutions. It should also be noted that there may exist other points subject to KKT variations inside the same classification calibration region since the solutions for $\lambda \in (-1, 0)$ and $\lambda \in ((L-2)/2, L-1)$ depend on the class probability distribution, which in turn depends on λ. This is not the case for $\lambda > (L-1)$ since the transition between the two possible solutions is only defined by the class probabilities; that is, the KKT conditions are constant given any $\lambda > (L-1)$ and any point. It means that the margin $\lambda = (L-1)$ imposed by the reinforced multicategory hinge loss [28], though guaranteeing classification calibration, may slow down the convergence of the the SMO algorithm given that the optimizer is searching across different KKT

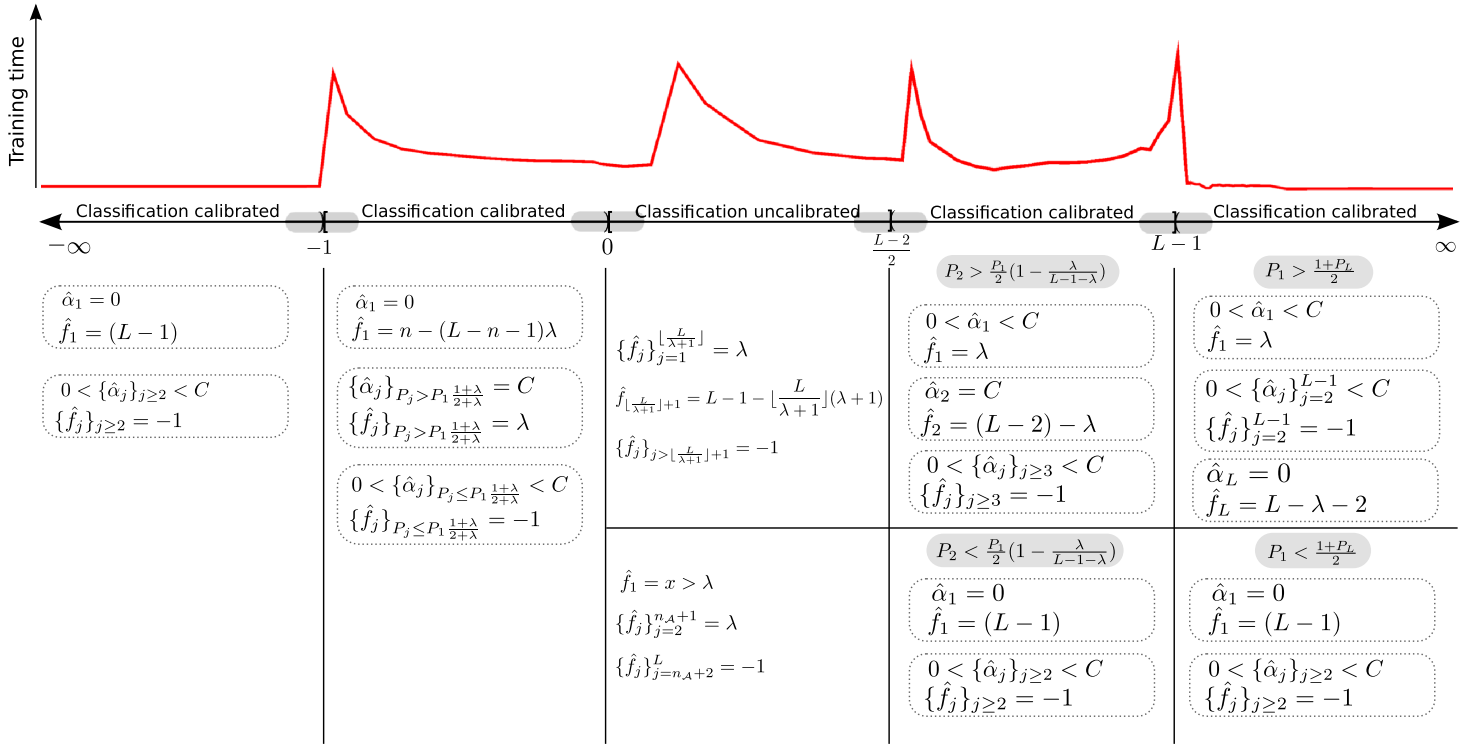


FIG 3. *Training times of the SMO algorithm trained with a single point* $\boldsymbol{x}$ *as a function of the* λ*-regions defined by the classification calibration domain and the optimal decision functions* $\hat{f}_j(x) = \sum_{i'=1}^{N} \hat{\alpha}_{i'j} y_{i'j} K(\boldsymbol{x}_{i'}, \boldsymbol{x})$.

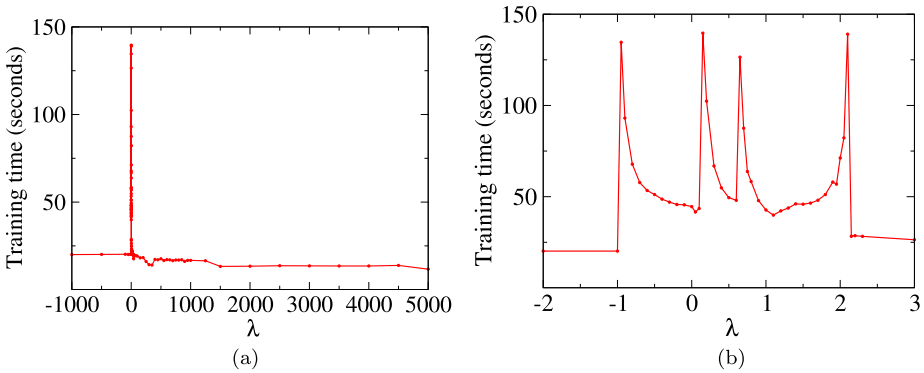


FIG 4. *Training times of SMO as a function of the margin of positive points* λ *for a single point with class probabilities* $P_1 = 0.375$, $P_2 = 0.34$, *and* $P_3 = 0.28$. *Figure 4a shows the training times for* $\lambda \in [-1000, 5000]$. *Figure 4b provides detailed training times for* $\lambda \in [-2, 3]$.

regions. Proposition 3 and Figure 3 suggest that the margin of the positive points should be chosen somewhere in $(-\infty, -1) \cup (L-1, +\infty)$ with enough space with respect to the tolerance T to not incur in KKT instability problems. The case $\lambda \in (-\infty, -1)$ corresponds to Lee et al.'s loss function [23]. However, values of $\lambda \gg (L-1)$ provide the best training times as shown in Figure 4. The advantage of strongly considering the margin of the positive points in terms of training times will be also confirmed in the following section.

6. Experimental evaluation

The aim of this section is to conduct an empirical evaluation in terms of classification accuracy and training times of the λ-SVM model introduced in Section 5. We used four real-world datasets from the UCI data repository [24] described in Table 1. Some of these datasets involve real applications such as classification on gas sensor arrays [32]. Base error was obtained by predicting the majority class in each dataset. In Covtype dataset, a random selection of 50,000 points was performed. In the Abalone dataset, age bands were obtained dividing age by 5. These datasets were chosen because they have a large number of training points compared to the dimensionality. This favors large values of the cost parameter C, which in turn can reveal differences between classification calibrated and uncalibrated loss functions since the regularization term almost vanishes. Otherwise, under appropriate regularization, all SVM models are classification calibrated [36].

We generated five different partitions of each experiment. The first 90% of samples was selected as the training set, and the remaining 10% of samples constituted the test set. The training samples were used to build the λ-SVM model. We used a function with compact support as a kernel. Kernels with non-zero tails such as the Gaussian kernel can be detrimental in scenarios with finite number of points and C very large since points that are significantly far from

TABLE 1
Datasets used to evaluate λ-SVMs

Dataset	*no. examples*	*no. classes*	*no. attributes*	*Base error*
Sensor	13,910	6	128	78.37%
Pendigit	10,992	10	16	89.59%
Covtype	50,000	7	54	51.24%
Abalone	4,177	6	8	51.59%

the point of interest can still have a notable contribution, especially when there are not enough points in the neighborhood of the point of interest. Specifically, we used the compactly supported kernel proposed and analyzed in [15, 16, 41]. The compactly supported kernel can be written as follows

$$K_{D,\nu}(\boldsymbol{x},\boldsymbol{x'}) \quad = \quad \phi_{D,\nu}(\boldsymbol{x},\boldsymbol{x'})K(\boldsymbol{x},\boldsymbol{x'}) \ ,$$

and,

$$\phi_{D,\nu}(\boldsymbol{x},\boldsymbol{x'}) \quad = \quad \left(\left[1-\frac{\|\boldsymbol{x}-\boldsymbol{x'}\|}{D}\right]_+\right)^{\nu} \ ,$$

where $K(\boldsymbol{x},\boldsymbol{x'})$ is the Gaussian kernel, $D>0$, and $\nu \geq \frac{M+1}{2}$ (M is the number of features). This kernel preserves positive definiteness as shown in [16]. The function $\phi_D(\cdot)$ induces sparsity since all entries satisfying $\|\boldsymbol{x}-\boldsymbol{x'}\| \geq D$ are set to zero in the kernel matrix. Therefore, the constant D is called the thresholding or truncation parameter as it regulates the support size of the kernel $K_{D,\nu}$. The parameter ν controls the degree of smoothness or differentiability of $\phi_{D,\nu}$. Different choices of D and ν produce different compactly supported kernels. When $D \to 0$, $K_{D,\nu}(\boldsymbol{x},\boldsymbol{x'})$ evaluates as zero for every $\boldsymbol{x} \neq \boldsymbol{x'}$, and it is equal to 1, otherwise. When $D \to \infty$, $K_{D,\nu}(\boldsymbol{x},\boldsymbol{x'})$ recovers the Gaussian kernel. Since the value of ν has no influence in the sparsity of the kernel, it is generally fixed at some value [41]. In this paper, we fixed $\nu = \lceil \frac{M+1}{2} \rceil$ in order to ensure positive definiteness. We normalized the parameter γ, which determines the Gaussian kernel width, by the number of features: $K(\boldsymbol{x},\boldsymbol{x'}) = \exp\left(-\frac{\gamma}{M}\|\boldsymbol{x}-\boldsymbol{x'}\|^2\right)$. We defined D as a function of γ as follows $D = (\sqrt{\gamma/M})^{-1}$ to reduce the number of parameters to adjust by cross validation. The intuition behind the definition of D is to maintain certain consistency between the Gaussian kernel width and the support size. The wider the Gaussian kernel, the larger support size.

The optimal cost parameter C and kernel width γ were those with the lowest error when performing 10 cross–validation on the training set. The cost parameter took values in the grid $\{10^i \mid i = 0, 1, \ldots, 7\}$, and the kernel width γ was selected from the grid $\{10^i \mid i = -3, -2, \ldots, 3\}$. The test set was used to report a reliable estimation of the performance of the model. The algorithm used a tolerance level of $T = 5 \cdot 10^{-2}$ to exit. We imposed a training time limit of $2,500$ seconds for the Sensor, Pendigit and Abalone datasets, and a time limit of $4,000$ seconds for the Covtype dataset. We used our C++ implementation of λ-SVMs, which is provided as Supplementary Material. The Matlab code for λ-SVMs can be also found in the Supplementary Material [33].

TABLE 2

λ-SVMs classification error rates and training times. L denotes the number of classes in the dataset. Lee et al. [23], Huerta et al. [19] and Liu and Yuan [28]'s loss functions are indicated as Lee, ISVM, and RML (Reinforced Multicategory Loss), respectively. Loss function with λ = 0.1 and ISVM loss function represent classification uncalibrated scenarios for these datasets. Classification errors and training times correspond to those values of the cost parameter (C opt.) and kernel width (γ opt.) with the lowest cross-validation error. Training times marked with (∗) correspond to cases in which the cross validation runs did not finish in the time limit for the largest values of C

		λ						
		-10,000 Lee	**0.1**	**1 ISVM**	**(L-1) RML**	**100**	**1,000**	**10,000**
Sensor	Err.(%)	0.56 ±0.13	0.42 ±0.08	0.43 ±0.10	0.46 ±0.10	0.55 ±0.11	0.45 ±0.11	0.35 ±0.07
	Time (s)	282	287	291	134	75	79	106
	C opt.	$3 \cdot 10^6$	$1 \cdot 10^6$	$3 \cdot 10^6$	$3 \cdot 10^6$	$8 \cdot 10^6$	$2 \cdot 10^6$	$1 \cdot 10^5$
	γ opt.	0.0600	0.0600	0.0700	0.0700	0.2700	0.9500	0.9500
Pendigit	Err.(%)	0.36 ±0.10	0.44 ±0.09	0.36 ±0.06	0.35 ±0.09	0.27 ±0.07	0.76 ±0.15	0.82 ±0.13
	Time (s)	1141(∗)	1180(∗)	1088(∗)	747	60	35	158
	C opt.	$1 \cdot 10^7$	$1 \cdot 10^7$	$8 \cdot 10^6$	$8 \cdot 10^6$	$1 \cdot 10^7$	$1 \cdot 10^7$	$3 \cdot 10^5$
	γ opt.	0.0260	0.0255	0.0260	0.0215	0.0450	0.4300	1.0000
Covtype	Err.(%)	13.47 ±0.11	13.47 ±0.09	13.45 ±0.11	13.47 ±0.11	13.44 ±0.09	13.44 ±0.09	13.44 ±0.09
	Time (s)	830(∗)	854(∗)	847(∗)	798	1230	1225	1227
	C opt.	$4 \cdot 10^4$	$1 \cdot 10^4$	$1 \cdot 10^5$	$4 \cdot 10^4$	$1 \cdot 10^7$	$1 \cdot 10^7$	$1 \cdot 10^7$
	γ opt.	2.5000	2.5000	2.5000	2.5000	2.5000	2.5000	2.5000
Abalone	Err.(%)	29.43 ±1.05	31.15 ±0.76	30.43 ±1.17	29.23 ±1.01	29.86 ±1.10	29.81 ±1.03	31.24 ±0.92
	Time (s)	247	350	359(∗)	221	164	30	6
	C opt.	$8 \cdot 10^5$	$1 \cdot 10^6$	$1 \cdot 10^6$	$8 \cdot 10^5$	$1 \cdot 10^7$	$1 \cdot 10^7$	$2 \cdot 10^6$
	γ opt.	0.0206	0.0026	0.0005	0.0023	0.0007	0.1000	2.2000

Table 2 shows the average classification errors and training times (in seconds) over the five test sets when different values of λ are considered in the loss function. Results correspond to the optimal cost parameter C (C *opt.*) and kernel width γ (γ *opt.*) determined by cross–validation. The values of λ were chosen to have different classification calibration scenarios according to the analysis presented in Section 4. Recall that $\lambda < -1$ recovers the classification calibrated loss function originally proposed by Lee et al. [23], $\lambda = 1$ provides the ISVM loss function [19], and $\lambda = (L-1)$ is equivalent to the reinforced multicategory hinge loss [28].

The minimum classification error is always achieved by a classification calibrated loss function with $\lambda \geq (L-1)$. In general, classification errors for $\lambda \geq (L-1)$ are either lower or similar to those corresponding to classification uncalibrated scenarios, while training times are usually lower. For example, given the optimal C and γ for each value of λ in the Pendigit dataset, λ-SVMs

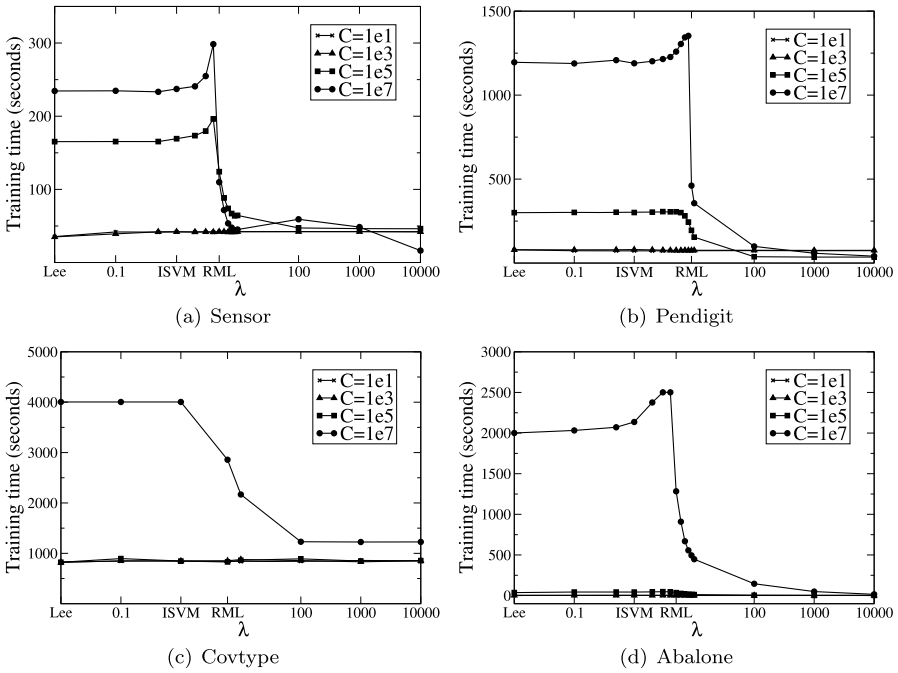


FIG 5. *λ-SVM results training times (in seconds) as a function of λ for different values of the cost parameter C. Results for the mode of the kernel parameter γ with the lowest cross-validation error for each λ and C are shown. Lee et al. [23], Huerta et al. [19], and Liu and Yuan [28]'s loss functions are indicated as Lee, ISVM, and RML (Reinforced Multicategory Loss), respectively.*

with large λ are at least 7 times faster than λ-SVMs with smaller values of λ. Classification rates for the other classification calibrated loss corresponding to $\lambda < -1$ are competitive, but training is slower than for $\lambda \geq (L-1)$. This emphasizes the importance of counting the margin of positive points in the loss function in contrast to [28]. Moreover, the fact that training did not finish for the smallest values of λ in several datasets also corroborates the remarkable relevance of shifting overweight onto the margin of the positive points. It should be noted that in the Covtype dataset, the training times for $\lambda \gg (L-1)$ are the highest since the optimal cost parameter C is set to 10^7 in these cases; however, the lowest values of λ did not explore the complete C grid given that they expired the training time limit. The following analysis of training times as a function of λ for a given γ and C will show the advantages of taking $\lambda \gg (L-1)$ in terms of computational cost.

Figure 5 shows the average training times for different values of the cost parameter C as a function of λ. In order to have comparable training times, the mode of the optimal kernel parameter γ across all the cross validation runs is chosen for each dataset. Figure 5 shows that the training times for $\lambda \gg (L-1)$ are significantly lower than those corresponding to loss functions with negative

λ or λ in the interval $((L-2)/2, (L-1))$. Differences are especially noticeable for the largest values of the cost parameter C. This result proves the advantage of strongly overweighting the margin of the positive points, and makes preferable the use of $\lambda \gg (L-1)$ instead of $\lambda < -1$ (Lee et al. loss function), $\lambda = 1$ (ISVM), or $\lambda = (L-1)$ (reinforced multicategory). Finally, the long training times observed for λ in the interval $((L-2)/2, (L-1))$ and in the proximity of $\lambda = (L-1)$ are presumably due to the presence of different KKT regions as analyzed in Section 5. Thus, avoiding values for λ in or close to the interval $((L-2)/2, L-1)$ is strongly recommended.

In short, our classification calibrated loss functions not only provide consistency guarantees that are directly reflected in the performance of the classification models, but they also provide excellent training times when the error of the positive points is significantly overweighted. A good value for λ should be large enough to strongly consider the margin of the positive points and safely keep away from the region where transitions between different families of solutions are possible. For example, setting $\lambda = 100L$ seems an appropriate choice in terms of classification calibration and training times according to our experimental results. Nevertheless, the best value for λ should ideally be determined empirically for each dataset by cross validation.

6.1. Comparison with other multiclass-SVM implementations

The aim of this section is to compare our multiclass loss function in terms of classification accuracy and computational times with other multiclass SVMs implementation and other loss functions different from those that can be treated as special cases of λ-SVM. In this section, we compare the λ-SVM solver with the MSVMpack package [21], an open source software package that implements the generic multiclass SVM formulation proposed by Guermeur [17]. MSVMpack uses a Quadratic Programming solver based on the Frank-Wolfe method [12], and each step of the descent is obtained by solving a linear program (LP) by means of the `lp_solve` solver [29]. MSVMpack implements four multiclass loss functions: Weston and Watkins [39], Crammer and Singer [9], Lee et al. [23], and Guermeur and Monfrini [18]. For more details about the MSVMpack package, the reader is referred to [21].

We followed the same experimental setup described in Section 6. We included the kernel with compact support in the MSVMpack implementation thanks to the flexibility of this software package to customize kernel functions. The Covtype dataset is not included in the comparison given its high computational cost. Both implementations, MSVMpack and λ-SVMs were configured to run in one single processor in order to better control the computational times. Please, note that our goal is not to compete with the excellent implementation provided by MSVMpack, but to provide insight into SVMs' multiclass loss functions in terms of classification calibration properties and computational cost. The results for MSVMpack for the Sensors, Pendigit, and Abalone datasets and the four loss functions (Weston and Watkins, Crammer and Singer, Lee et al., and Guermeur and Monfrini) are shown in Table 3.

TABLE 3

MSVMpack classification error rates and training times. Weston and Watkins [39], Crammer and Singer [9], Lee et al. [23], and Guermeur and Monfrini [18]'s loss functions are indicated as WW, CS, Lee, and MSVM2, respectively. Classification errors and training times correspond to those values of the cost parameter (C opt.) and kernel width (γ opt.) with the lowest cross-validation error

		Loss function			
		WW	**CS**	**Lee**	**MSVM2**
Sensor	Err.(%)	0.53 ± 0.07	0.65 ± 0.11	0.61 ± 0.05	0.46 ± 0.03
	Time (s)	612 ± 13	249 ± 15	804 ± 9	1225 ± 16
	C opt.	$5 \cdot 10^5$	$1 \cdot 10^5$	10^5	$1 \cdot 10^6$
	γ opt.	0.0004	0.0035	0.0009	0.0001
Pendigit	Err.(%)	0.27 ± 0.08	0.36 ± 0.09	0.31 ± 0.07	0.29 ± 0.08
	Time (s)	2500 ± 0	113 ± 3	760 ± 13	1246 ± 5
	C opt.	$2 \cdot 10^6$	$1 \cdot 10^5$	$1 \cdot 10^5$	$8 \cdot 10^5$
	γ opt.	0.0340	0.0170	0.0600	0.0500
Abalone	Err.(%)	30.74 ± 1.10	30.27 ± 0.86	30.74 ± 1.28	30.12 ± 1.00
	Time (s)	32 ± 4	24 ± 4	112 ± 2	91 ± 4
	C opt.	$8 \cdot 10^3$	$6 \cdot 10^4$	$3 \cdot 10^5$	$5 \cdot 10^4$
	γ opt.	0.6200	0.0226	0.0040	0.0015

MSVMpack classification rates are similar to those obtained by λ-SVM. When the optimal λ is chosen in Table 2, λ-SVMs classification rates are equal or higher than those obtained by any of the loss functions implemented by MSVMpack. This means that using our loss function only can improve the classification accuracy. Regarding the training times, in general, λ-SVMs are faster for large values of λ while maintaining competitive classification accuracies. Since MSVMpack and λ-SVMs implement Lee et al.'s loss function, both implementations can be compared. For Lee et al.'s loss function, experimental results show that (i) MSVMpack and λ-SVM provide similar results; and (ii) training times are dataset-dependent: MSVMpack implementation is faster that λ-SVM implementation in the Pendigit and Abalone datasets, while λ-SVM is faster than MSVMpack in the Sensors dataset. Overall, these experimental results show the efficiency of MSVMpack implementation, but they also reveal that there is still room for improvement in the loss function itself.

7. Conclusions

In this paper, we have proposed a family of multiclass hinge loss functions regulated by a control parameter λ that controls the margin of the positive points of a given class. These surrogate loss functions, Ψ_y, exhibit different classification calibration properties as a function of λ. We have determined the values of λ for which the proposed loss functions are classification calibrated, and we have shown that our family of loss functions allows us to define a classification calibrated hinge loss function for every multiclass classification problem. Unlike other classification calibrated hinge loss functions, we can give arbitrarily high

weight to the margin of the positive points, which is empirically shown to be positive for learning. Our family of loss functions is general enough to recover Lee et al. [23] and Liu and Yuan [28]'s classification calibrated loss functions by setting $\lambda \leq -1$ and $\lambda = (L-1)$, respectively, with L the number of classes. However, we show that other values of λ allow overcoming some limitations of previous approaches while maintaining classification calibration properties.

We have embedded our family loss functions in the Support Vector Machine's formalism (λ-SVM) and implemented a Sequential Minimum Optimization (SMO) algorithm. We have shown that the optimization algorithm has different convergence rates that can be explained in terms of the classification calibration domain and the different families of SVMs' solutions and KKT conditions defined by λ. In particular, values of $\lambda \gg (L-1)$ provide the fastest convergence while guaranteeing classification calibration.

We have compared the performance of λ-SVMs in four real-world datasets to conclude that classification calibrated loss functions considering the margin of positive points only can improve classification uncalibrated loss functions in terms of classification accuracy. Additionally, λ-SVMs with large values for λ exhibit the lowest training times, which matches with our theoretical analysis of SMO's solutions. These results reveal the importance of strongly overweighting the positive samples in the learning process.

In conclusion, a value of λ large enough would guarantee classification calibration while taking the maximum advantage of the positive examples and providing good convergence rates. Though the optimal value for λ should be determined in a validation phase, our theoretical and empirical results indicate that $\lambda = 100L$ is a good choice. It not only ensures classification calibration, but it also provides good classification performance and training times.

Appendix A: Detailed proof of Theorem 2

This Appendix provides a detailed proof of Theorem 2. In what follows, we assume that class probabilities $\{P_1, P_2, \ldots, P_L\}$ for a point $\boldsymbol{x}$ are all different and ordered as $P_1 > P_2 > \ldots > P_L$, and let $f_1, f_2, \ldots, f_L$ be the decision functions associated with these class probabilities. Before addressing the proof, let us formulate two properties of our loss function that make the classification calibration analysis more tractable.

Property 4. $\Psi_y(\boldsymbol{f}(\cdot))$ *satisfies* $\arg\min_j\{\Psi_j(\boldsymbol{f}(\boldsymbol{x_i}))\} = \arg\max_j\{f_j(\boldsymbol{x_i})\} = pred(\boldsymbol{x_i})$.

Property 5. *Given the ordered class probabilities* $P_1 > P_2 > \ldots > P_L$*, the minimizer* $\hat{\boldsymbol{f}}$ *of Eq. (6) must verify:* $\hat{f}_1 \geq \hat{f}_2 \geq \ldots \geq \hat{f}_L$.

The proof of Theorem 2 is the result of the combination of Lemmas 6–8.

Lemma 6. *Given a multiclass classification problem with* L *classes, the* λ*-parametrized family of loss functions defined in Eq. (4)–(5) is classification calibrated for* $\lambda < -1$.

Proof. Firstly, we show that the minimizer $\hat{\boldsymbol{f}}$ of Eq. (6) is lower bounded by -1 for $\lambda < -1$. The solution $f_1 = L - 1$ and $f_2 = f_3 = \ldots = f_L = -1$ is a feasible solution lower bounded by -1, and it evaluates as $(1 - P_1)L$ in Eq. (6). Let $\boldsymbol{f^1}$ be another solution with $f_j^1 < -1$. We obtain the following chain of inequalities for the objective function in Eq. (6)

$$\begin{aligned}
&\sum_{l=1}^{L} P_l[\lambda - f_l^1]_+ + (1 - P_l)[1 + f_l^1]_+ \\
&= \sum_{l=1}^{L} P_l[\lambda - f_l^1]_+ + \sum_{l \neq j} (1 - P_l)[1 + f_l^1]_+ \\
&\geq \sum_{l \neq j} (1 - P_l)[1 + f_l^1]_+ \geq \sum_{l \neq j} (1 - P_1)(1 + f_l^1) \\
&= (1 - P_1)(L - 1 - f_j^1) \geq (1 - P_1)L \; .
\end{aligned}$$

Then, any solution with $f_j < -1$ produces a larger value in the Ψ-risk than the solution $f_1 = L - 1; f_2 = f_3 = \ldots = f_L = -1$, and, thus, it cannot be minimizer. Therefore, in what follows, we only need to consider $\boldsymbol{f}$ with $f_j \geq -1$ for all $j = 1, \ldots, L$. Imposing the sum-to-zero constraint, $\sum_{l=1}^{L} f_l = 0$, we obtain the following inequalities for all f_j

$$-1 \leq f_j \leq L - 1 \; , \tag{27}$$

and, thus, all the terms $[\lambda - f_l]_+$ in Eq. (6) vanish, and the problem is equivalent to that proposed by Lee et al. in which the positive examples of a class do not take part in the loss function [23]. This case has already been shown to be classification calibrated [23, 26]. We include the proof for completeness' sake. For $\lambda < -1$, the following equality holds

$$\begin{aligned}
&\min_{\boldsymbol{f}} \sum_{l=1}^{L} P_l[\lambda - f_l]_+ + (1 - P_l)[1 + f_l]_+ \\
&= \min_{\boldsymbol{f}} \sum_{l=1}^{L} (1 - P_l)(1 + f_l) = (L - 1) - \min_{\boldsymbol{f}} \sum_{l=1}^{L} P_l f_l \; .
\end{aligned}$$

Consequently, minimizing Eq. (6) is equivalent to maximizing $\sum_{l=1}^{L} P_l f_l$. Then, the problem reduces to

$$\begin{aligned}
\max_{\boldsymbol{f}} \quad & \sum_{l=1}^{L} P_l f_l \; , \\
\text{s.t.} \quad & \sum_{l=1}^{L} f_l = 0 \; , \\
& f_l \geq -1 \qquad \text{for} \quad l = 1, 2, \ldots, L \; .
\end{aligned} \tag{28}$$

The Lagrangian of Problem (28) is given by

$$\mathcal{L}(\boldsymbol{f}, \boldsymbol{\alpha}, \mu) = \sum_{l=1}^{L} P_l f_l + \sum_{l=1}^{L} \alpha_l (f_l + 1) - \mu \sum_{l=1}^{L} f_l \ , \tag{29}$$

and the maximizer must satisfy the Karush-Kuhn-Tucker (KKT) conditions [5]:

- Stationarity: $\frac{\partial \mathcal{L}}{\partial f_l} = P_l + \alpha_l - \mu = 0$ for all $l = 1, 2, \ldots, L$.
- Primal feasibility: $f_l \geq -1$ for all $l = 1, 2, \ldots, L$, and $\sum_{l=1}^{L} f_l = 0$.
- Dual feasibility: $\alpha_l \geq 0$ for all $l = 1, 2, \ldots, L$, and $\mu \geq 0$.
- Complementary slackness: $\alpha_l (f_l + 1) = 0$ for all $l = 1, 2, \ldots, L$.

From the complementary slackness condition, we can ensure that either $f_l = -1$ (and $P_l = \mu - \alpha_l$) or $\alpha_l = 0$ (and $P_l = \mu$). Note that it is not possible to have $f_l = -1$ for all $l = 1, 2, \ldots, L$ since it violates the sum-to-zero condition, and only one f_l can have $\alpha_l = 0$ since all class probabilities are assumed to be different. Taking into account that the probability associated with $\alpha_l = 0$ is maximum since the remaining probabilities are defined as $P_i = \mu - \alpha_i$ with $\alpha_i \geq 0$, the optimal solution is $\hat{f}_1 = L - 1$ ($P_1 = \mu$) and $\hat{f}_m = -1$ ($P_m = \mu - \alpha_m$) for $m > 1$. This solution is classification calibrated. □

Lemma 7. *Given a multiclass classification problem with L classes, the λ-parametrized family of loss functions defined in Eq. (4)–(5) is classification calibrated for $\lambda \in [-1, 0) \cup ((L-2)/2, L-1]$ and classification uncalibrated for $\lambda \in [0, (L-2)/2]$.*

Proof. For the time being, let us assume that the optimal decision functions are lower bounded by -1. We show that this assumption is correct at the end of the proof.

For $-1 \leq \lambda \leq L-1$, λ generates a disjoint partition of the decision functions $\{f_l\}_{l=1}^{L}$ into the subsets $\mathcal{A} := \{l; f_l > \lambda\}$ and $\mathcal{B} := \{l; f_l \leq \lambda\}$. Then, the following equalities hold

$$\begin{aligned}
&\min_{\boldsymbol{f}} \sum_{l=1}^{L} P_l [\lambda - f_l]_+ + (1 - P_l)[1 + f_l]_+ \\
&= \min_{\boldsymbol{f}} \sum_{l=1}^{L} P_l [\lambda - f_l]_+ + (1 - P_l)(1 + f_l) \\
&= \min_{\boldsymbol{f}} \sum_{l=1}^{L} P_l \left([\lambda - f_l]_+ - f_l \right) + (L - 1) \\
&= \min_{\boldsymbol{f}} \sum_{l \in \mathcal{A}} P_l (-f_l) + \sum_{l \in \mathcal{B}} P_l (\lambda - 2 f_l) + (L - 1) \\
&= \min_{\boldsymbol{f}} \left\{ - \sum_{l \in \mathcal{A}} P_l f_l - 2 \sum_{l \in \mathcal{B}} P_l f_l \right\} + \lambda \sum_{l \in \mathcal{B}} P_l + (L - 1) \ .
\end{aligned}$$

Then, we have the following optimization problem

$$\begin{aligned} \max_{\boldsymbol{f}} \quad & \sum_{l\in\mathcal{A}} P_l f_l + 2\sum_{l\in\mathcal{B}} P_l f_l \ , \\ \text{s.t.} \quad & \sum_{l=1}^{L} f_l = 0 \ , \\ & f_l > \lambda \qquad\qquad \text{for} \quad l \in \mathcal{A} \ , \\ & f_l \leq \lambda \qquad\qquad \text{for} \quad l \in \mathcal{B} \ , \\ & f_l \geq -1 \qquad\qquad \text{for} \quad l \in \mathcal{B} \ . \end{aligned} \tag{30}$$

The Lagrangian of Problem (30) is given by

$$\begin{aligned} \mathcal{L}(\boldsymbol{f},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma},\mu) \quad = \quad & \sum_{l\in\mathcal{A}} P_l f_l + 2\sum_{l\in\mathcal{B}} P_l f_l \\ + \quad & \sum_{l\in\mathcal{A}} \alpha_l (f_l - \lambda) + \sum_{l\in\mathcal{B}} \beta_l (\lambda - f_l) \\ + \quad & \sum_{l\in\mathcal{B}} \gamma_l (f_l + 1) - \mu \sum_{l\in\mathcal{A}\cup\mathcal{B}} f_l \ . \end{aligned}$$

On the one hand, the maximizer of Problem (30) must satisfy the KKT conditions for $l \in \mathcal{A}$:

- Stationarity: $\frac{\partial \mathcal{L}}{\partial f_l} = P_l + \alpha_l - \mu = 0$ for all $l \in \mathcal{A}$.
- Complementary slackness: $\alpha_l (f_l - \lambda) = 0$ for all $l \in \mathcal{A}$.
- Primal feasibility: $f_l > \lambda$ for all $l \in \mathcal{A}$, and $\sum_{l\in\mathcal{A}\cup\mathcal{B}} f_l = 0$.
- Dual feasibility: $\alpha_l \geq 0$ for all $l \in \mathcal{A}$, and $\mu \geq 0$.

From the complementary slackness condition, we can ensure that either $f_l = \lambda$ or $\alpha_l = 0$; and, then, $P_l = \mu - \alpha_l$ or $P_l = \mu$, respectively. From the primal feasibility condition, it is not possible to have $f_l = \lambda$. Additionally, if there exists f_l with $\alpha_l = 0$, it must be unique since it implies $P_l = \mu$ and all the class probabilities are different. In fact, the probability associated with $\alpha_l = 0$ is maximum according to Property (5).

On the other hand, the maximizer of Problem (30) must satisfy the KKT conditions for $l \in \mathcal{B}$:

- Stationarity: $\frac{\partial \mathcal{L}}{\partial f_l} = 2P_l - \beta_l + \gamma_l - \mu = 0$ for all $l \in \mathcal{B}$.
- Complementary slackness: $\beta_l (\lambda - f_l) = 0$, and $\gamma_l (f_l + 1) = 0$ for all $l \in \mathcal{B}$.
- Primal feasibility: $-1 \leq f_l \leq \lambda$ for all $l \in \mathcal{B}$, and $\sum_{l\in\mathcal{A}\cup\mathcal{B}} f_l = 0$.
- Dual feasibility: $\beta_l \geq 0$ and $\gamma_l \geq 0$ for all $l \in \mathcal{B}$, and $\mu \geq 0$.

From the complementary slackness condition, we can differentiate four cases:

- CASE A: $\beta_l \neq 0$ and $\gamma_l \neq 0$. This case is impossible since it implies $f_l = \lambda$ and $f_l = -1$ simultaneously.
- CASE B: $\beta_l \neq 0$ and $\gamma_l = 0$; then, $f_l = \lambda$ and $P_l = (\mu + \beta_l)/2 > \mu/2$.
- CASE C: $\beta_l = 0$ and $\gamma_l \neq 0$; then $f_l = -1$ and $P_l = (\mu - \gamma_l)/2 < \mu/2$.

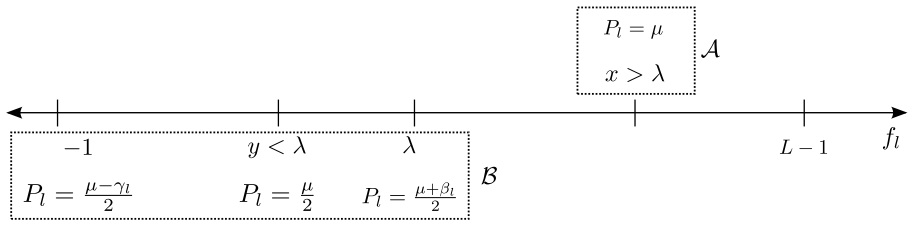


FIG 6. *Relationship between the set of class probabilities* $\{P_l\}_{l=1}^L$ *and the set of decision functions* $\{f_l\}_{l=1}^L$ *for* $-1 \leq \lambda \leq L-1$ *according to the KKT conditions of the* Ψ*-risk minimizer in Eq. (6).* x *and* y *are possible values of the decision function of a given class satisfying* $\lambda < x \leq L-1$ *and* $-1 < y < \lambda$*, respectively.*

- CASE D: $\beta_l = 0$ and $\gamma_l = 0$; then, $-1 < f_l < \lambda$ and $P_l = \mu/2$.

Figure 6 summarizes the analysis of the KKT conditions for $\mathcal{A}$ and $\mathcal{B}$ subsets. Given a fixed value for λ and according to the relationships between the decision functions and the class probabilities imposed by the KKT conditions, different configurations for $\mathcal{A}$ and $\mathcal{B}$ are possible:

CASE I: $-1 \leq \lambda < 0$. In order to satisfy $\sum_{l \in \mathcal{A} \cup \mathcal{B}} f_l = 0$, it is necessary that there exists x positive ($x > \lambda$). The decision function f_l taking the value x has to be that with the maximum class probability (Property (5)). It can be seen that the remaining decision functions take the value -1 for class probabilities lower than $P_1(1+\lambda)/(2+\lambda)$, and they evaluate as λ otherwise. In case that there exists $P_l = P_1/2$, its associated decision function f_l also takes the value λ as it maximizes the objective function in Problem (30). Assuming that n decision functions are equal to -1 and $(L-n-1)$ decision functions are equal to λ, the value of x is imposed by the primal feasibility condition $\sum_{l \in \mathcal{A} \cup \mathcal{B}} f_l = 0$: $x = n - (L-n-1)\lambda \leq (L-1)$. Therefore, our family of loss functions is classification calibrated for $-1 \leq \lambda \leq 0$.

CASE II: $\lambda = 0$. Two configurations are possible in this case:

- CASE II.1. $\boldsymbol{f} = \mathbf{0}$. The problem is classification uncalibrated since there is not a single maximum.
- CASE II.2. The decision functions associated with the n lowest class probabilities take the value -1, the decision function corresponding to the maximum probability takes the value $x = n$, and the remaining decision functions evaluate as $\lambda = 0$. Hence, the problem is classification calibrated.

In order to establish when the minimizer is characterized by CASE II.2, we need to determine when it is better for the objective function in Problem (30) to have a decision function taking the value -1 instead of $\lambda = 0$. By Property (5), it is sufficient to find out when Problem (30) is larger for $f_L = -1$ (and $f_1 = 1$) than for $f_L = 0$ (and $f_1 = 0$): $2P_L(-1) + P_1(1) > 2P_L(0) + P_1(0) \Rightarrow P_L < P_1/2$. When $P_L = P_1/2$, the objective function in (30) is $2y\frac{P_1}{2} - yP_1 = 0$ and then y

can take any value in the interval $(-1, 0)$. Therefore, the loss function is classification calibrated when $P_L < \frac{P_1}{2}$ and classification uncalibrated otherwise. This implies that the loss function is classification uncalibrated for $\lambda = 0$, since there exists at least one probability distribution that makes the loss function classification uncalibrated.

CASE III: $0 < \lambda < L - 1$. Two different cases should be analyzed:

- CASE III.1: $\mathcal{A} = \emptyset$ and $\mathcal{B} \neq \emptyset$.
- CASE III.2: $\mathcal{A} \neq \emptyset$ and $\mathcal{B} \neq \emptyset$.

Note that it is not possible to have $\mathcal{A} \neq \emptyset$ and $\mathcal{B} = \emptyset$ since the primal feasibility condition $\sum_{l \in \mathcal{A} \cup \mathcal{B}} f_l = 0$ is not satisfied.

First of all, we analyze the distribution of the solutions of Problem (30) assuming CASE III.1. The loss function is classification calibrated when the minimizer has a single $f_l = \lambda$. For the primal feasibility condition $\sum_{l \in \mathcal{A} \cup \mathcal{B}} f_l = 0$, it must be satisfied that $(-1)\,(L-2) + y + \lambda = 0$, and, thus, $y = (L-2) - \lambda$. Imposing $-1 < y < \lambda$, we get that y only exists for $\lambda > (L-2)/2$; otherwise, more than one decision function needs to be equal to λ. Then, assuming $\mathcal{A} = \emptyset$, the loss function is classification uncalibrated for $0 \leq \lambda \leq (L-2)/2$ and classification calibrated for $(L-2)/2 < \lambda \leq (L-1)$.

Now, we analyze CASE III.2 by using the results from CASE III.1. CASE III.2 is always classification calibrated as there always exists a single maximum $f_1 = x > \lambda$. Therefore, we need to determine when it is better for the objective function in Problem (30) to have f_1 in $\mathcal{A}$ ($f_1 = x > \lambda$) instead of having f_1 in $\mathcal{B}$ ($f_1 \leq \lambda$). As there always exists a feasible solution for the CASE III.1, making $f_1 = x > \lambda$ is only possible when it actually maximizes the value of the objective function in Problem (30). Then, for $(L-2)/2 < \lambda \leq (L-1)$ our family of loss functions is classification calibrated since CASE III.1 and CASE III.2 are both classification calibrated.

It remains to analyze the case $0 < \lambda \leq (L-2)/2$ where CASE III.1 is classification uncalibrated, and CASE III.2 is classification calibrated. Let us characterize the solutions in both cases.

- CASE III.1. $n_{\mathcal{B}}$ decision functions are equal to λ, a single decision function is equal to $-1 < y < \lambda$, and $(L - 1 - n_{\mathcal{B}})$ decision functions are equal to -1. According to the sum-to-zero-constraint, we have $(L - 1 - n_{\mathcal{B}})(-1) + n_{\mathcal{B}}\lambda + y = 0$, and, thus,

$$y = L - 1 - n_{\mathcal{B}}(\lambda + 1)\,. \tag{31}$$

  Since $-1 < y < \lambda$, we obtain $n_{\mathcal{B}} = \lfloor L/(\lambda+1) \rfloor$ with $n_{\mathcal{B}} \geq 2$ for $0 \leq \lambda < (L-2)/2$.
- CASE III.2. $n_{\mathcal{A}}$ decision functions are equal to λ, a single decision function is equal to $x > \lambda$, and $(L - 1 - n_{\mathcal{A}})$ decision functions are equal to -1. Without loss of generality, we assume that there is not a class probability verifying $P_l = P_1/2$, and, thus, we do not have any decision function

with value y[2]. According to the sum-to-zero-constraint, we have $(L-1-n_{\mathcal{A}})(-1)+n_{\mathcal{A}}\lambda+x=0$, and, thus,

$$x = L-1-n_{\mathcal{A}}(\lambda+1)\ . \tag{32}$$

Since $\lambda < x < L-1$, we obtain $0 \leq n_{\mathcal{A}} < \lfloor L/(\lambda+1)-1 \rfloor = n_{\mathcal{B}}-1$.

The value of the objective function in Problem (30) for CASE III.1 is

$$-2\sum_{i=n_{\mathcal{B}}+2}^{L} P_i + 2yP_{n_{\mathcal{B}}+1} + 2\lambda\sum_{i=1}^{n_{\mathcal{B}}} P_i\ , \tag{33}$$

while the value of the objective function in Problem (30) for CASE III.2 is

$$-2\sum_{i=n_{\mathcal{A}}+2}^{L} P_i + 2\lambda\sum_{i=2}^{n_{\mathcal{A}}+1} P_i + xP_1\ . \tag{34}$$

Therefore, the loss function for $0 \leq \lambda \leq (L-2)/2$ is classification uncalibrated if there exists a distribution of probabilities $\{P_i\}_{i=1}^{L}$ for which Eq. (33) is larger than Eq. (34) for all $n_{\mathcal{A}} = 0, 1, \ldots, n_{\mathcal{B}}-1$. Then, we need to impose the difference between Eq. (33) and Eq. (34) to be positive for all $n_{\mathcal{A}} = 0, 1, \ldots, n_{\mathcal{B}}-1$: $-2\sum_{i=n_{\mathcal{B}}+2}^{L} P_i + 2yP_{n_{\mathcal{B}}+1} + 2\lambda\sum_{i=1}^{n_{\mathcal{B}}} P_i - \left(-2\sum_{i=n_{\mathcal{A}}+2}^{L} P_i + 2\lambda\sum_{i=2}^{n_{\mathcal{A}}+1} P_i + xP_1\right) > 0$. Grouping terms, we obtain $2(y+1)P_{n_{\mathcal{B}}+1} + 2(\lambda+1)\sum_{i=n_{\mathcal{A}}+2}^{n_{\mathcal{B}}} P_i - (x-2\lambda)P_1 > 0$. Replacing y and x according to equalities in Eq. (31) and Eq. (32), respectively, we obtain

$$\begin{aligned} &2(L-1-n_{\mathcal{B}}(\lambda+1)+1)P_{n_{\mathcal{B}}+1} \\ &+2(\lambda+1)\sum_{i=n_{\mathcal{A}}+2}^{n_{\mathcal{B}}} P_i - (L-1-n_{\mathcal{A}}(\lambda+1)-2\lambda)P_1 > 0. \end{aligned} \tag{35}$$

To simplify the notation, we define $\theta_{n_{\mathcal{B}}+1} = 2(L-1-n_{\mathcal{B}}(\lambda+1)+1) > 0$. Note that the subset of values of λ satisfying $\theta_{n_{\mathcal{B}}+1} = 0$ is a measure-zero set as it corresponds to values of λ such that $(\lambda+1) = \alpha L$ for $\alpha \in \mathbb{Z}^+$. Then, the loss function is classification uncalibrated if we can find a probability distribution $\{P_i\}_{i=1}^{L}$ for which Eq. (35) holds for all $n_{\mathcal{A}} = 0, 1, \ldots, n_{\mathcal{B}}-1$. We have the following system of linear inequalities

$$\left\{\begin{array}{ll} P_1(L-1-0\ (\lambda+1)-2\lambda) - 2(\lambda+1)P_{n_{\mathcal{B}}} - \ldots - 2(\lambda+1)P_3 - 2(\lambda+1)P_2 & < \theta_{n_{\mathcal{B}}+1}P_{n_{\mathcal{B}}+1} \\ P_1(L-1-1\ (\lambda+1)-2\lambda) - 2(\lambda+1)P_{n_{\mathcal{B}}} - \ldots - 2(\lambda+1)P_3 & < \theta_{n_{\mathcal{B}}+1}P_{n_{\mathcal{B}}+1} \\ \vdots & \vdots \\ P_1(L-1-(n_{\mathcal{B}}-2)\ (\lambda+1)-2\lambda) - 2(\lambda+1)P_{n_{\mathcal{B}}} & < \theta_{n_{\mathcal{B}}+1}P_{n_{\mathcal{B}}+1} \\ P_1(L-1-(n_{\mathcal{B}}-1)\ (\lambda+1)-2\lambda) & < \theta_{n_{\mathcal{B}}+1}P_{n_{\mathcal{B}}+1} \end{array}\right\}.$$

From the last inequality we have $P_{n_{\mathcal{B}}+1} > (P_1\ (L-1-(n_{\mathcal{B}}-1)\ (\lambda+1)-2\lambda))/\theta_{n_{\mathcal{B}}+1}$, and, thus, $r = ((L-1-(n_{\mathcal{B}}-1)\ (\lambda+1)-2\lambda))/\theta_{n_{\mathcal{B}}+1} < 1$ is a

[2]Since the loss function is classification uncalibrated in this domain ($n_{\mathcal{B}} \geq 2$), not considering a subset of probability distributions does not affect to the analysis.

necessary condition to have a classification uncalibrated loss function. In fact, it is easy to see that $r < 1/2$ for $\lambda > 0$. Imposing $P_1(\lambda+1) - 2(\lambda+1)P_i < 0$ for all $i = 2, \ldots, n_{\mathcal{B}}$, probability distributions $\{P_i\}_{i=1}^{L}$ that make the loss function classification uncalibrated for all $0 < \lambda \leq (L-2)/2$ can be constructed as follows,

$$\begin{aligned} P_i &= a_i P_1 \qquad \text{for} \quad i = 2, \ldots, L \,, \\ P_1 &= \left(1 + \textstyle\sum_{i=2}^{L} a_i\right)^{-1} \,, \end{aligned}$$

where the coefficients a_i are any real numbers satisfying

$$\begin{aligned} \tfrac{1}{2} &< a_{n_{\mathcal{B}}} < a_{n_{\mathcal{B}}-1} < \ldots < a_3 < a_2 < 1 \,, \\ r &< a_{n_{\mathcal{B}}+1} < \tfrac{1}{2} \,, \\ 0 &< a_L < a_{L-1} < \ldots < a_{n_{\mathcal{B}}+2} < r \,. \end{aligned}$$

In particular, a probability distribution making the loss function classification uncalibrated for all $0 < \lambda \leq (L-2)/2$ is obtained by taking the limit of r when $\lambda \to 0$ (r is a decreasing function w.r.t. λ); that is, $\lim_{\lambda\to 0} r(\lambda) = \left(\frac{1}{2}\right)^{-}$ and $\lim_{\lambda\to 0} n_{\mathcal{B}} = L^{-}$. Then, the distribution given by

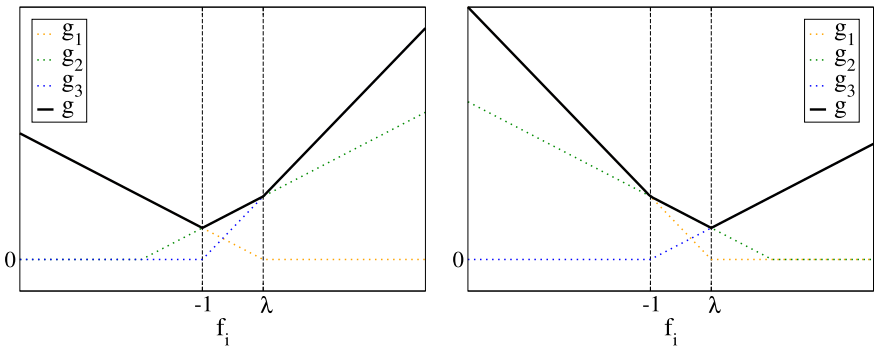
$$\begin{aligned} \tfrac{1}{2} &< a_{L-1} < a_{L-2} < \ldots < a_3 < a_2 < 1 \,, \\ a_L &= \tfrac{1}{2} \,, \end{aligned}$$

is classification uncalibrated for all $0 < \lambda \leq (L-2)/2$. Note that in the limit this distribution is the same as the one obtained for $\lambda = 0$.

To conclude, let us show that the minimizer of the empirical Ψ-risk is lower bounded by -1. The term in Eq. (6) associated with class i can be written as the following piecewise function

$$g(f_i) = \begin{cases} g_1(f_i) = P_i(\lambda - f_i) & f_i < -1 \,, \\ g_2(f_i) = (-2P_i + 1)f_i + (P_i(\lambda - 1) + 1) & -1 \leq f_i \leq \lambda \,, \\ g_3(f_i) = (1 - P_i)(1 + f_i) & f_i \geq \lambda \,, \end{cases} \tag{36}$$

and hence, the Ψ-risk for a point $\boldsymbol{x}$ is expressed as $\sum_{i=1}^{L} g(f_i)$. Function $g(f_i)$ is shown in Figure 7. The monotonicity of $g(f_i)$ in the interval $[-1, \lambda]$ depends on the prior probability P_i, being monotonically increasing for $P_i < 1/2$ and monotonically decreasing otherwise. Note that class probabilities $\{P_i\}_{i=2}^{L}$ always satisfy $P_i < 1/2$, except for the maximum probability that can be larger than $1/2$. The sum-to-zero constraint together with Property (5) force f_1 to be positive. This constraint also affects to $f_2, f_3, \ldots, f_L$, which might take values larger than -1 if making $f_2, f_3, \ldots, f_L$ equal to -1 has a negative impact on the minimizer due to the subsequent increase of f_1. In other words, if $f_{i>1}$ has a value larger than -1 in the solution of Problem (30), it means that it is not



(a) $P_i < 1/2$ (b) $P_i > 1/2$

FIG 7. *Contribution of class i to the empirical Ψ-risk of a single point as a function of the decision value f_i when the probability of the class is either 7a lower than $1/2$ or 7b larger than $1/2$.*

beneficial for the minimizer to decrease the value of this decision function by paying the cost of increasing the value of other decision functions. If a decision function $f_{j>1}$ has a value greater than -1, setting $f_j < -1$ is obviously worse than having $f_j = -1$ since it does increase not only the contribution of its own class but also the contributions of some other classes, whose decision functions are forced to augment to fulfill $\sum_{l=1}^{L} f_l = 0$. Note that an increase of f_1 could be beneficial for the minimizer only for $f_i \in (-1, \lambda)$ when $P_i > 1/2$ since $g(f_i)$ is monotonically decreasing in this domain. However, this case is impossible since it only applies to the majority class and $f_1 \geq \lambda$ according to the preceding analysis summarized in Figure 6. Without imposing decision functions to be lower bounded by -1 and assuming instead that the decision functions are lower bounded by $\nu < -1$, it is also true that $f_1 \geq \lambda$. The KKT conditions for $\boldsymbol{f}$ lower bounded by $\nu < -1$ can be easily inferred from the above analysis. In this case, the decision functions are in the set $\{\nu, y, \lambda, x\}$ with y unique and $\nu \leq y \leq \lambda < L - 1$. Then, necessarily $f_1 \geq \lambda$. □

Lemma 8. *Given a multiclass classification problem with L classes, the λ-parametrized family of loss functions defined in Eq. (4)–(5) is classification calibrated for $\lambda > L - 1$.*

Proof. Firstly, we show that the Ψ-risk minimizer $\hat{\boldsymbol{f}}$ in Eq. (6) is upper bounded by λ for $\lambda > L - 1$. Let us assume that there exists a solution $\boldsymbol{f}^1$ such that one decision function, f_j^1, is larger than λ. According to Property (5), this function has to be f_1^1. Then, we parametrize $\boldsymbol{f}^1$ as $f_1^1 = \lambda + \epsilon$ with $\epsilon > 0$, and $f_m^1 = -1 + \epsilon_m$ with $\epsilon_m \in \mathbb{R}$ for $m > 1$. As $\boldsymbol{f}^1$ is a feasible solution, it must satisfy $\sum_{l=1}^{L} f_l = 0$, and, thus, we obtain the following equality

$$\lambda + \epsilon - L + 1 + \sum_{m>1} \epsilon_m = 0 \,. \tag{37}$$

We can construct an alternative solution, $\boldsymbol{f}^2$, upper bounded by λ as $f_1^2 = \lambda$, $f_m^2 = -1$ with $1 < m < L$, and $f_L^2 = L-\lambda-2$. It can be shown that the objective function in Eq. (6) is smaller for $\boldsymbol{f}^2$ than for $\boldsymbol{f}^1$, and, thus, $\boldsymbol{f}^1$ cannot be a minimizer of Eq. (6). The difference between the value of Eq. (6) for $\boldsymbol{f}^1$ and $\boldsymbol{f}^2$ is $(1-P_1)\epsilon+\sum_{1<l<L}\{P_l(-\epsilon_l)+(1-P_l)[\epsilon_l]_+\}+P_L(-\lambda+L-1-\epsilon_L)+(1-P_L)[\epsilon_L]_+$. Replacing $(-\lambda + L - 1)$ according to Eq. (37) and taking into account that $(1-P_1) = \sum_{l>1} P_l$, we obtain $\sum_{l>1} P_l\epsilon - \sum_{l>1} P_l\epsilon_l + \sum_{l>1}(1-P_l)[\epsilon_l]_+ + P_L\epsilon + \sum_{l>1} P_L\epsilon_l$. Differentiating the subsets of positive and negative ϵ_l, we obtain

$$\begin{aligned}
& \sum_{l>1,\epsilon_l\leq 0}\{P_l(\epsilon-\epsilon_l)+P_L\epsilon_l\}+\sum_{l>1,\epsilon_l>0}\{P_l(\epsilon-2\epsilon_l)+P_L\epsilon_l+\epsilon_l\} \\
= & \sum_{l>1,\epsilon_l\leq 0}\{P_l(\epsilon-\epsilon_l)+P_L\epsilon_l\}+\sum_{l>1,\epsilon_l>0}\{P_l\epsilon+\epsilon_l(-2P_l+P_L+1)\} \\
\geq & \sum_{l>1,\epsilon_l\leq 0} P_L\epsilon+\sum_{l>1,\epsilon_l>0}\{P_l\epsilon+\epsilon_l(-2P_l+1)\}\geq 0\,.
\end{aligned}$$

Therefore, $\boldsymbol{f}^1$ cannot be minimizer, and the optimal decision functions in Eq. (6) are upper bounded by λ.

We generate a disjoint partition of the decision functions $\{f_l\}_{l=1}^L$ into the subsets $\mathcal{C} := \{l; f_l \geq -1\}$ and $\mathcal{D} := \{l; f_l < -1\}$. Then, the following equalities hold

$$\begin{aligned}
& \min_{\boldsymbol{f}} \sum_{l=1}^{L} P_l[\lambda - f_l]_+ + (1-P_l)[1+f_l]_+ \\
& = \min_{\boldsymbol{f}} \sum_{l=1}^{L} P_l(\lambda - f_l) + (1-P_l)[1+f_l]_+ \\
& = \min_{\boldsymbol{f}} \sum_{l\in\mathcal{C}} P_l(\lambda - f_l) + \sum_{l\in\mathcal{C}}(1-P_l)(1+f_l) + \sum_{l\in\mathcal{D}} P_l(\lambda - f_l) \\
& = \min_{\boldsymbol{f}} \left\{ -\sum_{l\in\mathcal{C}}(2P_l-1)f_l - \sum_{l\in\mathcal{D}} P_l f_l \right\} + \lambda + |\mathcal{C}| - \sum_{l\in\mathcal{C}} P_l
\end{aligned}$$

We need to solve the following optimization problem

$$\begin{aligned}
\max_{\boldsymbol{f}} \quad & \sum_{l\in\mathcal{C}}(2P_l-1)f_l + \sum_{l\in\mathcal{D}} P_l f_l\,, \\
\text{s.t.} \quad & \sum_{l=1}^{L} f_l = 0\,, \\
& f_l \geq -1 \qquad \text{for} \quad l\in\mathcal{C}\,, \\
& f_l \leq \lambda \qquad \text{for} \quad l\in\mathcal{C}\,, \\
& f_l < -1 \qquad \text{for} \quad l\in\mathcal{D}\,.
\end{aligned} \tag{38}$$

The Lagrangian of Problem (38) is given by

$$\begin{aligned}\mathcal{L}(\boldsymbol{f},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma},\mu) = \quad & \sum_{l\in\mathcal{C}}(2P_l-1)f_l + \sum_{l\in\mathcal{D}} P_l f_l \\ & + \sum_{l\in\mathcal{C}} \alpha_l(f_l+1) + \sum_{l\in\mathcal{C}} \beta_l(\lambda - f_l) \\ & + \sum_{l\in\mathcal{D}} \gamma_l(-1-f_l) - \mu \sum_{l\in\mathcal{C}\cup\mathcal{D}} f_l \ .\end{aligned}$$

On the one hand, the maximizer must satisfy the KKT conditions for $l \in \mathcal{C}$:

- Stationarity: $\frac{\partial \mathcal{L}}{\partial f_l} = 2P_l - 1 + \alpha_l - \beta_l - \mu = 0$ for all $l \in \mathcal{C}$.
- Complementary slackness: $\alpha_l(f_l+1) = 0$ and $\beta_l(\lambda - f_l) = 0$ for all $l \in \mathcal{C}$.
- Primal feasibility: $f_l \geq -1$ and $f_l < \lambda$ for all $l \in \mathcal{C}$, and $\sum_{l\in\mathcal{C}\cup\mathcal{D}} f_l = 0$.
- Dual feasibility: $\alpha_l \geq 0$ and $\beta_l \geq 0$ for all $l \in \mathcal{C}$, and $\mu \geq 0$.

From the complementary slackness condition, we can differentiate four cases:

- CASE A: $\alpha_l \neq 0$ and $\beta_l \neq 0$. This case is impossible since it implies $f_l = -1$ and $f_l = \lambda$ simultaneously.
- CASE B: $\alpha_l \neq 0$ and $\beta_l = 0$; then, $f_l = -1$ and $P_l = (1 + \mu - \alpha_l)/2$.
- CASE C: $\alpha_l = 0$ and $\beta_l \neq 0$; then, $f_l = \lambda$ and $P_l = (\mu + 1 + \beta_l)/2$.
- CASE D: $\alpha_l = 0$ and $\beta_l = 0$; then, $-1 < f_l < \lambda$ and $P_l = (\mu + 1)/2$.

On the other hand, the maximizer of Problem (38) must satisfy the KKT conditions for $l \in \mathcal{D}$:

- Stationarity: $\frac{\partial \mathcal{L}}{\partial f_l} = P_l - \gamma_l - \mu = 0$ for all $l \in \mathcal{D}$.
- Complementary slackness: $\gamma_l(-1 - f_l) = 0$ for all $l \in \mathcal{D}$.
- Primal feasibility: $f_l < -1$ for all $l \in \mathcal{D}$, and $\sum_{l\in\mathcal{C}\cup\mathcal{D}} f_l = 0$.
- Dual feasibility: $\gamma_l \geq 0$ for all $l \in \mathcal{D}$, and $\mu \geq 0$.

From the complementary slackness condition, we can ensure that either $f_l = -1$, which is not possible for all $l = 1, 2, \ldots, L$ given the sum-to-zero constraint, or $\gamma_l = 0$ (and $P_l = \mu$). If there exists a decision function f_l with $\gamma_l = 0$ ($f_l = z < -1$), it must be unique since all the prior probabilities are different, and it has to be that associated with the lowest probability. Figure 8 summarizes the analysis of the KKT conditions for $\mathcal{C}$ and $\mathcal{D}$. Let us analyze the two feasible scenarios:

- CASE I: $\mathcal{C} \neq \emptyset$ and $\mathcal{D} = \emptyset$. The analysis of the solutions is equivalent to CASE III.1 in the proof of Lemma 7. The number of decision functions taking the value λ is given by $n_{\mathcal{B}} = n_{\mathcal{C}} = \lfloor L/(\lambda+1) \rfloor$, which is zero for $\lambda > L - 1$. The minimizer in this case is $\hat{f}_1 = y = L - 1$ and $\hat{f}_{m>1} = -1$. Therefore, this case is classification calibrated.
- CASE II: $\mathcal{C} \neq \emptyset$ and $\mathcal{D} \neq \emptyset$. For the time being, we assume that there does not exist P_l such that $P_l = (P_L + 1)/2$, and, thus, we do not have any decision function such that $f_l = y$. Then, assuming that n decision functions take the value λ, the value of z is given by $z = n(-\lambda - 1) + L - 1$ in order to satisfy the primal constraint $\sum_{l=1}^{L} f_l = 0$. Imposing $z < -1$, we obtain $n > L/(\lambda + 1)$, which is greater than zero for $\lambda >$

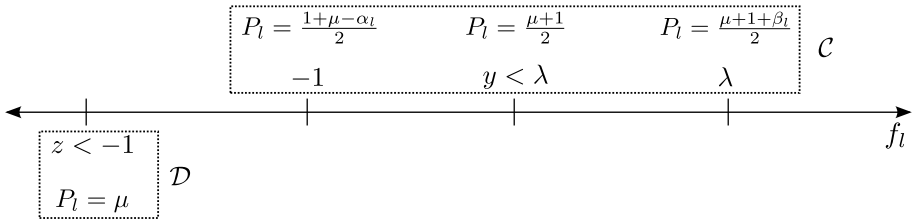


FIG 8. *Relationship between the set of class probabilities* $\{P_l\}_{l=1}^{L}$ *and the set of decision functions* $\{f_l\}_{l=1}^{L}$ *for* $\lambda > L-1$ *according to the KKT conditions of the* Ψ*-risk minimizer in Eq. (6).* y *and* z *are possible values of the decision function of a given class satisfying* $-1 < y < \lambda$ *and* $z < -1$*, respectively.*

$L-1$, and, thus, at least one decision function takes the value λ. When only one decision function takes the value λ ($n = 1$), our loss function is classification calibrated. In fact, this is the case. It is easy to see that the difference between the Ψ-risk corresponding to the solution for $n = 1$ and any other solution for $n > 1$ is negative, and, thus, the solution for $n > 1$ cannot be minimizer. Therefore, our loss functions are also classification calibrated in this case, and the minimizer is given by $\hat{f}_1 = \lambda$, $\hat{f}_L = L-\lambda-2$, and $\hat{f_m} = -1$ for $2 \leq m \leq L-1$.

The next question to solve is to determine when the minimizer of our loss functions is defined by either CASE I or CASE II ($n = 1$). The difference in the Ψ-risk between CASE I and II is $(L-1-\lambda)+2P_1(\lambda-L+1)+P_L(-\lambda-1-L)$, which is positive when $P_1 > (1+P_L)/2$. Then, the minimizer is defined by CASE I when $P_1 < (1+P_L)/2$ and by CASE II, otherwise.

Finally, the subset of probabilities not considered in the preceding analysis and corresponding to case when there exists P_l such that $P_l = (P_L+1)/2$ is also classification calibrated for $\lambda > L-1$. Note that this case is only well-defined when $P_1 > (P_L+1)/2$; otherwise, β_l must be $\beta_l < 0$, which violates the dual feasibility condition. It can be seen that having the decision functions in either subset $\{z, -1, y\}$ or $\{z, -1, y, \lambda\}$ does not improve the solution $\hat{f}_1 = \lambda$, $\hat{f}_L = L-\lambda-2$, and $\hat{f_m} = -1$ for $2 \leq m \leq L-1$ (CASE II).

Summing up, our family of loss functions is classification calibrated when $\lambda > L-1$. □

Acknowledgments

The authors acknowledge the referees' comments and suggestions that helped to improve the manuscript. This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via the Federal Bureau of Investigations, Finance Division. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of ODNI,

IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. I.R-L acknowledges partial support by Spain's grants TIN2013-42351-P (MINECO) and S2013/ICE-2845 CASI-CAM-CM (Comunidad de Madrid). The authors gratefully acknowledge the use of the facilities of Centro de Computación Científica (CCC) at Universidad Autónoma de Madrid.

Supplementary Material

C++ and Matlab implementations of λ-SVMs
(doi: 10.1214/15-EJS1073SUPP; .zip).

References

[1] Allwein, E. L., Schapire, R. E. and Singer, Y. (2001). Reducing multiclass to binary: A unifying approach for margin classifiers. *The Journal of Machine Learning Research* **1** 113–141. MR1884092

[2] Bartlett, P. L., Jordan, M. I. and Mcauliffe, J. D. (2004). Large Margin Classifiers: Convex Loss, Low Noise, and Convergence Rates. In *Advances in Neural Information Processing Systems 16* (S. Thrun, L. K. Saul and B. Schölkopf, eds.) 1173–1180. MIT Press. MR1820960

[3] Bartlett, P. L., Jordan, M. I. and McAuliffe, J. D. (2006). Convexity, classification, and risk bounds. *Journal of the American Statistical Association* **101** 138–156. MR2268032

[4] Ben-David, S., Eiron, N. and Long, P. M. (2003). On the difficulty of approximately maximizing agreements. *Journal of Computer and System Sciences* **66** 496–514. MR1981222

[5] Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press. MR2061575

[6] Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)* **2** 27.

[7] Collins, M., Schapire, R. E. and Singer, Y. (2002). Logistic regression, AdaBoost and Bregman distances. *Machine Learning* **48** 253–285.

[8] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning* **20** 273–297.

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

[10] Cristianini, N. and Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press.

[11] Feldman, V., Guruswami, V., Raghavendra, P. and Wu, Y. (2012). Agnostic learning of monomials by halfspaces is hard. *SIAM Journal on Computing* **41** 1558–1590. MR3029261

[12] Frank, M. and Wolfe, P. (1956). An algorithm for quadratic programming. *Naval research logistics quarterly* **3** 95–110. MR0089102

[13] Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences* **55** 119–139. MR1473055

[14] Friedman, J. H., Hastie, T. J. and Tibshirani, R. J. (2000). Additive logistic regression: A statistical view of boosting. *Annals of Statistics* 337–374. MR1790002

[15] Genton, M. G. (2002). Classes of kernels for machine learning: A statistics perspective. *The Journal of Machine Learning Research* **2** 299–312. MR1904760

[16] Gneiting, T. (2002). Compactly supported correlation functions. *Journal of Multivariate Analysis* **83** 493–508. MR1945966

[17] Guermeur, Y. (2012). A generic model of multi-class support vector machine. *International Journal of Intelligent Information and Database Systems* **6** 555–577.

[18] Guermeur, Y. and Monfrini, E. (2011). A quadratic loss multi-class SVM for which a radius-margin bound applies. *Informatica (ISSN 0868-4952) International Journal* **22** 73–96. MR2885660

[19] Huerta, R., Vembu, S., Amigó, J. M., Nowotny, T. and Elkan, C. (2012). Inhibition in multiclass classification. *Neural Computation* **24** 2473–2507. MR2986778

[20] Keerthi, S. S., Shevade, S. K., Bhattacharyya, C. and Murthy, K. R. K. (2001). Improvements to Platt's SMO algorithm for SVM classifier design. *Neural Computation* **13** 637–649.

[21] Lauer, F. and Guermeur, Y. (2011). MSVMpack: A multi-class support vector machine package. *The Journal of Machine Learning Research* **12** 2293–2296. MR2825427

[22] Lebanon, G. and Lafferty, J. (2001). Boosting and maximum likelihood for exponential models. *Advances in Neural Information Processing Systems* **14** 447.

[23] Lee, Y., Lin, Y. and Wahba, G. (2004). Multicategory support vector machines: Theory and application to the classification of microarray data and satellite radiance data. *Journal of the American Statistical Association* **99** 67–81. MR2054287

[24] Lichman, M. (2013). UCI Machine Learning Repository.

[25] Lin, Y. (2002). Support vector machines and the Bayes rule in classification. *Data Mining and Knowledge Discovery* **6** 259–275. MR1917926

[26] Liu, Y. (2007). Fisher consistency of multicategory support vector machines. *The Journal of Machine Learning Research–Proceedings Track* **2** 291–298.

[27] Liu, Y. and Shen, X. (2006). Multicategory ψ-learning. *Journal of the American Statistical Association* **101** 500–509. MR2256170

[28] Liu, Y. and Yuan, M. (2011). Reinforced multicategory support vector machines. *Journal of Computational and Graphical Statistics* **20** 901–919. MR2878954

[29] Notebaert, M. B., Eikland, K. and P. (2009). An Open Source (Mixed-Integer) Linear Programming System. Software available at http://lpsolve.sourceforge.net/.

[30] Platt, J. C. (1999). Fast training of support vector machines using sequential minimal optimization. In *Advances in Kernel Methods* 185–208. MIT press.

[31] Reid, M. D. and Williamson, R. C. (2010). Composite binary losses. *The Journal of Machine Learning Research* **11** 2387–2422. MR2727769

[32] Rodriguez-Lujan, I., Fonollosa, J., Vergara, A., Homer, M. and Huerta, R. (2014). On the calibration of sensor arrays for pattern recognition using the minimal number of experiments. *Chemometrics and Intelligent Laboratory Systems* **130** 123–134.

[33] Rodriguez-Lujan, I. and Huerta, R. (2015). Supplement to "A Fisher consistent multiclass loss function with variable margin on positive examples". DOI: 10.1214/15-EJS1073SUPP.

[34] Scholkopf, B. and Smola, A. J. (2001). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press.

[35] Shen, X., Tseng, G. C., Zhang, X. and Wong, W. H. (2003). On ψ-learning. *Journal of the American Statistical Association* **98** 724–734. MR2011686

[36] Steinwart, I. (2005). Consistency of support vector machines and other regularized kernel classifiers. *IEEE Transactions on Information Theory* **51** 128–142. MR2234577

[37] Tewari, A. and Bartlett, P. L. (2007). On the consistency of multiclass classification methods. *The Journal of Machine Learning Research* **8** 1007–1025. MR2320680

[38] Wang, L. and Shen, X. (2007). On L1-norm multiclass support vector machines. *Journal of the American Statistical Association* **102**. MR2370855

[39] Weston, J. and Watkins, C. (1998). Multi-class support vector machines Technical Report, Department of Computer Science, Royal Holloway, University of London.

[40] Zhang, C. and Liu, Y. (2013). Multicategory large-margin unified machines. *The Journal of Machine Learning Research* **14** 1349–1386. MR3081927

[41] Zhang, T. (2004). Statistical analysis of some multi-category large margin classification methods. *The Journal of Machine Learning Research* **5** 1225–1251. MR2248016

[42] Zhang, Z., Jordan, M. I., Li, W.-J. and Yeung, D.-Y. (2009). Coher-

ence Functions for Multicategory Margin-based Classification Methods. In *Twelfth International Conference on Artificial Intelligence and Statistics (AISTATS)* 647–654.

[43] Zhang, Z., Liu, D., Dai, G. and Jordan, M. I. (2012). Coherence functions with applications in large-margin classification methods. *The Journal of Machine Learning Research* **13** 2705–2734. MR2989912

[44] Zou, H., Zhu, J. and Hastie, T. (2008). New multicategory boosting algorithms based on multicategory fisher-consistent losses. *The Annals of Applied Statistics* 1290–1306. MR2655660