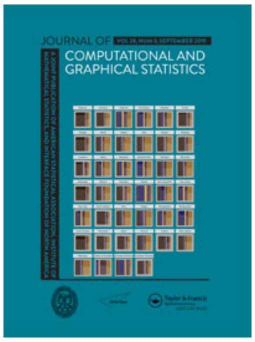

Taylor & Francis
Taylor & Francis Group

Journal of Computational and Graphical Statistics

ISSN: 1061-8600 (Print) 1537-2715 (Online) Journal homepage: https://amstat.tandfonline.com/loi/ucgs20

Multiclass Probability Estimation With Support Vector Machines

Xin Wang, Hao Helen Zhang & Yichao Wu

To cite this article: Xin Wang, Hao Helen Zhang & Yichao Wu (2019) Multiclass Probability Estimation With Support Vector Machines, Journal of Computational and Graphical Statistics, 28:3, 586-595, DOI: 10.1080/10618600.2019.1585260

To link to this article: https://doi.org/10.1080/10618600.2019.1585260

JOURNAL OF COMPUTATIONAL AND GRAPHICAL STATISTICS
2019, VOL. 28, NO. 3, 586–595
https://doi.org/10.1080/10618600.2019.1585260

Taylor & Francis
Taylor & Francis Group

Multiclass Probability Estimation With Support Vector Machines

Xin Wang[a], Hao Helen Zhang[b], and Yichao Wu[c]

[a]Qiagen, Cary, NC; [b]Department of Mathematics, University of Arizona, Tucson, AZ; [c]Department of Mathematics, Statistics, and Computer Science, University of Illinois at Chicago, Chicago, IL

ABSTRACT

Multiclass classification and probability estimation have important applications in data analytics. Support vector machines (SVMs) have shown great success in various real-world problems due to their high classification accuracy. However, one main limitation of standard SVMs is that they do not provide class probability estimates, and thus fail to offer uncertainty measure about class prediction. In this article, we propose a simple yet effective framework to endow kernel SVMs with the feature of multiclass probability estimation. The new probability estimator does not rely on any parametric assumption on the data distribution, therefore, it is flexible and robust. Theoretically, we show that the proposed estimator is asymptotically consistent. Computationally, the new procedure can be conveniently implemented using standard SVM softwares. Our extensive numerical studies demonstrate competitive performance of the new estimator when compared with existing methods such as multiple logistic regression, linear discrimination analysis, tree-based methods, and random forest, under various classification settings. Supplementary materials for this article are available online.

ARTICLE HISTORY
Received August 2017
Revised October 2018

KEYWORDS
LDA; Logistic regression; Multiclass classification; Probability estimation; Support vector machines

1. Introduction

Multiclass classification problems are commonly encountered in biomedical studies. For example, in cancer diagnosis, it is usually critical to categorize tumors into subclasses based on their phenotypic features and genomic information, to tailor the treatment and drug dose for optimal outcomes. One motivating example of this work is to classify small round blue cell tumors (SRBCTs) of childhood (Khan et al. 2001) into four subtypes: Burkitt lymphoma (BL), Ewing sarcoma (EWS), neuroblastoma (NB), and rhabdomyosarcoma (RMS), based on their gene expression profiles. The SRBCTs dataset consists of 2308 gene expression measurements, which are obtained from glass-slide cDNA microarrays following standard National Human Genome Research Institute protocols. An accurate cancer subtype classification can provide better cancer diagnosis and prognosis, leading to novel therapeutic approaches ultimately.

In a multiclass classification problem, the observations $\{(\boldsymbol{x}_i, y_i), i = 1, 2, \ldots, n\}$ are randomly drawn from a distribution $P(\boldsymbol{X}, Y)$, where $\boldsymbol{x}_i \in \mathcal{S} \subset R^d$ is the input vector and $y_i \in \{1, 2, \ldots, k\}$ is the class label. Here n is the sample size, d is the input dimensionality, and $k \geq 3$ is the number of classes. The main task is to learn a decision function $f : \mathcal{S} \to \{1, \ldots, k\}$ which assigns a class label to a data point based on its input. See Agresti and Coull (1998) and references therein for a comprehensive overview of classical methods for multiclass classification. There are two types of classifiers: hard classifiers and soft classifiers. Hard classifiers learn the classification boundary directly, and popular examples include support vector machines (SVMs, Cortes and Vapnik 1995; Vapnik 1998), multicategory psi-learning (Liu and Shen 2006; Qiao and Liu 2009), multiclass boosting algorithms (Zou, Zhu, and Hastie 2008; Wang 2013), multiclass adaboost (Zhu et al. 2009). Soft classifiers first estimate the class probabilities $p_j(\boldsymbol{x}) = P(Y = j|\boldsymbol{X} = \boldsymbol{x}), j = 1, 2, \ldots, k$, and then construct the decision rule using the argmax rule $f(\boldsymbol{x}) = \arg\max_{j=1,\ldots,k} p_j(\boldsymbol{x})$. Commonly used soft classifiers include multiple logistic regression, linear discriminant analysis (LDA), and quadratic discriminant analysis (QDA).

Multiclass SVMs have been widely studied in literature, including Weston and Watkins (1999), Lee, Lin, and Wahba (2004), Wang and Shen (2007), Liu (2007), Liu and Yuan (2011), Zhang and Liu (2013), Huang et al. (2013), etc., and the methods are shown to achieve high prediction accuracy in various applications such as cancer diagnosis, handwritten digits recognition, junk email detection (Burges 1998; Cristianini and Shawe-Taylor 2000; Zhu et al. 2004). Lin (2002) shows that binary SVMs target directly on the Bayes classification boundary without estimating class probabilities. Zhang (2004) proves some general results on the consistency of multiclass classification methods. However, one major limitation of standard SVMs is that they do not provide class probability estimates. Take cancer diagnosis as an example. In addition to labeling a patient as "subtype A" or "subtype B," it is often desired to report some uncertainty measure about the classification decision as well. Many methods have been proposed for multiclass probability estimation from various perspectives, including multiple logistic regression, classification tree methods (Breiman et al. 1984), kernel regression (Zhu and Hastie 2005; Tu and Wang

CONTACT Hao Helen Zhang hzhang.work@gmail.com North Carolina State University, 2311 Stinson Drive, Campus Box 8203, 5220 SAS Hall, Raleigh, NC 27695-8203.

Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/r/JCGS.

Supplementary materials for this article are available online. Please go to www.tandfonline.com/r/JCGS.

2013). Researchers have been intrigued by how to endow SVMs with the feature of estimating class probabilities. For binary problems, Wang, Shen, and Liu (2008) suggested the idea of first training a series of weighted SVMs (WSVMs) and then aggregating multiple binary decisions to construct class probabilities. Multiclass problems are however much more difficult than binary ones due to their intrinsic complexity.

Recently, Wu, Zhang, and Liu (2010) generalized Wang, Shen, and Liu (2008) from $k = 2$ to $k \geq 3$ and showed its promising performance when the number of classes k is moderate. In practice, one concern about Wu, Zhang, and Liu (2010) is its high computational cost for a large k. The total number of WSVM classifiers required to fit by the method increases exponentially with k, so the method is slow when $k \geq 4$. In addition, whenever a new class is added to the data, the method needs to refit the previous solutions since the decision is based on considering all the classes simultaneously. In this article, we propose a simple yet effective approach to estimating multiclass probabilities from SVMs. The key idea is as follows: decompose multiclass problems into multiple binary problems, estimate class probabilities for each pair of classes by using binary WSVMs, and construct multiclass probabilities by assembling pairwise probability estimates. Compared to Wu, Zhang, and Liu (2010), the new method is much faster and its computational cost is quadratic in k. The new method can handle multiclass problems with a larger k, say, $k = 10$. One can implement the procedure by standard binary SVM software or R packages without extra programming effort. Furthermore, due to its divide-and-conquer nature, the new method enjoys parallel computing. Theoretically, the new estimator is asymptotically consistent. Our extensive numerical studies demonstrate its competitive performance compared to existing methods such as multiple logistic regression and tree-based methods. The idea of multiclass probability estimation by pairwise coupling has been actively studied in literature by Hastie and Tibshirani (1998), Wu, Lin, and Weng (2004), Van Calster et al. (2007), and various schemes have been suggested. One major advantage of the new estimator is that it is model-free and fully nonparametric, while other methods either assume the availability of pairwise class probabilities or estimate them using the estimates of some prespecified forms.

The rest of the article is organized as follows. Section 2 presents the main methodology and theoretical properties of the proposed multiclass probability estimator. Section 3 provides an efficient computational algorithm for implementation. Section 4 conducts simulated studies, and Section 5 presents real data examples, followed by concluding remarks in Section 6.

2. Main Methodology

2.1. Weighted SVMs for Binary Classification

In binary classification problems, the class label y is typically coded as $\{+1, -1\}$ for convenience. Denote the posterior probability for $+1$ class by $p_1(\boldsymbol{x}) = P(Y = +1 | \boldsymbol{X} = \boldsymbol{x})$. Large-margin classifiers, including SVMs, train the classifiers by solving a regularization problem of the form

$$\min_{f \in \mathcal{F}} n^{-1} \sum_{i=1}^{n} L(y_i f(\boldsymbol{x}_i)) + \lambda J(f), \qquad (1)$$

where $L(\cdot)$ is the loss function, $\mathcal{F}$ is some functional space containing f, $J(f)$ is a penalty operator to control model complexity, and $\lambda > 0$ is the regularization parameter which balances the data fit and the model complexity. The product term $y_i f(\boldsymbol{x}_i)$ is known as the *functional margin*. Standard binary SVMs use the hinge loss $L(z) = (1-z)_+ = \max\{0, 1-z\}$. If the function f has a linear form $f(\boldsymbol{x}) = \beta_0 + \boldsymbol{x}^T \boldsymbol{\beta}_1$, we call it a linear classifier. To achieve nonlinear classification, we use a bivariate Mercer kernel $\mathbf{K}(\cdot, \cdot)$ in the learning process and train a flexible classifier f of the form $\beta_0 + \sum_{i=1}^{n} \theta_i \mathbf{K}(\boldsymbol{x}_i, \boldsymbol{x})$, due to the representer theorem (Kimeldorf and Wahba 1971). For kernel SVMs, the functional space $\mathcal{F}$ is the reproducing kernel Hilbert space (RKHS, Wahba 1990) induced by $\mathbf{K}(\cdot, \cdot)$, denoted as $\mathcal{H}_{\mathbf{K}}$, and the penalty $J(f) = \|f\|^2_{\mathcal{H}_{\mathbf{K}}} = \sum_{i=1}^{n} \sum_{l=1}^{n} \theta_i \theta_l \mathbf{K}(\boldsymbol{x}_i, \boldsymbol{x}_l)$. The regularization problem has the form

$$\min_{\beta_0, \theta_1, \ldots, \theta_n} n^{-1} \sum_{i=1}^{n} L(y_i f(\boldsymbol{x}_i)) + \lambda \sum_{i=1}^{n} \sum_{l=1}^{n} \theta_i \theta_l \mathbf{K}(\boldsymbol{x}_i, \boldsymbol{x}_l),$$
$$\text{where } f(\boldsymbol{x}) = \beta_0 + \sum_{i=1}^{n} \theta_i \mathbf{K}(\boldsymbol{x}_i, \boldsymbol{x}). \qquad (2)$$

Lin (2002) proved that the theoretical minimizer of the expectation of hinge loss $E[1 - Yf(X)]_+$ has the same sign as the Bayes rule $\text{sign}[p_1(\boldsymbol{x}) - \frac{1}{2}]$. In other words, binary SVMs target directly on the Bayes rule without estimating $p_1(\boldsymbol{x})$.

Wang, Shen, and Liu (2008) proposed a novel approach to estimating $p_1(\boldsymbol{x})$ using the weighted SVM for binary classification problems. The basic idea is as follows: assign data points from class -1 (and from class $+1$) with a weight π (and $1 - \pi$), and minimize the weighted hinge loss

$$\min_{f \in \mathcal{H}_K} n^{-1} \left[(1 - \pi) \sum_{y_i = 1} L(y_i f(\boldsymbol{x}_i)) + \pi \sum_{y_i = -1} L(y_i f(\boldsymbol{x}_i)) \right]$$
$$+ \lambda J(f), \qquad (3)$$

where $0 \leq \pi \leq 1$. One can show that, the theoretical minimizer of the expectation of the weighted hinge loss $E\{(1 - \pi) I(Y = +1)[1 - Yf(\boldsymbol{X})]_+ + \pi I(Y = -1)[1 + f(\boldsymbol{X})]_+\}$ has the same sign as $\text{sign}[p_1(\boldsymbol{X}) - \pi]$ for any fixed π. Consequently, using a series of π values $0 < \pi_1 < \cdots < \pi_M < 1$, one can solve (3) repeatedly and obtain multiple classifiers $\hat{f}_{\pi_1}, \ldots, \hat{f}_{\pi_M}$. For any $\boldsymbol{x}$, there exists a unique m^* such that $\hat{f}_{\pi_{m^*}}(\boldsymbol{x})$ and $\hat{f}_{\pi_{m^*+1}}(\boldsymbol{x})$ have opposite signs, implying that π_{m^*} and π_{m^*+1} satisfy $\text{sign}[p_1(\boldsymbol{x}) - \pi_{m^*}] \neq \text{sign}[p_1(\boldsymbol{x}) - \pi_{m^*+1}]$. This leads to a consistent probability estimator $\hat{p}_1(\boldsymbol{x}) = \frac{1}{2}(\pi_{m^*} + \pi_{m^*+1})$. Wang, Shen, and Liu (2008) showed that this estimator has competitive numerical performance, compared to other approaches like Platt's method (Platt 1999).

2.2. New Method for Multiclass Probability Estimation

We consider multiclass classification problems with $k \geq 3$, with y being coded as $\{1, 2, \ldots, k\}$. Wu, Zhang, and Liu (2010) extend the scheme of Wang, Shen, and Liu (2008) from binary to multiclass problems by solving a series of multiclass WSVM problems, each problem associated with a weight vector $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_k) \in A_k$, where $A_k = \{(\pi_1, \ldots, \pi_k) : 0 \leq \pi_j \leq$

$1, \sum_{j=1}^{k} \pi_j = 1\}$. To construct class probabilities, Wu, Zhang, and Liu (2010) need to train multiple multiclass WSVMs, whose weight vectors cover the entire hyperplane A_k. Consequently, the computational cost of the procedure is high if k is large, as the number of grid points covering A_k increases exponentially with k. This motivates us to develop a simpler and faster alternative to estimate multiclass probabilities based on SVMs. One main advantage of the new method is its computational feasibility for large k.

Define the posterior class probabilities $p_j(\boldsymbol{x}) = P(Y = j|\boldsymbol{X} = \boldsymbol{x}), j = 1, \dots, k$. For each pair of classes j and j' with $j \neq j'$, define the pairwise conditional probability as

$$q_{j|(j,j')}(\boldsymbol{x}) = \frac{P(Y = j|\boldsymbol{X} = \boldsymbol{x})}{P(Y = j|\boldsymbol{X} = \boldsymbol{x}) + P(Y = j'|\boldsymbol{X} = \boldsymbol{x})}. \quad (4)$$

The quantity $q_{j|(j,j')}(\boldsymbol{x})$ can be interpreted as the conditional probability of a data point with $\boldsymbol{X} = \boldsymbol{x}$ belonging to class j given that it belongs to either class j or class j'. By definition, $q_{j|(j,j')}(\boldsymbol{x}) + q_{j'|(j,j')}(\boldsymbol{x}) = 1$ for any $j \neq j'$ and any $\boldsymbol{x} \in \mathcal{S}$. It is easy to see that

$$\frac{q_{j|(j,j')}(\boldsymbol{x})}{q_{j'|(j,j')}(\boldsymbol{x})} = \frac{P(Y = j|\boldsymbol{X} = \boldsymbol{x})}{P(Y = j'|\boldsymbol{X} = \boldsymbol{x})} = \frac{p_j(\boldsymbol{x})}{p_{j'}(\boldsymbol{x})}, \quad j' \neq j.$$

In the following, Lemma 1 shows that the class probability $p_j(\boldsymbol{x})$'s can be constructed from the pairwise conditional probabilities.

Lemma 1. For any $j \in \{1, \dots, k\}$, we have

$$p_j(\boldsymbol{x}) = \frac{q_{j|(j,j')}(\boldsymbol{x})/q_{j'|(j,j')}(\boldsymbol{x})}{\sum_{l=1}^{k} q_{l|(l,j')}(\boldsymbol{x})/q_{j'|(l,j')}(\boldsymbol{x})}, \quad \text{for } j' \neq j. \quad (5)$$

The above holds for any arbitrary choice of j'.

Proof. For any fixed j and $j' \neq j$, we have

$$\frac{q_{j|(j,j')}(\boldsymbol{x})/q_{j'|(j,j')}(\boldsymbol{x})}{\sum_{l=1}^{k} q_{l|(l,j')}(\boldsymbol{x})/q_{j'|(l,j')}(\boldsymbol{x})} = \frac{p_j(\boldsymbol{x})/p'_j(\boldsymbol{x})}{\sum_{l=1}^{k} p_l(\boldsymbol{x})/p_{j'}(\boldsymbol{x})}$$

$$= \frac{p_j(\boldsymbol{x})}{\sum_{l=1}^{k} p_l(\boldsymbol{x})} = p_j(\boldsymbol{x}).$$

□

The idea of the new estimator is described as follows. We first decompose a multiclass classification problem into multiple binary problems. Then, for each pair (j, j') with $1 \leq j < j' \leq k$, we fit the binary WSVMs and construct the pairwise conditional probability estimates $\hat{q}_{j|(j,j')}$. Finally, we compute $\hat{p}_j(\boldsymbol{x})$'s using (5). The following is the computational algorithm:

Step 1: For each pair (j, j') with $1 \leq j < j' \leq k$, define the univariate function $R_{j,j'}(y) = 1$ if $y = j$; $= -1$ if $y = j'$. We fit a series of kernel WSVMs by solving

$$\min_{f \in \mathcal{H}_K} n^{-1} \left[(1 - \pi_m) \sum_{y_i = j} L(R_{j,j'}(y_i) f(\boldsymbol{x}_i)) + \pi_m \sum_{y_i = j'} L(R_{j,j'}(y_i) f(\boldsymbol{x}_i)) \right] + \lambda J(f) \quad (6)$$

over a grid of points $0 < \pi_1 < \cdots < \pi_M < 1$. For each $m = 1, \dots, M$, denote the solution to (6) as $\hat{f}_{j,j',\pi_m}(\boldsymbol{x})$.

Step 2: For each pair (j, j'), construct the pairwise conditional probability estimate as

$$\hat{q}_{j|(j,j')}(\boldsymbol{x}) = \left[\arg\min_{\pi_m} \{\hat{f}_{j,j',\pi_m}(\boldsymbol{x}) < 0\} + \arg\max_{\pi_m} \{\hat{f}_{j,j',\pi_m}(\boldsymbol{x}) > 0\} \right] /2, \quad \forall \boldsymbol{x} \in \mathcal{S}.$$

Step 3: Compute the posterior class probabilities estimates as

$$\hat{p}_j(\boldsymbol{x}) = \frac{\hat{q}_{j|(j,j')}(\boldsymbol{x})/\hat{q}_{j'|(j,j')}(\boldsymbol{x})}{\sum_{l=1}^{k} \hat{q}_{l|(l,j')}(\boldsymbol{x})/\hat{q}_{j'|(l,j')}(\boldsymbol{x})}, \quad j = 1, \dots, k. \quad (7)$$

In Equation (7), we have slightly abused the notation by defining $\hat{q}_{j|(j,j)}(\boldsymbol{x}) = 1$ for any $\boldsymbol{x}$. For now, we assume that the regularization parameter λ is fixed at Step 1. In Section 3, we will discuss the issue of parameter tuning. Since the WSVM is model-free, the new estimator does not rely on any parametric model assumption on the data and is hence robust.

Next, we establish the consistency of the proposed probability estimator (7), which provides theoretical justifications for the new estimator. We first start with Lemma 2, which states theoretical optimality of the minimizer of the expected weighted hinge loss for the pairwise binary classification problems. Proof of Lemma 2 follows similar arguments as the proof of Lemma 1 in Wang, Shen, and Liu (2008). In the following, we only outline key steps in the proof and refer to Wang, Shen, and Liu (2008) for more details.

Lemma 2. Assume $0 < \pi < 1$. For any class $j \in \{1, \dots, k\}$, choose $j' \neq j$. Define

$$A(f) = E\Big[(1 - \pi) I(Y = j) L(R_{j,j'}(Y) f(\boldsymbol{X})) + \pi I(Y = j') L(R_{j,j'}(Y) f(\boldsymbol{X}))\Big].$$

The minimizer of $A(f)$ is given by $f^*(\boldsymbol{X}) = q_{j|(j,j')}(\boldsymbol{X}) - \pi$.

Proof. We fix $\pi \in (0, 1)$ and $j \neq j' \in \{1, \dots, k\}$, since the following argument holds for arbitrary choices of their values. For notation convenience, denote the class label for the (j, j')-classification problem by $\widetilde{Y} = R_{j,j'}(Y)$; in other words, $\widetilde{Y} = +1$ if $Y = j$ and $\widetilde{Y} = -1$ if $Y = j$. Given $\boldsymbol{X}$, the label $\widetilde{Y}$ follows a binary distribution $P(\widetilde{Y} = +1|\boldsymbol{X}) = q_{j|(j,j')}(\boldsymbol{X})$ and $P(\widetilde{Y} = -1|\boldsymbol{X}) = q_{j'|(j,j')(X)} = 1 - q_{j|(j,j')}(\boldsymbol{X})$. Furthermore, define the weight function $W(\widetilde{Y}) = 1 - \pi$ if $\widetilde{Y} = +1$ and $W(\widetilde{Y}) = \pi$ if $\widetilde{Y} = -1$. Then we have $A(f) = E\big[W(\widetilde{Y})(1 - \widetilde{Y}f)_+\big]$. For any f, define its truncation to the interval $[-1, +1]$ as: $f_{\pm1} = f$ when $|f| \leq 1$ and sign(f) otherwise. Since $A(f) \geq A(f_{\pm1})$, the minimizer of $A(f)$ must take values in $[-1, +1]$. For any f taking value in $[-1, +1]$, we have $(1 - \widetilde{Y}f)_+ = 1 - \widetilde{Y}f$, and therefore $\min_f A(f) = EW(\widetilde{Y}) - \max_f E\big\{E\big[W(\widetilde{Y})\widetilde{Y}|\boldsymbol{X}\big] f(\boldsymbol{X})\big\}$. It is easy to see that $E\big[W(\widetilde{Y})\widetilde{Y}|\boldsymbol{X}\big] = P(\widetilde{Y} = +1|\boldsymbol{X})(1 - \pi) - (1 - P(\widetilde{Y} = +1|\boldsymbol{X}))\pi = P(\widetilde{Y} = +1|\boldsymbol{X}) - \pi = q_{j|(j,j')}(\boldsymbol{X}) - \pi$. Therefore, the minimizer of $A(f)$ is given by sign $\big(q_{j|(j,j')}(\boldsymbol{X}) - \pi\big)$. □

Lemma 2 essentially states that, for each pair of classes $j \neq j'$ and for any $\pi \in (0, 1)$, the minimizer of the weighted SVM directly estimates $q_{j|(j,j')}(\boldsymbol{x}) - \pi$. Following Wang, Shen, and Liu

(2008), we can show that $\hat{q}_{j|(j,j')}(\boldsymbol{x})$ converges to $q_{j|(j,j')}(\boldsymbol{x})$ as the sample n goes to infinity, as long as the functional space is rich enough. In the following Theorem 1, we show that our probability estimator is consistent for $p_j(\boldsymbol{x})$'s under general conditions. The results are obtained by using the conclusion of Lemma 2 and the relationship between $q_{j|(j,j')}(\boldsymbol{x})$ and $p_j(\boldsymbol{x})$'s. The proof is similar to that of Theorem 2 in Wang, Shen, and Liu (2008) and hence omitted. For convenience, we define $\pi_0 = 0$ and $\pi_{M+1} = 1$. For a grid $0 = \pi_0 < \pi_1 < \cdots < \pi_M < \pi_{M+1} = 1$, we define the grid size $d_\pi = \max\{\pi_m - \pi_{m-1}, m = 1, \ldots, M+1\}$.

Theorem 1. Define the estimated class probabilities $\hat{p}_j(\boldsymbol{x}), j = 1, \ldots, k$ from the solutions to (6) and (7). Then if $\lambda \to 0$ and the grid size $d_\pi \to 0$ as $n \to \infty$, the proposed probability estimators are asymptotically consistent, that is, $\hat{p}_j(\boldsymbol{x}) \to p_j(\boldsymbol{x})$ for $j = 1, 2, \ldots, k$ as $n \to \infty$.

Remark. Though we focus on the SVM only, the proposed estimation scheme and theoretical results can be extended to other large-margin classifiers as long as the loss is Fisher consistent.

3. Computation and Implementation

3.1. Kernel Learning Optimization

To train the weighted SVMs, we use a sequence of weights $\Pi_M = \{\frac{m}{M}, m = 0, \ldots, M\}$ where $M > 0$ is an integer. For each pair of classes (j, j'), we solve the optimization problem (6) and get the solution $f(\boldsymbol{x}) = \beta_0 + \sum_{i=1}^n \theta_i \mathbf{K}(\boldsymbol{x}_i, \boldsymbol{x})$, which has a finite representation due to Kimeldorf and Wahba (1971) and Wahba (1990). Correspondingly, the roughness penalty becomes $J(f) = \sum_{i=1}^n \sum_{l=1}^n \theta_i \theta_l \mathbf{K}(\boldsymbol{x}_i, \boldsymbol{x}_l)$. By introducing slack variables $\xi_i > 0$, $i = 1, \ldots, n$, we can reformulate the optimization problem (6) as the following equivalent form:

$$\min_{\beta_0, \theta_1, \ldots, \theta_n, \xi_1, \ldots, \xi_n} \left[(1 - \pi_m) \sum_{y_i = j} \xi_i + \pi_m \sum_{y_i = j'} \xi_i \right] + \lambda \sum_{i=1}^n \sum_{l=1}^n \theta_i \theta_l \mathbf{K}(\boldsymbol{x}_i, \boldsymbol{x}_l) \qquad (8)$$

$$\text{subject to } \xi_i \geq 0, \quad i = 1, \ldots, n;$$

$$\xi_i \geq 1 - R_{j,j'}(y_i) \left(\sum_{l=1}^n \theta_l \mathbf{K}(\boldsymbol{x}_l, \boldsymbol{x}_i) + \beta_0 \right),$$

$$i \in \{m : y_m = j \text{ or } j'\}.$$

The optimization problem (8) is a standard quadratic programming (QP) problem, which can be solved by standard packages in R and MATLAB. For any fixed $\pi_m \in \Pi_M$, denote the solution to (8) as $\hat{f}^{\lambda}_{j,j',\pi_m}(\boldsymbol{x}) = \hat{\beta}_0 + \sum_{i=1}^n \hat{\theta}_i \mathbf{K}(\boldsymbol{x}_i, \boldsymbol{x})$. After collecting all $\hat{f}^{\lambda}_{j,j',\pi_m}(\boldsymbol{x}), \pi_m \in \Pi_M$, we can compute $\hat{q}^{\lambda}_{j|(j,j')}(\boldsymbol{x}) = [\arg\min_{\pi_m \in \Pi_M} \{\hat{f}_{j,j',\pi_m}(\boldsymbol{x}) < 0\} + \arg\max_{\pi_m \in \Pi_M} \{\hat{f}_{j,j',\pi_m}(\boldsymbol{x}) > 0\}]/2$. The estimated pairwise conditional probability $\hat{q}^{\lambda}_{j|(j,j')}(\boldsymbol{x})$ depends on the regularization parameter λ, which should be selected adaptively using the data. We discuss the tuning issue in next section.

3.2. Parameter Tuning

The regularization parameter λ in (8) needs to be tuned adaptively with the data. For a fixed λ, we get an estimated pairwise conditional probability $\hat{q}^{\lambda}_{j|(j,j')}(\cdot)$, whose closeness to the true probability $q_{j|j,j'}(\cdot)$ can be measured by the generalized Kullback–Leibler (GKL) distance

$$\begin{aligned} &\mathrm{GKL}(q_{j|(j,j')}, \hat{q}^{\lambda}_{j|j,j')}) \qquad (9) \\ &= E\left[q_{j|(j,j')}(\boldsymbol{X}) \log \frac{q_{j|(j,j')}(\boldsymbol{X})}{\hat{q}^{\lambda}_{j|(j,j')}(\boldsymbol{X})} \right. \\ &\left. \quad + (1 - q_{j|(j,j')}(\boldsymbol{X})) \log \frac{1 - q_{j|(j,j')}(\boldsymbol{X})}{1 - \hat{q}^{\lambda}_{j|(j,j')}(\boldsymbol{X})} \right] \\ &= C - E\left[q_{j|(j,j')}(\boldsymbol{X}) \log \hat{q}^{\lambda}_{j|(j,j')}(\boldsymbol{X}) \right. \\ &\left. \quad + (1 - q_{j|(j,j')}(\boldsymbol{X})) \log(1 - \hat{q}^{\lambda}_{j|(j,j')}(\boldsymbol{X})) \right], \end{aligned}$$

where the constant $C = E[q_{j|(j,j')}(\boldsymbol{X}) \log q_{j|(j,j')}(\boldsymbol{X}) + (1 - q_{j|(j,j')}(\boldsymbol{X})) \log(1 - q_{j|(j,j')}(\boldsymbol{X}))]$, which does not depend on $\hat{q}^{\lambda}_{j|(j,j')}(\cdot)$, and the expectation is taken with respect to $\boldsymbol{X}$.

In practice, the quantity GKL is not computable because it involves the true $q_{j|(j,j')}$'s, which are generally unknown. Therefore, we need to derive a computable proxy of GKL using the data. Note that $E\left[(R_{j,j'}(Y) + 1)/2 | \boldsymbol{X}, Y \in \{j, j'\}\right] = q_{j|(j,j')}(\boldsymbol{X})$. After removing C from (9), we obtain an empirical version of GKL distance (up to a constant)

$$\begin{aligned} &\mathrm{EGKL}(\hat{q}^{\lambda}_{j|(j,j')}) \qquad (10) \\ &= -\frac{1}{2n_{j,j'}} \sum_{i: y_i = j \text{ or } j'} \left[(1 + R_{j,j'}(y_i)) \log \hat{q}^{\lambda}_{j|(j,j')}(\boldsymbol{x}_i) \right. \\ &\left. \quad + (1 - R_{j,j'}(y_i)) \log(1 - \hat{q}^{\lambda}_{j|(j,j')}(\boldsymbol{x}_i)) \right], \end{aligned}$$

where $n_{j,j'}$ denotes the number of observations with $y_i = j$ or j'. Since EGKL approximates GKL up to a constant C, it provides a proper measure to evaluate $\hat{q}^{\lambda}_{j|(j,j')}(\boldsymbol{x})$. The same type of measure was used in Wang, Shen, and Liu (2008). In the simulated examples, we generate a tuning set of size n and compute EGKL based on the tuning set, and the optimal λ is chosen as the minimizer of EGKL. In the real data examples, we split the data into the training set and the tuning set, and then the tuning set is used to compute EGKL and select the optimal λ.

3.3. Merging Pairwise Conditional Probabilities

Lemma 1 shows that different choices of j' would yield the same solution for the new estimator. Therefore, from a theoretical perspective, the proposed estimator does not depend on the choice of the baseline class j'. In practice, there might be slight differences among the solutions obtained using different j' due to the numerical variation. For the sake of computational stability, we suggest using j' which gives the largest pairwise conditional probability estimate $\hat{q}_{j'|(j,j')}(\boldsymbol{x})$. As implied by Equation (7), the quality of the estimates relies heavily on the ratio $\hat{q}_{j|(j,j')}/\hat{q}_{j'|(j,j')}$, so we choose j' to prevent the denominator from being too close to zero. In particular, this choice of j' can be achieved as follows. For each $\boldsymbol{x}$, we compare $\hat{q}_{j|(j,j')}(\boldsymbol{x})$ and $\hat{q}_{j'|(j,j')}(\boldsymbol{x})$ to see which gives a bigger value for each pair of classes $(j,j'), j \neq j'$. In practice, we suggest to first fit the binary classifiers all possible pairs, and then identify the class $\hat{l}(\boldsymbol{x})$ which produces larger pairwise conditional probability values most frequently. The final probability estimator is then defined as $\hat{p}_j(\boldsymbol{x}) = \hat{p}_{j|(j,\hat{l}(\boldsymbol{x}))}(\boldsymbol{x})$.

4. Numerical Studies

We illustrate empirical performance of the new estimator under five scenarios and compare it with existing methods, including cumulative logit model (CLM), baseline logit model (BLM), kernel multi-category logistic regression (KMLR, Zhu and Hastie 2005), classification tree (TREE, Breiman et al. 1984), the multiclass WSVM method (Wu, Zhang, and Liu 2010) denoted as WZL (2010), random forest (RF), and a recent proposal (Tu and Wang 2013) denoted as XW (2013). The CLM assumes that $\log \frac{\sum_{l=1}^{j} p_l(\boldsymbol{x})}{1-\sum_{l=1}^{j} p_l(\boldsymbol{x})} = \beta_{j0} + \boldsymbol{x}^T\boldsymbol{\beta}_j, j = 1, 2, \ldots, k-1$. The BLM method uses one class (say class k) as the baseline class and assumes $\log \frac{p_j(\boldsymbol{x})}{p_k(\boldsymbol{x})} = \beta_{j0} + \boldsymbol{x}^T\boldsymbol{\beta}_j, j = 1, 2, \ldots, k-1$. TREE and RF methods are implemented using R packages and tuned with a built-in cross-validation procedure.

To train the binary weighted SVM, we set the weights as $\pi_m = m/20, m = 1, \ldots, 19$. For kernel WSVMs, we use Gaussian kernel $R(\boldsymbol{x}_1, \boldsymbol{x}_2) = e^{-\|\boldsymbol{x}_1-\boldsymbol{x}_2\|_2^2/\sigma^2}$, where $\| \cdot \|_2$ denotes the l_2-norm. Following Wu, Zhang, and Liu (2010), we select the parameter σ from the set $\{1,2,3,4,5,6\}\sigma_M/4$, where $\sigma_M = \text{Median}\{\| \boldsymbol{x}_i - \boldsymbol{x}_j \|_2 \colon y_i \neq y_j\}$ is the median pairwise Euclidean distance between different classes. The parameter λ is selected from the set $\log_{10}(\lambda) \in \{-8, -7, \ldots, 7, 8\}$ based on GKL. To evaluate the performance of EGKL, we also generate a tuning set of size n to implement EGKL. For real data analysis, it is not feasible to compute the tuning criterion GKL without knowing true class probabilities, but EGKL is its computable approximation. To demonstrate the performance of EGKL as an approxy of GKL, we report the results of both tuning criteria in all the simulation examples.

To evaluate the performance in probability estimation, we generate a test of size $\tilde{n} = 10n$, $\{(\tilde{\boldsymbol{x}}_i, \tilde{y}_i), i = 1, 2, \ldots, \tilde{n}\}$ and compute the following three measures for any probability estimate $\hat{p}_j(\boldsymbol{x}), j = 1, \ldots, k$,

- 1-norm error $\frac{1}{\tilde{n}} \sum_{i=1}^{\tilde{n}} \sum_{j=1}^{k} |\hat{p}_j(\tilde{\boldsymbol{x}}_i) - p_j(\tilde{\boldsymbol{x}}_i)|$
- 2-norm error $\frac{1}{\tilde{n}} \sum_{i=1}^{\tilde{n}} \sum_{j=1}^{k} (\hat{p}_j(\tilde{\boldsymbol{x}}_i) - p_j(\tilde{\boldsymbol{x}}_i))^2$
- EGKL loss $\frac{1}{\tilde{n}} \sum_{i=1}^{\tilde{n}} \sum_{j=1}^{k} p_j(\tilde{\boldsymbol{x}}_i) \log \frac{p_j(\tilde{\boldsymbol{x}}_i)}{\hat{p}_j(\tilde{\boldsymbol{x}}_i)}$.

For each example, we run 100 Monte Carlo simulations and report the average performance measures along with their SEs.

We conduct five simulated studies. In particular, Examples 1 and 2 are two three-class ($k = 3$) cases, Example 3 is a four-class ($k = 4$) case, and Example 4 is a five-class ($k = 5$) case. Different from Examples 1–4, where the BLM method is the oracle procedure, Example 5 considers the scenario where the BLM method is not the oracle. Figure 1 contains scatterplots of the training set for Examples 1–4, and as well as the underlying true (Bayes) classification boundary for each example. Since the method WZL (2010) is computationally infeasible for $k \geq 4$, it is implemented only for $k = 3$.

Example 1 (Three-class, linear case). The data are generated as follows: (i) Generate Y_i uniformly from $\{1, 2, 3\}$; (ii) Given $Y_i = y_i$, the input $\boldsymbol{X}_i = \boldsymbol{x}_i$ follows a bivariate normal $N(\boldsymbol{\mu}(y_i), \boldsymbol{\Sigma})$ with $\boldsymbol{\mu}(y_i) = (\cos(2y_i\pi/3), \sin(2y_i\pi/3))^T$ and $\boldsymbol{\Sigma} = 0.7^2\mathbf{I}_2$, where $\mathbf{I}_2$ is a 2×2 identity matrix. The sample size $n = 400$.

Table 1 summarizes the estimation performance of all the methods. The BLM serves as the oracle since it specifies the underlying model correctly. Overall speaking, the new estimator consistently works best among all and shows substantial gain in estimation accuracy over other methods. The tuning criteria GKL and EGKL give similar performance, suggesting that EGKL is a good computable proxy of GKL in real data analysis. We note that, the TREE method may give zero probability estimates for some $\boldsymbol{x}$ in some classes, yielding infinity ("Inf" in Table 1) for EGKL. Correspondingly, the SEs are not available and therefore denoted as NaN (standing for "Not A Number"). In term of the computing time, it takes on average 8 sec and 568 sec for the new method and WZL(2010), respectively, to get the solutions in Example 1; it takes on average 7 sec and 6360 sec for the new method and WZL(2010), respectively, to get the solutions in Example 2. We observe a significant gain in computation efficiency of the new method compared to WZL(2010), which is consistent to theoretical complexity analysis.

Example 2 (Three-class, nonlinear case). For any $\boldsymbol{x} = (x_1, x_2)^T$, define

$$\begin{aligned} f_1(\boldsymbol{x}) &= -x_1 + 0.1x_1^2 - 0.05x_2^2 + 0.1 \\ f_2(\boldsymbol{x}) &= -0.2x_1^2 + 0.1x_2^2 - 0.2 \\ f_3(\boldsymbol{x}) &= x_1 + 0.1x_1^2 - 0.05x_2^2 + 0.1, \end{aligned}$$

and $p_j(\boldsymbol{x}) = e^{f_j(\boldsymbol{x})}/(\sum_{l=1}^{3} e^{f_l(\boldsymbol{x})})$ for $j = 1, 2, 3$. The data are generated as follows: (i) generate independent X_1 and X_2 uniformly from $[-3, 3]$ and $[-6, 6]$, respectively; (ii) given $\boldsymbol{X} = \boldsymbol{x}$, the label Y takes the value j with probability $p_j(\boldsymbol{x})$ for $j = 1, 2, 3$. The training data size $n = 1000$. The BLM can be regarded as the oracle.

For the proposed methods, there are three different ways to achieve nonlinear classification. The first way is through basis expansion, by expanding the input $\boldsymbol{x} = (x_1, x_2)^T$ into a five-dimensional vector $\tilde{\boldsymbol{x}} = (x_1^2, x_2^2, x_1x_2, x_1, x_2)^T$, and then solves (3) with linear SVMs using $\tilde{\boldsymbol{x}}$ as the input vector. The second way is to use kernel SVMs with the Gaussian kernel $\mathbf{K}(\boldsymbol{x}_1, \boldsymbol{x}_2) = e^{-\|\boldsymbol{x}_1-\boldsymbol{x}_2\|_2^2/\sigma^2}$ with $\sigma^2 = 2\sigma_M^2$, where σ_M^2 is the median of the Euclidean distances from positive examples to negative examples. The third way is to use kernel SVMs associated with Spline kernel given by (Gu 2002). Table 1 (the second panel) summarizes the numerical results for the new methods implemented with basis expansion. In term of all three performance measures, the proposed estimators consistently outperform other competing methods.

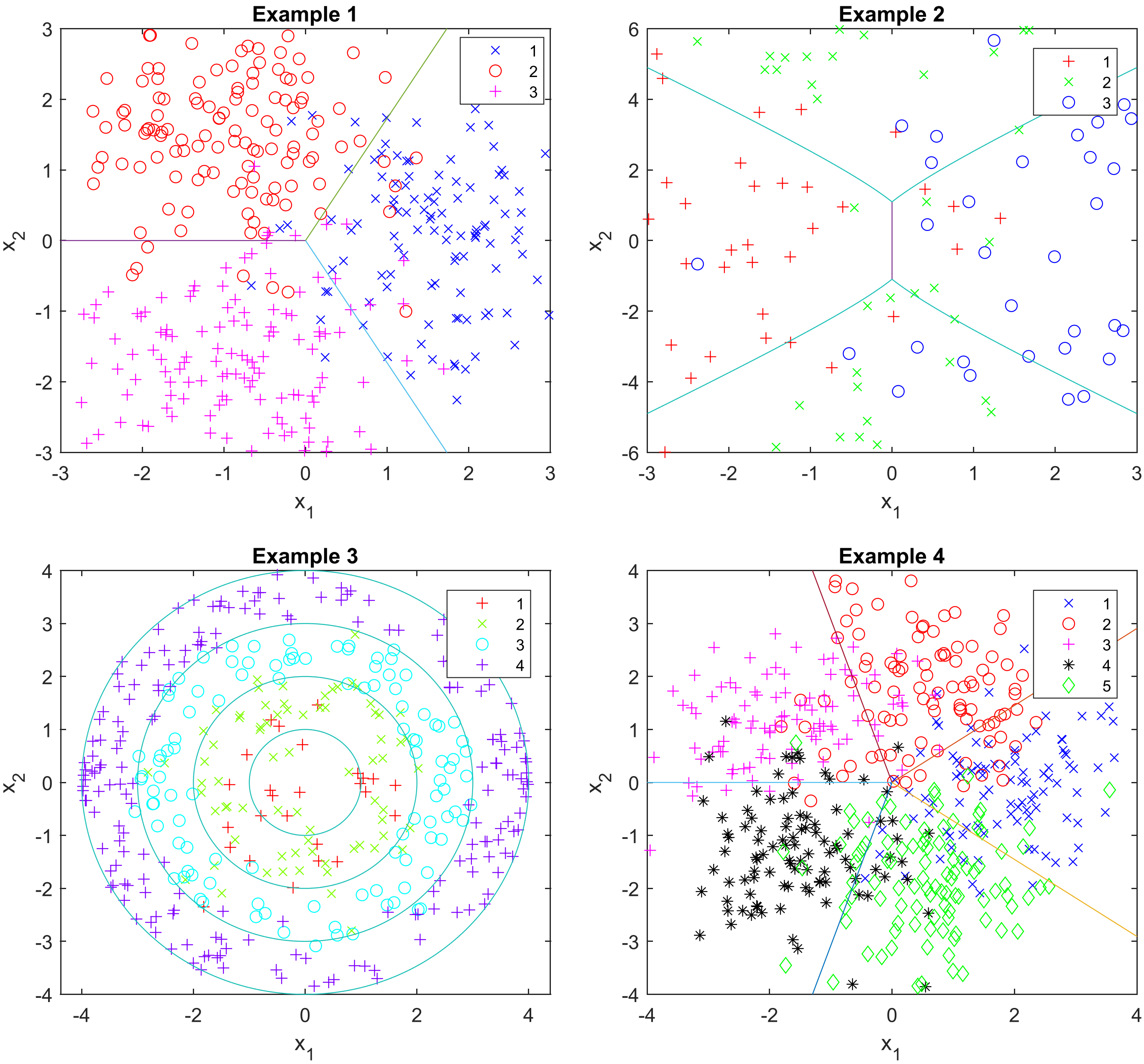

Figure 1. This figure contain the scatterplots of the training datasets for Examples 1–4, along with the true class boundaries (Bayes rules). In each plot, different colors (or symbols) denote training data points from different classes.

Table 1. Probability estimation results for Examples 1 and 2.

	New-GKL	New-EGKL	WZL(2010)	XW(2013)	KMLR	TREE	Oracle
			Example 1				
1-norm	10.1 (1.6)	10.7 (1.5)	11.0 (2.3)	32.4 (1.9)	53.6 (2.6)	27.5 (3.3)	6.2 (1.9)
2-norm	0.6 (0.2)	0.6 (0.2)	0.9 (0.3)	6.4 (1.2)	12.4 (1.1)	6.0 (1.2)	0.4 (0.2)
EGKL	2.1 (0.5)	2.2 (2.2)	2.6 (0.8)	12.5 (1.0)	24.8 (1.7)	Inf (NaN)	0.8 (0.5)
			Example 2				
	New-GKL	Mew-EGKL	WZL(2010)	XW(2013)	KMLR	TREE	Oracle
1-norm	24.5 (4.7)	24.1 (6.2)	36.4 (4.4)	31.1 (1.1)	59.3 (4.8)	60.1 (10.6)	19.4 (5.2)
2-norm	4.1 (1.6)	5.0 (2.6)	7.9 (2.1)	5.3 (3.6)	16.4 (2.3)	22.2 (4.8)	3.2 (1.9)
EGKL	9.1 (2.9)	10.9 (10.6)	13.6 (2.9)	8.8 (0.5)	28.9 (3.2)	Inf (NaN)	9.1 (9.2)

NOTE: The table compares the new method with four other methods: WZL(2010), XW(2013), kernel logistic regression (KMLR), and TREE method, in terms of three performance measures: 1-norm error, 2-norm error, and EGKL. The new method is tuned with GKL and EGKL, respectively. The oracle method is implemented by the BLM. The numbers in parentheses are SEs. The table suggests the proposed methods overall outperform other methods under comparison in these two examples and are the closest to the oracle method.

Table 2. Average probability estimation results for Examples 3 and 4.

Example 3 (three-class example)

	New-GKL	New-EGKL	XW(2013)	RF	KMLR	TREE	Oracle
1-norm	25.8 (1.6)	25.8 (1.6)	36.3 (2.5)	29.8 (1.8)	132.0 (8.2)	43.9 (5.4)	11.4 (2.1)
2-norm	5.3 (0.8)	5.3 (0.8)	7.4 (0.9)	8.6 (1.0)	86.1 (12.4)	18.3 (2.4)	1.7 (0.6)
EGKL	12.8 (1.4)	12.8 (1.3)	14.1 (1.5)	Inf (NaN)	247.8 (91.0)	Inf (NaN)	4.6 (2.1)

Example 4 (five-class example)

	New-GKL	New-EGKL	XW(2013)	RF	KMLR	TREE	Oracle
1-norm	15.3 (1.6)	16.0 (2.1)	39.1 (1.7)	41.5 (1.8)	105.6 (1.5)	40.1 (3.2)	7.3 (1.7)
2-norm	1.0 (0.2)	1.1 (0.3)	5.8 (1.2)	9.0 (0.8)	32.9 (0.9)	7.8 (1.2)	0.3 (0.1)
EGKL	3.6 (0.4)	4.0 (0.4)	15.0 (0.9)	Inf (NaN)	76.4 (2.2)	Inf (NaN)	0.6 (0.3)

NOTE: The table compares the proposed method with three other methods: XW(2013), kernel logistic regression (KMLR), RF, and TREE method, in terms of three performance measures: 1-norm error, 2-norm error, and EGKL. The new method is tuned with GKL and EGKL, respectively. The oracle method is implemented by the BLM. The numbers in parentheses are SEs. The table suggests the proposed methods overall outperform other methods under comparison in these two examples and are the closest to the oracle method.

Table 3. Average probability estimation results for Example 5.

Example 5

	New-GKL	New-EGKL	WZL (2010)	KMLR	TREE	BLM
1-norm	18.3 (2.6)	19.8 (3.6)	21.8 (2.2)	63.1 (1.9)	24.4 (3.3)	31.0 (1.1)
2-norm	0.3 (0.0)	0.3 (0.0)	4.5 (1.0)	16.7 (0.9)	7.7 (1.4)	6.9 (0.3)
EGKL	7.0 (1.4)	7.8 (7.7)	11.8 (2.6)	31.8 (1.4)	Inf (NaN)	12.7 (0.4)

NOTE: The table compares the proposed method with four existing methods: WZL(2010), kernel logistic regression (KMLR), TREE, and BLM methods, in terms of three measures: 1-norm error, 2-norm error, and EGKL. The new method is tuned with GKL and EGKL, respectively. The numbers in parentheses are SEs. The results suggest that the proposed methods consistently give the best prediction performance among all the methods, including the BLM.

Example 3 (Four-class, nonlinear case). For $\boldsymbol{x} = (x_1, x_2)^T$, define

$$f_1(\boldsymbol{x}) = -|x_1^2 + x_2^2|,\ \ f_2(\boldsymbol{x}) = -|x_1^2 + x_2^2 - 1.5^2|,$$
$$f_3(\boldsymbol{x}) = -|x_1^2 + x_2^2 - 2.5^2|,\ \ f_4(\boldsymbol{x}) = -|x_1^2 + x_2^2 - 3.5^2|,$$

and let $p_j(\boldsymbol{x}) = \exp f_j(\boldsymbol{x})/[\sum_{l=1}^4 \exp(f_l(\boldsymbol{x}))]$ for $j = 1, 2, 3, 4$. We generate the data as follows: (i) generate the two dimensional predictor $\boldsymbol{X}$ uniformly for the disc $\{\boldsymbol{x} : x_1^2 + x_2^2 \le 16\}$; (ii) given $\boldsymbol{X} = \boldsymbol{x}$, the label Y takes the value j with probability $p_j(\boldsymbol{x})$ for $j = 1, 2, 3, 4$. In this setting, the BLM model assumption holds, so the BLM procedure can be regarded as the oracle. The training sample size $n = 400$. Table 1 (the top panel) summarizes the performance of all the methods under comparison, evaluated on a test set of size $\tilde{n} = 10n$.

For the proposed method, there are three different ways to achieve nonlinear classification. The first way is through basis expansion, by expanding the input $\boldsymbol{x} = (x_1, x_2)^T$ into a five-dimensional vector $\tilde{\boldsymbol{x}} = (x_1^2, x_2^2, x_1x_2, x_1, x_2)^T$ and then implementing the linear SVM with $\tilde{\boldsymbol{x}}$ as the input vector. The second way is to use kernel SVMs with the Gaussian kernel $\mathbf{K}(\boldsymbol{x}_1, \boldsymbol{x}_2) = e^{-||\boldsymbol{x}_1 - \boldsymbol{x}_2||_2^2/\sigma^2}$ with $\sigma^2 = 2\sigma_M^2$, where σ_M^2 is the median of the Euclidean distances from positive examples to negative examples. The third way is to use kernel SVMs associated with the spline kernel given by Gu (2013). We report the results using the basis expansion technique, since those based on Gaussian kernel and the spline kernel are quite similar.

Similar to the previous examples, we observe that the new estimator consistently works best among all and shows substantial gain in estimation accuracy over other methods. The tuning criteria GKL and EGKL give similar performance, suggesting that EGKL is a good computable proxy of GKL. We note that, the TREE method may give zero probability estimates for some $\boldsymbol{x}$ in some classes, yielding infinity ("Inf" in Table 1) for EGKL. Correspondingly, the SEs are not available and therefore denoted as NaN (standing for "Not A Number").

Example 4 (Five-class, linear case). This is a five-class linear example designed to illustrate the new method's ability to handle an even larger number of classes. The data are generated as follows: (i) generate the label Y uniformly from $\{1, 2, 3, 4, 5\}$; (ii) Given $Y = y$, generate $\boldsymbol{X}$ from the bivariate normal distribution $N(\boldsymbol{\mu}(y), \Sigma)$, where $\boldsymbol{\mu}(y) = (\cos(2y\pi/5), \sin(2y\pi/5))^T$, $\Sigma = \mathbf{I}_2$, and $\mathbf{I}_2$ is a 2×2 identity matrix. The training sample size $n = 500$. Table 2 (the bottom panel) shows that the proposed probability estimator gives the best estimation performance among all the methods under comparison. With regard to tuning, GKL and its computable proxy EGKL give very similar performance.

Example 5 (Three-class, BLM is not the oracle). In the above examples, the BLM method is the oracle and hence gives the best performance. In this example, we consider a scenario where the BLM model assumption does not hold any more. Assume $\boldsymbol{X}$ is uniformly sampled from a disc $\{\boldsymbol{x} : x_1^2 + x_2^2 \le 100\}$. Define $h_1(\boldsymbol{x}) = -5x_1\sqrt{3} + 5x_2$, $h_2(\boldsymbol{x}) = -5x_1\sqrt{3} - 5x_2$ and $h_3(\boldsymbol{x}) = 0$. Then consider the transformation $f_j(\boldsymbol{x}) = \Phi^{-1}(T_2(h_j(\boldsymbol{x})))$ for $j = 1, 2, 3$, where $\Phi(\cdot)$ is the cdf of the standard normal distribution and $T_2(\cdot)$ is the cdf of t_2 distribution. Then we set the probability functions as $p_j(\boldsymbol{x}) = \exp f_j(\boldsymbol{x})/(\sum_{l=1}^3 \exp(f_l(\boldsymbol{x})))$ for $j = 1, 2, 3$, and let $n = 400$. Table 3 shows that the proposed methods consistently give the best prediction performance among all the methods under comparison, including the BLM. The tuning criteria GKL and its computable proxy EGKL give very similar performance.

Table 4. Average misclassification error rates on benchmark data examples.

	Data information			Methods				
Dataset	Class #	n	$\tilde{n}$	New	XW(2013)	RF	TREE	BLM
Zip(3,6,9)	k=3	1966	513	1.2 (0.3)	3.2 (0.1)	1.4 (0.2)	8.2 (1.2)	3.3 (2.6)
Zip(full)	k=10	7291	2007	7.1 (0.7)	- (-)	5.9 (0.4)	27.8 (1.2)	58.6 (2.8)
Ecoli	k=4	222	110	14.7 (2.3)	23.5 (4.0)	14.4 (2.9)	18.9 (4.7)	29.8 (2.2)
Yeast	k=5	989	495	37.3 (0.8)	47.6 (2.4)	36.1 (1.7)	41.2 (2.3)	39.0 (2.1)

NOTE: The table contains the data information and the analysis results for the four benchmark datasets. The first column contains four dataset names. The next three columns contain the dataset information: k is the number of classes, n is the training set size, and $\tilde{n}$ is the test set size. The remaining columns compares the new method with XW(2013), RF, tree, and BLM, on four datasets. The results suggest that the proposed method and RF are among the top two classifiers and their performance is overall similar in these four studies.

5. Benchmark Data Analysis

In this section, we illustrate performance of the new estimator on real-world data. Four real datasets are considered, representing two scenarios: $d < n$ and high dimensional case $d > n$.

5.1. Low-Dimensional Cases ($d < n$)

We apply the new methods to three benchmark datasets: zip code, ecoli data, and Yeast data. The data information is summarized in Table 4.

The zip code data consist of 7291 hand-written zip code digits automatically scanned from envelopes by the U.S. Postal Service. As the original scanned digits have different sizes and orientations, we desalinate and normalize the images, resulting in 16 × 16 gray-scale images (Le Cun et al. 1990). The test set size is 2007. Since the dataset is quite large and has 10 classes, we start with a small classification problem by distinguishing three classes (digits 3, 6, and 9). Then we use the whole dataset to evaluate the methods on the 10-class problem.

For the E. coli dataset, the goal is to predict the cellular localization sites of E. coli proteins (Horton and Nakai 1996). The original data consist of eight different cellular sites. Since some classes have fewer observations than others, we merge some classes and form a four-class problem.

The goal of the Yeast dataset analysis is to determine the cellular localization of the yeast proteins (Horton and Nakai 1996). There are 10 different sites, which include: CYT (cytosolic or cytoskeletal); NUC (nuclear); MIT (mitochondrial); ME3 (membrane protein, no N-terminal signal); ME2 (membrane protein, uncleaved signal); ME1 (membrane protein, cleaved signal); EXC (extracellular); VAC (vacuolar); POX (peroxisomal); and ERL (endoplasmic reticulum lumen). After combining some small classes, we end up with a five-class classification problem.

For zip code datasets, we split the training data into two halves (one half for training, and the other for tuning) 10 times, and report the average error on the test set. For the E. coli and Yeast datasets, we randomly split the entire data equally as the training, tuning, and testing sets for 10 times, and report the average test error (with SEs in parentheses). Table 4 suggests that the new method and RF are overall among the top two classifiers and they perform quite similarly in these studies.

5.2. High-Dimensional Case: $d > n$

One advantage of the new method is its ability to fit nonlinear classifiers for high dimensional data with $d > n$, thanks to the kernel trick. We apply the new method to the children cancer dataset of Khan et al. (2001). The goal is to classify SRBCTs of childhood, based on cDNA gene expression profiles, into four classes: NB, RMS, non-Hodgkin lymphoma (NHL), and the Ewing family of tumors (EWS). The 2308 gene profiles are provided at *http://research.nhgri.nih.gov/microarray/Supplement/*, and the training size is 63 (containing 23 EWS, 8 BL, 12 NB, 20 RMS) and the test set size is 20 (containing 6 EWS, 3 BL, 6 NB, 5 RMS).

We first standardize the data by linear transformations. Specifically, we using the following formula to standardize the gene expression value $\tilde{x}_{gi}$ for the gth gene of subject i:

$$x_{gi} = \frac{\tilde{x}_{gi} - \frac{1}{n}\sum_{l=1}^{n}\tilde{x}_{gl}}{sd(\tilde{x}_{g1}, \dots, \tilde{x}_{gj})}.$$

After standardization, all genes are ranked by their marginal relevance to the class label, using the criterion suggested by (Dudoit et al. 2002). Specifically, the relevance measure of gene g is calculated as the ratio of between-class sum of squares to within-class sum of squares as follows:

$$R(g) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{k} I(y_i = j)(\bar{x}_{g\cdot}^{(j)} - \bar{x}_{g\cdot})^2}{\sum_{i=1}^{n}\sum_{j=1}^{k} I(y_i = j)(x_{gi} - \bar{x}_{g\cdot}^{(j)})^2},$$

where n is the training size, $\bar{x}_{g\cdot}^{(j)}$ is the average expression of gene g for observations in class j, and $\bar{x}_{g\cdot}$ is the overall mean expression level of gene g in the training set. We select top 100 and bottom 100 genes as the covariates based on the relevance measure R. Figure 2 plots the estimated class probabilities of NB (green), RMS (blue), BL (red), and EWS (purple), given by the proposed method on 20 testing points. The method makes only one mistake, by misclassifying one NB sample to RMS. It has been found in the literature that, gene expressions carry sufficient information to differentiate cancer subtypes very well. It explains why most methods can achieve high accuracy (Zhu et al. 2004; Zhang et al. 2008).

6. Concluding Remarks

We propose a simple yet effective approach to estimating multiclass probabilities using SVMs. The main advantages of the new method are its flexibility, high prediction accuracy, and computational efficiency for large k. Its divide-and-conquer feature makes the algorithm enjoy parallel computing. We also establish the consistency of the new probability estimator. Our choice assures the consistency of $\hat{p}_j$'s, provided $q_{j|(j,j')}$'s being estimated

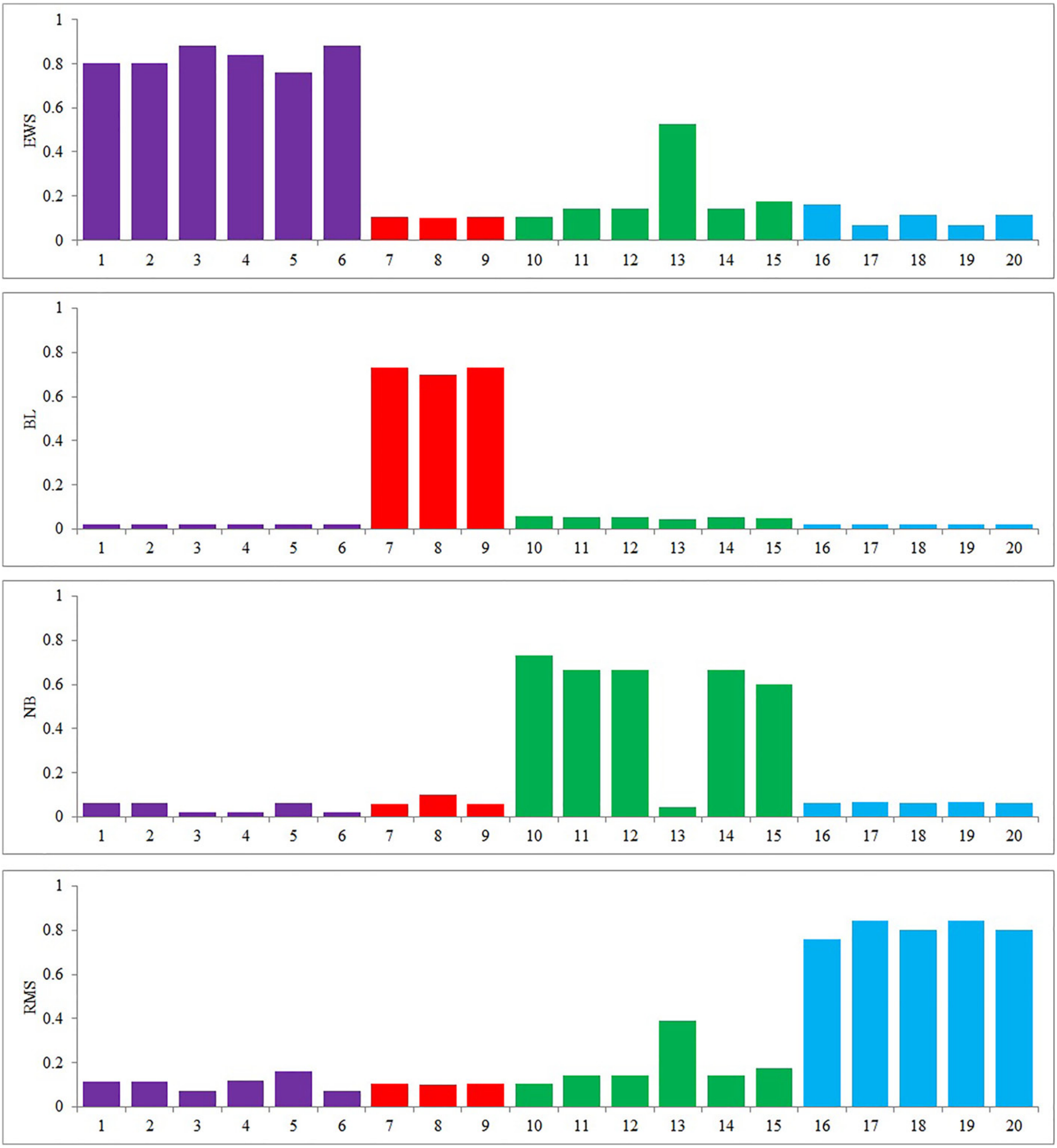

Figure 2. This figure plots the estimated probabilities (shown as the bar heights) for the SRBCT test set. From the top to the bottom, the four panels present the estimated class probabilities of one subclass test samples: EWS, BL, NB, and RMS, respectively. Four colors (purple, red, green, blue), respectively, represent the estimated probabilities of belonging to class EWS, BL, NB, RMS, by the proposed method.

consistently. Furthermore, one-vs-one subproblems generally involves small sample sizes and demand low computational cost. When the number of subproblems $\binom{k}{2}$ is large, we suggest using parallel computing.

One interesting yet challenging topic for future research is how to conduct variable selection under the proposed estimation framework. Though it is straightforward to implement variable selection for each pairwise classification problem, it is unclear how to combine the selected variable sets as different pairwise classification may select different important predictors. It would be very useful to come up with an optimal way of aggregating the selected variables.

Supplementary Materials

Code: Contains the MATLAB source code (to implement the proposed method, and generate data for simulated examples).

Funding

This research was supported in part by National Science Foundation grants CCF-1740858 and DMS-1055210 and NSFC 11571009.

References

Agresti, A., and Coull, B. A. (1998), "Approximate Is Better Than 'Exact' for Interval Estimation of Binomial Proportions," *The American Statistician*, 52, 119–126. [586]

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984), *Classification and Regression Trees*, Belmont, CA: Wadsworth Publishing Company. [586,590]

Burges, C. (1998), "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery*, 2, 121–167. [586]

Cortes, C., and Vapnik, V. (1995), "Support-Vector Networks," *Machine Learning*, 20, 273–297. [586]

Cristianini, N., and Shawe-Taylor, J. (2000), *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*, Cambridge, UK: Cambridge University Press. [586]

Dudoit, S., Fridlyand, J. and Speed, T. P. (2002), "Comparison of Discrimination Methods for the Classification of Tumors Using Gene Expression Data," *Journal of the American Statistical Association*, 97, 77–87. [593]

Gu, C. (2002), Smoothing spline ANOVA models, New York: Springer-Verlag. [590]

——— (2013), *Smoothing Spline ANOVA Models*, New York: Springer. [592]

Hastie, T., and Tibshirani, R. (1998), "Classification by Pairwise Coupling," in *Advances in Neural Information Processing Systems* (Vol. 10), eds. M. I. Jordan, M. J. Kearns, and A. S. Solla, Cambridge, MA: MIT Press. [587]

Horton, P., and Nakai, K. (1996), "A Probabilistic Classification System for Predicting the Cellular Localization Site of Protein," in *Proceedings of the International Conference on Intelligent Systems for Molecular Biology* (Vol. 4), pp. 109–115. [593]

Huang, H., Liu, Y., Du, Y., Perou, C., Hayes, N., Todd, M., and Marron, S. (2013), "Multiclass Distance Weighted Discrimination," *J. Mach. Learn. Res.*, 22, 953–969. [586]

Khan, J., Wei, J. S., Ringner, M., Saal, L. H., Ladanyi, M., Westermann, F., Berthold, F., Schwab, M., Antonescu, C. R., Peterson, C., and Meltzer, P. S. (2001), "Classification and Diagnostic Prediction of Cancers Using Gene Expression Profiling and Artificial Neural Networks," *Nature Medicine*, 7, 673–679. [586,593]

Kimeldorf, G., and Wahba, G. (1971), "Some Results on Tchebycheffian Spline Functions," *Journal of Mathematical Analysis and Applications*, 33, 82–95. [587,589]

Le Cun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1990), "Back-Propagation Applied to Handwritten Zipcode Recognition," *Neural Computation*, 1, 541–551. [593]

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. [586]

Lin, Y. (2002), "Support Vector Machines and the Bayes Rule in Classification," *Data Mining and Knowledge Discovery*, 6, 259–275. [586,587]

Liu, Y. (2007), "Fisher Consistency of Multicategory Support Vector Machines," in *Eleventh International Conference on Artificial Intelligence and Statistics*, pp. 289–296. [586]

Liu, Y., and Shen, X. (2006), "Multicategory Psi-Learning," *Journal of the American Statistical Association*, 101, 500–509. [586]

Liu, Y., and Yuan, M. (2011), "Reinforced Multicategory Support Vector Machine," *Journal of Computational and Graphical Statistics*, 20, 901–919. [586]

Platt, J. (1999), "Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods," in *Advances in Large Margin Classifiers*, Cambridge, MA: MIT Press. [587]

Qiao, X., and Liu, Y. (2009), "Adaptive Weighted Learning for Unbalanced Multicategory Classification," *Biometrics*, 65, 159–168. [586]

Tu, X., and Wang, J. (2013), "An Efficient Model-Free Estimation for Multiclass Conditional Probability," *Journal of Statistical Planning and Inference*, 143, 2079–2088. [587,590]

Van Calster, B., Luts, J., Suykens, J., Condous, G., Bourne, T., Timmerman, D., and Van Huffel, S. (2007), "Comparing Methods for Multi-class Probabilities in Medical Decision Making Using LS-SVMs and Kernel Logistic Regression," in *Artificial Neural Networks*, Berlin, Heidelberg: Springer, pp. 139–148. [587]

Vapnik, V. (1998), *Statistical Learning Theory*, New York: Wiley. [586]

Wahba, G. (1990), "Spline Models for Observational Data," in *CBMS-NSF Regional Conference Series*, Philadelphia, PA: Society for Industrial and Applied Mathematics (SIAM). [587,589]

Wang, J. (2013), "Boosting the Generalized Margin in Cost-Sensitive Multiclass Classification," *Journal of Computational and Graphical Statistics*, 22, 178–192. [586]

Wang, L., and Shen, X. (2007), "On L_1-Norm Multiclass Support Vector Machines," *Journal of the American Statistical Association*, 102, 583–594. [586]

Wang, J., Shen, X., and Liu, Y. (2008), "Probability Estimation for Large Margin Classifiers," *Biometrika*, 95, 149–167. [587,588,589]

Weston, J., and Watkins, C. (1999), "Support Vector Machines for Multiclass Pattern Recognition," in *Proceedings of 7th European Symposium on Artificial Neural Networks*, pp. 219–224. [586]

Wu, T.-F., Lin, C.-J., and Weng, R. C. (2004), "Probability Estimates for Multi-class Classification by Pairwise Coupling," *Journal of Machine Learning Research*, 5, 975–1005. [587]

Wu, Y., Zhang, H. H., and Liu, Y. (2010), "Robust Model-Free Multiclass Probability Estimation," *Journal of the American Statistical Association*, 105, 424–436. [587,588,590]

Zhang, C., and Liu, Y. (2013), "Multicategory Large-Margin Unified Machines," *Journal of Machine Learning Research*, 14, 1349–1386. [586]

Zhang, H. H., Liu, Y., Wu, Y., and Zhu, J. (2008), "Variable Selection for the Multicategory SVM via Adaptive Sup-Norm Regularization," *Electronical Journal of Statistics*, 2, 149–167. [593]

Zhang, T. (2004), "Statistical Analysis of Some Multi-category Large Margin Classification Methods," *Journal of Machine Learning Research*, 5, 1225–1251. [586]

Zhu, J., and Hastie, T. (2005), "Kernel Logistic Regression and the Import Vector Machine," *Journal of Computational and Graphical Statistics*, 14, 185–205. [586,590]

Zhu, J., Rosset, S., Hastie, T., and Tibshirani, R. (2004), "1-Norm Support Vector Machines," in *The Annual Conference on Neural Information Processing Systems* (Vol. 16). [586,593]

Zhu, J., Zou, H., Rosset, S., and Hastie, T. (2009), "Multi-class Adaboost," *Statistics and Its Interface*, 2, 349–360. [586]

Zou, H., Zhu, J., and Hastie, T. (2008), "New Multi-category Boosting Algorithms Based on Multi-category Fisher-Consistent Losses," *Annals of Applied Statistics*, 2, 1290–1306. [586]