

Better Multiclass Classification via a Margin-Optimized Single Binary Problem

Ran El-Yaniv^a, Dmitry Pechyony^{a,*}, Elad Yom-Tov^b

^aDepartment of Computer Science, Technion - Israel Institute of Technology, Haifa 32000, Israel

^bIBM Haifa Research Lab, Haifa 31905, Israel

Abstract

We develop a new multiclass classification method that reduces the multiclass problem to a single binary classifier (SBC). Our method constructs the binary problem by embedding smaller binary problems into a single space. A good embedding will allow for large margin classification. We show that the construction of such an embedding can be reduced to the task of learning linear combinations of kernels. We provide a bound on the generalization error of the multiclass classifier obtained with our construction and outline the conditions for its consistency. Our empirical examination of the new method indicates that it outperforms one-vs-all, all-pairs and the error-correcting output coding scheme at least when the number of classes is small.

Key words: multiclass classification, support vector machines, multiple kernel learning

1. Introduction

The widespread practice of employing support vector machines (SVMs) in applications provides a major incentive for the ongoing study of the *multiclass-to-binary reduction* problem to enable the use of binary SVMs for multiclass problems. However, despite numerous ideas on how SVMs can be applied to multiclass classification, the understanding of multiclass reductions appears to be somewhat limited, both theoretically and empirically. The confusion surrounding this problem has only increased with the availability of increasingly clever and sophisticated solutions, whose authors indicate that there is much to gain by using their approaches, but often without providing sufficient comparisons to other available methods.

Currently, the simplest multiclass-to-binary reduction method is the ‘one-vs-all’ (OVA). Two other

well-known reductions are the ‘all-pairs’ approach [11] (a.k.a. ‘one-vs-one’) and the ‘error-correcting output coding’ (ECOC) framework pioneered by [20] and [8]. One of the first comprehensive comparisons of multiclass reduction methods is reported in [13]. The authors claimed that the all-pairs approach is superior to other methods. Rifkin and Klautau’s prominent paper [19] later presented an in-depth critical assessment of many previous multiclass papers (including [13]). The authors stated that OVA is not inferior to all-pairs and ECOC, provided that adequate efforts are devoted to hyperparameter tuning. Despite the compelling arguments made in [19] for OVA, there remains an ongoing debate on the relative effectiveness of these three methods.

A lesser-known approach for solving multiclass problems via binary classification is the *single binary classifier* reduction (henceforth, SBC), which is any multiclass method that relies on a single, standard *binary* (soft) classifier. SBC reductions can be obtained by embedding the original problem in a higher-dimensional space consisting of the original features, as well as one or more other dimensions determined by fixed vectors,

* Corresponding author. Tel.: +972-48294325; fax: +972-48293900.

Email addresses: rani@cs.technion.ac.il (Ran El-Yaniv), pechyony@cs.technion.ac.il (Dmitry Pechyony), yomtov@il.ibm.com (Elad Yom-Tov).

termed here *extension features*. This embedding is implemented by replicating the training set points so that a copy of the original point is concatenated with each of the extension features' vectors. The *binary* labels of the replicated points are set to maintain a particular structure in the extended space. This construction results in an instance of an artificial binary problem, which is fed to a binary learning algorithm that outputs a single soft binary classifier. To classify a new point, the point is replicated and extended similarly and the resulting replicas are fed to the soft binary classifier, which generates a number of *signals*, one for each replica. The class is determined as a function of these signals.

In this paper, we propose a new type of kernel SBC (henceforth SBC-KERNEL) where, rather than using explicit extensions in feature space, we utilize implicit kernel transformations in Hilbert space. A different transformation is used for each class and these transformations are constructed to increase the margin of the resulting binary problem in which the entire multiclass problem is embedded (see details below). This SBC-KERNEL method is posed and derived as a kernel optimization problem using the SVM objective function.

We present a comparative study of the proposed SBC-KERNEL method, OVA, all-pairs, and ECOC, as well as three previous SBC methods. These results demonstrate impressive performance of SBC-KERNEL relative to the other algorithms. We also provide two theoretical results about SBC reductions. We observe that the recent risk bound for kernel machines with 'learned kernels' [21] can be extended to our setting, showing that SBC-KERNEL indeed generalizes well in cases where both the number of classes and the empirical error are small.

Consistency issues in the context of multiclass reductions have been raised by [16]. For example, most of the multiclass loss functions are not necessarily consistent (under empirical risk minimization strategy) unless certain conditions on the learning problem are met [25]. We prove the informal claim, made by [25], that the same conditions guarantee the consistency of the loss function implied by the SBC reduction.

2. On Some Known multiclass to Binary Reductions

2.1. General Reductions

Let $S = \{(x_i, y_i)\}_{i=1}^m$ be a training set of m examples, where x_i are points in some d -dimensional space \mathcal{X} and each y_i is a label in $\mathcal{Y} = \{1, \dots, c\}$. A *multiclass classifier* h is any function $h : \mathcal{X} \rightarrow \mathcal{Y}$.

Our goal in multiclass classification is to generate h so as to minimize its 0/1-loss average error over out-of-sample examples. We are concerned with *multiclass-to-binary reductions*, which are methods that solve multiclass classification through the use of binary classifiers such as standard SVMs. Three well-known and widely-used multiclass-to-binary reductions are one-vs-all (OVA) [9], all-pairs [11] and the error-correcting output coding (ECOC) framework [8]. These reductions decompose the multiclass problem to a number of binary problems.

Another family of reductions is the following single binary classification (SBC) reduction: Let M be a $c \times \ell$ matrix of *feature extensions*; the i th row of M is denoted by M_i . In the preprocessing stage, we construct c different copies of each training example x_i , where the r th copy of x_i is $z_{i,r} = x_i \circ M_r$, which is the extension (or concatenation) of the row vector x_i with the row vector M_r . The resulting instances, $\{z_{i,r}\}$, $i = 1, \dots, m$, $r = 1, \dots, c$, are assigned binary labels as follows: for each i and r , the instance $z_{i,r}$ is labeled by $y_{i,r} = +1$ iff $y_i = r$ (i.e., the original label y_i of x_i is r). Otherwise, $z_{i,r}$ is labeled by $y_{i,r} = -1$. The resulting binary-labeled training set $S' = \{(z_{i,r}, y_{i,r})\}$ is of size cm and each instance (excluding the label) has $d + \ell$ dimensions. In the second stage of binary learning, we apply a standard learning algorithm (e.g., SVM) on the training set S' and the outcome is a soft binary classifier h_2 . To determine a label (in \mathcal{Y}) of a new instance x , we generate c copies of x , where the r th copy is $z_r = x \circ M_r$. The label we predict is $\arg\max_r h_2(z_r)$. As far as we know, the first documented SBC reduction is the *Kessler construction* [9] (Sec. 5.12.1), which was developed for the linearly separable case.

The SBC reduction has the following interpretation. Let $\mathcal{P}_j = \{(z_{i,j}, y_{i,j})\}_{i=1}^m$ be the set of the j th copies of each example. According to the definition of SBC, $y_{i,j} = 1$ if $y_i = j$; otherwise $y_{i,j} = -1$. Hence \mathcal{P}_j is one of the binary problems solved by the one-vs-all reduction. Since $S' = \cup_j \mathcal{P}_j$, SBC reduction tries to solve all one-vs-all binary problems simultaneously, using a single binary classifier. Using the feature extension matrix M , we separate and align the binary problems in a larger space to allow for their accurate simultaneous solution by a single binary classifier.

Unlike the SBC approach, *single-machine* constructions typically modify the standard SVM optimization problem to include c separate soft classifiers simultaneously, one for each class. See Section 3.1 in [19] for a survey of early approaches to single-machine SVM construction. We note here that the recent approaches to learning with structured outputs (e.g., see [23]) and the

exponential family formalism of [5] also fall into the category of single machine SVM constructions. While these papers do not include any in-depth treatment of the multiclass to binary problem, the instantiations of these approaches to multiclass problems have some similarities to our developments. We discuss this issue further in Section 3.

2.2. Specific SBC Reductions

A special case of the SBC reduction is the method proposed in [2], where the matrix M is taken to be the $c \times c$ identity matrix. We refer to this reduction as SBC-IDENTITY.

Example 2.1 Suppose that $\mathcal{Y} = \{1, 2, 3\}$ and the training set consists of the following four labeled examples: $\{(x_1, 1), (x_2, 2), (x_3, 3), (x_4, 2)\}$. Then, by the definition of SBC-IDENTITY, the feature extension matrix is

$$M = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

and the resulting labeled training set for the binary SBC-IDENTITY problem is

$$\begin{aligned} z_{1,1} &= x_1 \circ (1, 0, 0), & y_{1,1} &= +1 \\ z_{1,2} &= x_1 \circ (0, 1, 0), & y_{1,2} &= -1 \\ z_{1,3} &= x_1 \circ (0, 0, 1), & y_{1,3} &= -1 \\ z_{2,1} &= x_2 \circ (1, 0, 0), & y_{2,1} &= -1 \\ z_{2,2} &= x_2 \circ (0, 1, 0), & y_{2,2} &= +1 \\ z_{2,3} &= x_2 \circ (0, 0, 1), & y_{2,3} &= -1 \\ z_{3,1} &= x_3 \circ (1, 0, 0), & y_{3,1} &= -1 \\ z_{3,2} &= x_3 \circ (0, 1, 0), & y_{3,2} &= -1 \\ z_{3,3} &= x_3 \circ (0, 0, 1), & y_{3,3} &= +1 \\ z_{4,1} &= x_4 \circ (1, 0, 0), & y_{4,1} &= -1 \\ z_{4,2} &= x_4 \circ (0, 1, 0), & y_{4,2} &= +1 \\ z_{4,3} &= x_4 \circ (0, 0, 1), & y_{4,3} &= -1 . \end{aligned}$$

Another special case of the SBC reduction is the SBC-SINGLE method, obtained by taking the column $(1, 2, \dots, c)^T$ as the feature extensions matrix. In other words, a single feature is concatenated to the data such that the r th ‘‘replication’’ of x_i , is $z_{i,r} = x_i \circ r$. Binary labels are assigned to this data exactly as described above.

SBC-SINGLE is also a special case of the general ‘single-call’ SBC reduction of [1]. Here, given a $c \times \ell$ (ECOC) coding matrix M , each training example (x_i, y_i) is replicated ℓ times to create ℓ new training examples of the form $((x_i, s), M(y_i, s))$, where $M(y_i, s)$, $1 \leq s \leq \ell$, is a binary label. Using this training set, one induces a binary classifier denoted by h_2 . To classify a new point x , we similarly replicate it ℓ times, $z_i = x \circ i$, $i = 1, \dots, \ell$, and apply h_2 on each of the ℓ instances. The resulting vector of (soft) classifications $(h_2(z_1), \dots, h_2(z_\ell))$ is matched to the closest codeword (row) in M to determine the label. The matching can be done using a Euclidian norm (if h_2 is a soft classifier) or a Hamming distance (if h_2 is a hard classifier). We term this SBC reduction SBC-ECOC. Notice that SBC-SINGLE is a special case of SBC-ECOC applied with a matrix M that is the $c \times c$ identity matrix and thus without error-correction properties. The SBC-ECOC construction adds a single attribute to each example and replicates it ℓ times. This is in contrast to the family of SBC reductions we describe above, which also use a (feature extension) matrix, extend each example by ℓ additional binary features, and replicate each example c times.

3. Learning SBC Kernel Reductions

Rather than the explicit feature extensions in SBC reductions described in Section 2, we propose a general approach that utilizes arbitrary class mappings. Similar to ‘standard’ SBC reductions, each training example x_i is replicated c times and the r th copy of x_i is $z_{i,r} = \phi_r(x_i)$, where $\phi_r(\cdot)$ is an arbitrary transformation corresponding to the r th class. The new instances, $\{z_{i,r}\}$, are assigned binary labels as in standard SBC reductions and a binary soft classifier $h_2(\cdot)$ is trained. To predict the label of a new instance x , we generate c copies of x , where the r th copy is $z_r = \phi_r(x)$. The label we predict is $\arg\max_r h_2(z_r)$. In general, the transformations $\{\phi_r\}_{r=1}^c$ can have any form. High quality transformations should generate an easy binary problem such that the resulting binary classifier h_2 allows for accurate multiclass predictions. The binary, primal, SVM generated by SBC reduction with transformation functions $\{\phi_r(\cdot)\}_{r=1}^c$ is

$$\min_{\mathbf{w}, b} \|\mathbf{w}\|_2^2 + C \sum_{i=1}^m \ell_{SBC}(x_i, \mathbf{w}, b), \quad (1)$$

where $\ell_{SBC}(x_i, \mathbf{w}, b) \triangleq \sum_{j=1}^c (1 - y_{i,j} (\langle \mathbf{w}, \phi_j(x_i) \rangle + b))_+$, referred to as the SBC loss of x_i , is a sum of the hinge losses of all replications of x_i .

Remark 3.1 The recent approaches for learning with joint input-output spaces [23,5,26] reduce the multi-class problem to the solution of (1) with $\ell(x_i, \mathbf{w}, b) \triangleq (1 - \min_{j:j \neq y_i} \{\langle \mathbf{w}, \phi_{y_i}(x_i) \rangle - \langle \mathbf{w}, \phi_j(x_i) \rangle\})_+$. With this loss function the optimization problem is not a standard SVM, and hence requires the specialized ad-hoc optimization algorithms.

Relying on kernel methods, we consider implicit transformations given in terms of inner products $\langle \phi_r(x_i), \phi_s(x_j) \rangle$, between the transformed instances. These products can be specified as entries of kernel matrix \mathbf{K} of size $cm \times cm$ whose (u, v) entry, where $u = (i-1) \cdot c + r$ and $v = (j-1) \cdot c + s$, is $K(z_{i,r}, z_{j,s}) = \langle \phi_r(x_i), \phi_s(x_j) \rangle$. The resulting dual formulation of SVM is

$$\mathbf{g}_d(\mathbf{K}) \triangleq \max_{\alpha \in \mathbb{R}^{cm}} 2\alpha^T \mathbf{e} - \alpha^T G(\mathbf{K}) \alpha \quad (2)$$

$$\text{such that } \alpha^T \tilde{\mathbf{y}} = 0, \quad 0 \leq \alpha \leq C/(cm), \quad (3)$$

where $\tilde{\mathbf{y}}$ is a $1 \times cm$ vector whose $((i-1) \cdot c + r)$ th entry is $y_{i,r}$, $G(\mathbf{K})$ is a $cm \times cm$ matrix whose $((i-1) \cdot c + r, (j-1) \cdot c + s)$ th entry is $y_{i,r} y_{j,s} K(z_{i,r}, z_{j,s})$, and \mathbf{e} is a $1 \times cm$ vector whose entries are 1's. The binary classifier $h_\alpha(\cdot)$, resulting from the optimization program (2)-(3) has the following form. For any example x , the soft classification of its r th copy z_r is

$$h_\alpha(z_r) \triangleq \sum_{j=1}^m \sum_{s=1}^c y_{j,s} \alpha_{(j-1) \cdot c + s} K(z_{j,s}, z_r) + b. \quad (4)$$

The value of the parameter b is determined using a standard SVM technique (see [24]).¹

We define the kernel $K(z_{i,r}, z_{j,s})$ over the extended example in the following way. Let M be a $c \times \ell$ feature extension matrix as described in Section 2 and denote its r th row by M_r . Using an RBF kernel², $K_{RBF}(a, b) = \exp(-\|a - b\|_2^2 / (2\sigma^2))$, we have

$$K(z_{i,r}, z_{j,s}) \triangleq K_{RBF}(x_i, x_j) \cdot \exp\left(\frac{-\|M_r - M_s\|_2^2}{2\sigma^2}\right). \quad (5)$$

Algorithm 1 summarizes the SBC kernel reduction method with a single kernel.

With this definition (5) of the kernel, the matrix $\mathbf{K} = \mathbf{K}(M)$ can be represented as a Kronecker product [10]

¹ We set b as follows. Let $B = \{(i, r) \mid \alpha_{(i-1) \cdot c + r} > 0\}$. Let $b_{i,r} \triangleq y_{i,r} - \sum_{j=1}^m \sum_{s=1}^c y_{j,s} \alpha_{(j-1) \cdot c + s} K_{RBF}(z_{i,r}, z_{j,s})$ if $(i, r) \in B$ and zero otherwise. $b \triangleq \sum_{i=1}^m \sum_{r=1}^c b_{i,r} / |B|$.

² Any kernel could be used in this framework, but henceforth we discuss the RBF kernel because of its capacity and since an RBF kernel is easier to analyze using the Kronecker product, as discussed below.

Algorithm 1 SBC kernel reduction with a single kernel.

Input: Training set $\{(x_i, y_i)\}_{i=1}^m$, feature extension $c \times \ell$ matrix M , regularization constant $C > 0$.

Output: Dual coefficients α .

- 1: Let M_i be the i th row of M .
 - 2: Generate a kernel matrix \mathbf{K} of size $cm \times cm$ whose $((i-1) \cdot c + r, (j-1) \cdot c + s)$ entry ($1 \leq i, j \leq m$, $1 \leq r, s \leq c$) is $K(z_{i,r}, z_{j,s}) = K_{RBF}(x_i, x_j) \cdot \exp\left(\frac{-\|M_r - M_s\|_2^2}{2\sigma^2}\right)$.
 - 3: For $1 \leq i \leq m$, $1 \leq r \leq c$, let $y_{i,r} = 1$, if $y_i = r$ and $y_{i,r} = -1$ otherwise.
 - 4: Let $\tilde{\mathbf{y}}$ be a $1 \times cm$ vector whose $((i-1) \cdot c + r)$ th entry is $y_{i,r}$.
 - 5: Let $G(\mathbf{K})$ be a $cm \times cm$ matrix whose $((i-1) \cdot c + r, (j-1) \cdot c + s)$ th entry is $y_{i,r} y_{j,s} K(z_{i,r}, z_{j,s})$.
 - 6: Let α be the solution of (2)-(3), applied with $m, c, C, G(\mathbf{K}), \tilde{\mathbf{y}}$.
-

of two smaller matrices. Let \mathbf{K}_x be an $m \times m$ kernel matrix whose (i, j) entry is $K_{RBF}(x_i, x_j)$. Let \mathbf{K}_M be a $c \times c$ kernel matrix whose (r, s) entry is $K_{RBF}(M_r, M_s)$. From the definition of $\mathbf{K}(M)$ and the definition of the Kronecker product it follows that

$$\mathbf{K}(M) = \mathbf{K}_x \otimes \mathbf{K}_M. \quad (6)$$

3.1. Learning Linear Combinations of Basis Kernels

Let $\alpha_{j,s} \triangleq \alpha_{(j-1) \cdot c + s}$ and $\xi_{i,r} \triangleq \xi_{(i-1) \cdot c + r}$. The dual optimization problem (2)-(3) has the following kernelized primal formulation [18]:

$$\mathbf{g}_p(\mathbf{K}) \triangleq \min_{\alpha, \xi \in \mathbb{R}^{cm}, b \in \mathbb{R}} 2C \sum_{i=1}^m \sum_{r=1}^c \xi_{i,r} + \alpha^T \mathbf{K} \alpha \quad (7)$$

such that for all $i = 1, \dots, m$ and $r = 1, \dots, c$,

$$y_{i,r} \left(\sum_{j=1}^m \sum_{s=1}^c \alpha_{j,s} K(x_{i,r}, x_{j,s}) + b \right) \geq 1 - \xi_{i,r},$$

$$\xi_{i,r} \geq 0.$$

Our goal is to find a kernel matrix \mathbf{K} that minimizes the SVM objective function $\mathbf{g}_p(\mathbf{K})$. Since the strong duality property holds for the optimization problems (2) and (7), $\mathbf{g}_p(\mathbf{K}) = \mathbf{g}_d(\mathbf{K})$ and consequently, $\min_{\mathbf{K}} \{\mathbf{g}_p(\mathbf{K})\} = \min_{\mathbf{K}} \{\mathbf{g}_d(\mathbf{K})\}$. As proved in [15, see Proposition 15], the function $\mathbf{g}_d(\mathbf{K})$ with the constraint (3) is convex in \mathbf{K} . Hence, if we restrict \mathbf{K} to be in a convex closed domain and optimize w.r.t. \mathbf{K} , we can find the global minimum of $\mathbf{g}_d(\mathbf{K})$ using gradient descent methods.

However, a direct optimization over \mathbf{K} can be computationally expensive, since it involves solving semi-definite (SDP) optimization problems. Hence our strategy is to take a fixed matrix \mathbf{K}_x and n fixed feature extension matrices $M^{(1)}, \dots, M^{(n)}$, and to search for the best linear combination $\mathbf{K}_\mu = \sum_{i=1}^n \mu_i \mathbf{K}(M^{(i)})$ that optimizes the SVM objective function. Thus, the desired kernel matrix is $\mathbf{K}_\mu = \sum_{i=1}^n \mu_i \mathbf{K}^{(i)}$, where the variables to be optimized are $\boldsymbol{\mu} = (\mu_1, \dots, \mu_n)$. The optimization problem we face is related to extensive literature on learning the linear combinations of matrices. Our solution, shown in Section 3.1, is similar to the one presented in [4].

Algorithm 2 SBC-KERNEL algorithm for learning a linear combination of basis kernels.

Input: Training set $\{(x_i, y_i)\}_{i=1}^m$, feature extension $c \times \ell$ matrices $M^{(j)}$ ($1 \leq j \leq n$), regularization constant $C > 0$, optimization parameter $R > 0$.

Output: Dual coefficients $\boldsymbol{\alpha}$, kernel coefficients $\boldsymbol{\mu}$.

- 1: Let $\boldsymbol{\mu} = \boldsymbol{\mu}_0 \in (\mathbb{R}^+)^n$ be an initial guess for the kernel coefficients.
 - 2: Let M_i be the i th row of M .
 - 3: Generate a kernel matrix \mathbf{K} of size $cm \times cm$ whose $((i-1) \cdot c + r, (j-1) \cdot c + s)$ entry ($1 \leq i, j \leq m$, $1 \leq r, s \leq c$) is $K_{\boldsymbol{\mu}}(z_{i,r}, z_{j,s}) = K_{RBF}(x_i, x_j) \cdot \sum_{k=1}^n \mu_k \exp\left(\frac{-\|M_r^{(k)} - M_s^{(k)}\|_2^2}{2\sigma^2}\right)$.
 - 4: For $1 \leq i \leq m$, $1 \leq r \leq c$, let $y_{i,r} = 1$, if $y_i = r$ and $y_{i,r} = -1$ otherwise.
 - 5: Let $\tilde{\mathbf{y}}$ be a $1 \times cm$ vector whose $((i-1) \cdot c + r)$ th entry is $y_{i,r}$.
 - 6: Let $G(\mathbf{K}_\mu)$ be a $cm \times cm$ matrix whose $((i-1) \cdot c + r, (j-1) \cdot c + s)$ th entry is $y_{i,r} y_{j,s} K_{\boldsymbol{\mu}}(z_{i,r}, z_{j,s})$.
 - 7: Let $\boldsymbol{\alpha}$ be the solution of (2)-(3), applied with $m, c, C, G(\mathbf{K}_\mu), \tilde{\mathbf{y}}$.
 - 8: Make a gradient step: for all $1 \leq j \leq n$, $\mu_j = \mu_j + \boldsymbol{\alpha}^T G(\mathbf{K}_{\tilde{\boldsymbol{\mu}}_j}) \boldsymbol{\alpha}$, where $\tilde{\boldsymbol{\mu}}_j$ is an $1 \times n$ vector with 1 in the j th entry and 0 in other entries.
 - 9: Enforce the constraint $\sum_{j=1}^n \mu_j \leq R$: If $\sum_j \mu_j > R$, then normalize $\boldsymbol{\mu}$ such that $\sum_j \mu_j = R$.
 - 10: Return to Step 7 unless the direction of $\boldsymbol{\mu}$ is not changed by ‘‘much’’ (see the implementation details in Section 5).
-

Using (6) and distributive and associative properties of the Kronecker product³, we obtain

³ For matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ and a scalar k , $\mathbf{A} \otimes \mathbf{B} + \mathbf{A} \otimes \mathbf{C} = \mathbf{A} \otimes (\mathbf{B} + \mathbf{C})$ and $k(\mathbf{A} \otimes \mathbf{B}) = \mathbf{A} \otimes (k\mathbf{B})$.

$$\begin{aligned} \mathbf{K}_\mu &= \sum_{i=1}^n \mu_i \mathbf{K}(M^{(i)}) \\ &= \mathbf{K}_x \otimes \left(\sum_{i=1}^n \mu_i \mathbf{K}_{M^{(i)}} \right) \triangleq \mathbf{K}_x \otimes \mathbf{K}_{\mathbf{M}_\mu}, \end{aligned} \quad (8)$$

where $\mathbf{K}_{\mathbf{M}_\mu} \triangleq \sum_{i=1}^n \mu_i \mathbf{K}_{M^{(i)}}$ is a $c \times c$ kernel matrix, and this matrix implicitly represents the feature extension matrix that optimizes the SVM objective function. Let $V(r, s) \triangleq \sum_{i=1}^n \mu_i \exp\left(\frac{-\|M_r^{(i)} - M_s^{(i)}\|_2^2}{2\sigma^2}\right)$. By plugging in the kernel (8) in (4) we obtain that the resulting binary classifier $h_{\boldsymbol{\alpha}, \boldsymbol{\mu}}(\cdot)$ has the following form. For any example x , the soft classification of its r th copy z_r is

$$h_{\boldsymbol{\alpha}, \boldsymbol{\mu}}(z_r) \triangleq \sum_{j=1}^m \sum_{s=1}^c y_{j,s} \alpha_{(j-1) \cdot c + s} K_{RBF}(x_j, x) V(r, s) + b. \quad (9)$$

Clearly, different sets of matrices $\{M^{(i)}\}_{i=1}^n$ may lead to different realizations. In Section 5, we focus on a particular choice of such matrices.

We find a vector $\boldsymbol{\mu}$ of kernel coefficients using simple gradient descent. The gradient of $\mathbf{g}_d(\mathbf{K})$ is an $n \times 1$ vector and its i th entry is $\frac{\partial \mathbf{g}_d(\mathbf{K}_\mu)}{\partial \mu_i} = -\boldsymbol{\alpha}^T G(\mathbf{K}^{(i)}) \boldsymbol{\alpha}$. Since $\mathbf{K}^{(i)}$ is a kernel matrix, it is positive semidefinite. It can be verified that the matrix $G(\mathbf{K}^{(i)})$ is also positive semidefinite. Hence, for any $1 \leq i \leq n$, $\frac{\partial \mathbf{g}_d(\mathbf{K}_\mu)}{\partial \mu_i} \leq 0$. Thus, if we start from positive μ_i 's and proceed in the direction opposite to the direction of the gradient, they will remain positive throughout the optimization procedure. The positiveness of μ_i will ensure that the matrix $\mathbf{K}_\mu = \sum_{i=1}^n \mu_i \mathbf{K}^{(i)}$ will always remain a valid kernel matrix. We also impose the constraint $\sum_{j=1}^n \mu_j \leq R$, for some (fixed) R to ensure that the feasible set of kernels lies in a convex closed domain and to prevent possible overfitting. Thus, since $\mathbf{g}_d(\mathbf{K})$ is convex in \mathbf{K} , we can find the global minimum of $\mathbf{g}_d(\mathbf{K}_\mu)$, under the constraint $\sum_{i=1}^n \mu_i \leq R$, using gradient descent method with an appropriate step size.

Following [4], who proposed a similar routine⁴, we use a simple gradient descent procedure for finding the optimal \mathbf{K}^* , which is summarized in Algorithm 2.

⁴ The difference between our procedure and the one of [4] is that we don't have the constraint $\mu_i \geq 0$. As we showed above, this constraint is superfluous.

	Number of binary problems	Size of each binary problem
OVA	c	m
ALL-PAIRS	$c(c-1)/2$	$\approx 2m/c$
ECOC	ℓ	m
SBC-ECOC	1	ℓm
SBC-SINGLE	1	cm
SBC-IDENTITY	1	cm
SBC-KERNEL	1	cm

Table 1

Complexity of several known and new multiclass-to-binary reductions in terms of m =number of (multiclass) training examples, c =number of classes, ℓ =length of error-correcting code.

3.2. On the Complexity of Multiclass to Binary Reductions

It is interesting to consider the complexity of SBC reductions in terms of the number and the size of the binary problems involved. Table 3.2 compares the complexity of the SBC reductions described in Section 2.2, of our novel SBC-KERNEL reduction, and of several widely used multiclass-to-binary reductions.

4. Theoretical Perspective

In this section we present two theoretical results providing further insight on the properties of SBC reductions. Our first result, presented in Section 4.1, is a risk bound for the multiclass classifier obtained by an SBC reduction. Our second result, presented in Section 4.2, is concerned with the asymptotic analysis of the empirical risk minimization method over the loss function implied by the SBC reduction.

4.1. Risk Bound

Let $\ell_M(h_{\alpha, \mu}, (x, y))$ be the multiclass 0/1 loss of the SBC classifier, using the hypothesis $h_{\alpha, \mu}$ for its binary decisions, over a multiclass example (x, y) . Consider the set S_2 consisting of m independent replicas by drawing uniformly at random one of the c replicas of each example $(x, y) \in S$. Let $\hat{\ell}_\gamma(h_{\alpha, \mu}) \triangleq \frac{1}{m} \sum_{\{i \mid y_i h_{\alpha, \mu}(z_i) < \gamma, (z_i, y_i) \in S_2\}}$ be the empirical γ -margin error of $h_{\alpha, \mu}$ on the set S_2 , and $Q \triangleq \frac{8}{m} (2 + 256 \frac{R}{\gamma^2} \log \frac{\gamma em}{8\sqrt{R}} \log \frac{128mR}{\gamma^2} - \log \delta + n \log \frac{128em^3 R}{\gamma^2 n})$. The following lemma bounds the multiclass 0/1 error in terms of Q and γ -margin error on the set S_2 .

Lemma 4.1 *For any $\delta > 0$ and $\gamma > 0$, with probability of at least $1 - \delta$ over the random draw of the training set S_2 ,*

$$\ell_M(h_{\alpha, \mu}) \leq c \cdot \hat{\ell}_\gamma(h_{\alpha, \mu}) + c\sqrt{Q}. \quad (10)$$

The proof of this lemma appears in Appendix A.

While bound (10) is certainly not tight in general, it is useful when the number of classes and the empirical margin binary error are small⁵ (in particular, in the realizable case). Note also that the slack term in this bound increases with R . However, with a larger R , the search space for optimal μ also increases. Thus the parameter R expresses a tradeoff between the size of the search space and the value of the slack term in the risk bound.

4.2. Properties of Empirical Risk Minimization

We refer to the method of empirical risk minimization method over the loss function implied by the SBC reduction as ERM-SBC. Let $\mathbf{f}(x)$ be an $1 \times c$ vector with the r th entry, referred to as $f_r(x)$, which is a soft-classification given by ERM-SBC to the r th copy of x . The SBC loss of the example x is $\ell_{SBC}(\mathbf{f}(x)) = \sum_{r=1}^c (1 - y_r f_r(x))_+$. ERM-SBC finds a mapping \mathbf{f} minimizing $\mathbf{E}_{(x, y) \sim \mathcal{D}_M} \{\ell_{SBC}(\mathbf{f}(x))\}$ and classifies each $x \in \mathcal{X}$ as $\arg \max_{1 \leq r \leq c} \{f_r(x)\}$. Let $\mathcal{D}_{y|x}$ be the conditional distribution of the multiclass label y of the example x . The following lemma⁶ provides partial theoretical justification for the use of the SBC loss function.

Lemma 4.2 *Let $p \triangleq \min_x \max_{1 \leq i \leq c} \{\mathbf{P}_{y \sim \mathcal{D}_{y|x}}(y = i|x)\}$. If $p > \frac{1}{2}$ then ERM-SBC is consistent. Otherwise, ERM-SBC is not consistent.*

We conclude from this lemma that ERM-SBC is consistent in easy problems, when $p > \frac{1}{2}$. The proof of lemma 4.2 is based on ideas from [16] and appears in Appendix B. We note that the relevance of the lemma to the practical settings is rather limited. In particular, the actual learning algorithm does not perform empirical risk minimization, but incorporates a regularization term.

⁵ If $n = 1$, meaning that we use only a single basis kernel, then the tighter PAC-Bayesian bound [17] straightforwardly applies to our setting. This bound has a logarithmic dependence on the number of classes.

⁶ This result is also mentioned informally, without a proof, in [25] (Section 4.3).

5. Experiments

As discussed in Section 3, implementing our SBC-KERNEL approach requires that we select a set of feature extension matrices. We took c matrices, each of size $c \times c$, where $M^{(r)}$ was taken to be an all-zeros matrix except for a single unit entry, $M^{(r)}(r, r) = 1, r = 1, \dots, c$.

We used ten UCI datasets. The computational load associated with SBC reductions (due to replication), compelled us to restrict our experiment to datasets with a small number of classes and relatively few examples. Nominal attributes with t values were replaced by t binary features, where the i th binary feature was set to 1 iff the corresponding nominal attribute took the i th possible value. For each feature, its average and standard deviation over the training set was computed, and these were used to normalize the data (training and testing) by subtracting the average and dividing by the standard deviation.

Ten-fold cross-validation (10xCV) was used; namely, in each fold, the union of nine out of ten equally-sized random subsets were used for training, and the tenth for testing. In all our experiments, we used the *SVM*Torch implementation of SVM [6] and applied it with an RBF kernel. To optimize the parameters (σ and C), we followed [19] and used a simple greedy search via 10xCV over the training set as follows. Initial values of σ and C were set to 1. The value of σ was then increased or decreased by a factor of 2 until no improvements were observed for three consecutive attempts. Then, σ was fixed at the best value found and an identical optimization was performed over C .⁷ We operated SBC-KERNEL with R (arbitrarily) set to the square root of the training set size. The termination criterion of the algorithm was chosen to be $\frac{\|\mu_{\text{old}} - \mu_{\text{new}}\|_2}{\|\mu_{\text{old}}\|_2} \leq 0.001$, where μ_{old} and μ_{new} are the values of μ at the previous and at the current iteration, respectively.

In addition to our SBC-KERNEL method, we experimented with three other SBC reductions: SBC-SINGLE, SBC-IDENTITY, and SBC-ECOC (using a BCH coding matrix). We also tested the three traditional reductions: OVA, ALL-PAIRS and ECOC (applied with the same BCH coding matrix). Overall, we tested seven algorithms. For all the algorithms tested, we used the same parameter tuning strategy to search for a single best pair of parameters (σ and C) for all the binary classifiers in-

⁷ One can consider various ways to improve the optimization routine suggested by [19]. For example, it is possibly better to jointly optimize over C and σ , but computationally, this would be rather expensive.

involved in each multiclass application. While this method may favor reductions that utilize a smaller number of binary problems, we believe this is a fair comparison that allocates similar search resources to each algorithm while searching for effective hyper-parameters.

Table 2 presents the errors (%) obtained for each of the seven methods. The best results (lowest errors) in each row appear in boldface. The average ranks of the various algorithms appear in the last row of the table. These ranks were computed as averages of row ranks.⁸ The best performer in terms of ranks is SBC-KERNEL. The worst performer is SBC-SINGLE. We conducted a statistical test to assess the significance of the ranks. Specifically we applied the F_F test [7] with confidence level 90% and found that the difference between SBC-KERNEL and other algorithms is statistically significant. Comparing our results to [19] (over the two common datasets `Car` and `Page blocks`), we see that our results for OVA are slightly better, and our results for ALL-PAIRS are similar.

Our results show that SBC without a learned kernel (SBC-IDENTITY) is better than the standard multiclass reduction (OVA, ALL-PAIRS, ECOC). Thus the sources of the observed improvement over the standard multiclass reductions are both the SBC reduction *and* the kernel learning.

During the course of our experiments with the SBC-KERNEL method, we observed a significant correlation in most cases between the class distribution of the training data and the final μ_i weights reached after the optimization. These correlations indicate that the computational overhead might be reduced by directly using these class distributions as the final μ_i kernel weights, thus solving a single SVM problem. We leave this direction for future work.

6. Concluding Remarks

We introduced a powerful family of SBC reductions based on large margin optimization. For multiclass problems with a small number of classes, this approach is well motivated by a generalization bound, which is obtained as a corollary of a known bound for binary classification. We tested our method and compared it to six other known methods over UCI datasets with a small number of classes. These tests indicate that the proposed approach can yield superior performance when the number of classes is small.

⁸ For each row, if the errors of all algorithms are distinct, they are assigned the ranks in $\{1, \dots, 10\}$. When algorithms share exactly the same error, they are all assigned the same average rank.

	\mathcal{Y}	STANDARD REDUCTIONS			SBC REDUCTIONS			
		OVA	ALL-PAIRS	ECOC	SBC-ECOC	SBC-SINGLE	SBC-IDENTITY	SBC-KERNEL
Car	4	1.10	0.76	4.36	3.90	5.64	4.36	0.41
Page blocks	5	3.33	4.64	3.20	3.42	3.51	3.18	2.96
Iris	3	21.33	24.00	6.00	4.00	66.67	6.00	4.00
Wine	3	5.88	4.71	1.76	3.53	65.29	2.35	2.94
Vehicle	3	25.48	25.00	20.48	20.48	76.55	20.95	13.10
Scales	3	22.42	22.74	92.26	22.42	92.26	8.06	3.39
Lenses	3	40	55	80	40	80	40	40
New Thyroid	3	5.71	5.71	28.57	5.71	28.57	3.8	3.3
Postoperative	3	28.89	30	30	28.89	30	30	32.22
TAE	3	53.33	52	67.33	53.33	67.33	67.33	40
AVERAGE RANK		4	4.68	4.32	3.27	6.45	3.36	1.91

Table 2
Number of classes ($|\mathcal{Y}|$), 10xCV average test errors (%), and average ranks of seven algorithms. Best results for the dataset are boldfaced.

Many questions remain open for future research. The original Kessler construction relies on linear separability of the training set. Extensions that can handle arbitrary problems were studied in [12]. It would be interesting to compare these constructions to ours. In general, it would be very interesting to explore other types of feature extension matrices. A direct optimization of these matrices can also be considered. But, as discussed in Section 3.1, the resulting optimization problem will no longer be easy to solve. It would also be interesting to reverse-engineer the resulting kernel transformations and identify a single diagonal extension matrix with good performance. While we believe that such a matrix exists, finding it using the RBF kernel and our optimization procedure is difficult since the objective function becomes non-convex.

The main bottleneck in all SBC methods is the data replication, which poses a true bottleneck for large problems. For small problems, this computational load is affordable, and as we show, beneficial. To handle large problems, this bottleneck might be alleviated by using fast approximation to SVM optimization (e.g., see [3], [22], [14]).

References

- [1] E.L. Allwein, R.E. Schapire, and Y. Singer. Reducing multiclass to binary: a unifying approach for margin classifiers. *JMLR*, 1:113–141, 2000.
- [2] D. Anguita, S. Ridella, and D. Sterpi. A new method for multiclass support vector machines. In *Proceedings on the International Joint Conference on Neural Networks*, pages 407–412, 2004.
- [3] A. Bordes, S. Ertekin, J. Weston, and L. Bottou. Fast kernel classifiers with online and active learning. *JMLR*, 6:1579–1619, 2005.
- [4] O. Bousquet and D. Herrmann. On the complexity of learning the kernel matrix. In *NIPS*, 2003.
- [5] S. Canu and A. Smola. Kernel methods and the exponential family. *Neurocomputing*, 69:714–720, 2006.
- [6] R. Collobert and S. Bengio. SVM-Torch: support vector machines for large-scale regression problems. *JMLR*, 1:143–160, 2001.
- [7] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *JMLR*, 7:1–30, 2006.
- [8] T.G. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *JAIR*, 2:263–286, 1995.
- [9] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis*. Wiley, New York, 1973.
- [10] H. Eves. *Elementary matrix theory*. Dover publications, 1980.
- [11] J.H. Friedman. Another approach to polychotomous classification. Technical report, Stanford University, 1996.
- [12] S. Har-Peled, D. Roth, and D. Zimak. Constraint classification for multiclass classification and ranking. In *NIPS*. 2003.
- [13] C.W. Hsu and C.J. Lin. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13:415–425, 2002.
- [14] S. Keerthi, O. Chapelle, and D. DeCoste. Building support vector machines with reduced classifier complexity. *JMLR*, 6:1493–1515, 2006.
- [15] G. Lanckriet, N. Cristianini, P. Bartlett, L. El-Ghaoui, and M. Jordan. Learning the kernel matrix via semidefinite programming. *JMLR*, 5:27–72, 2004.
- [16] Y. Lee, Y. Lin, and G. Wahba. Multicategory support vector machines, theory, and application to the classification of microarray data and satellite radiance data. *J. of the Amer. Stat. Assoc.*, 99:67–81, 2004.
- [17] D. McAllester. Generalization bounds and consistency for structured labeling. In G. BakIr et al., editor, *Predicting Structured Data*. MIT Press, 2007.

- [18] R. Rifkin. *Everything old is new again: a fresh look at historical approaches in machine learning*. PhD thesis, MIT, 2002.
- [19] R. Rifkin and A. Klautau. In defense of one-vs-all classification. *JMLR*, 5:101–141, 2004.
- [20] T.G. Sejnowski and C.R. Rosenberg. Parallel networks that learn to pronounce English text. *Journal of Complex Systems*, 1(1):145–168, 1987.
- [21] N. Srebro and S. Ben-David. Learning bounds for support vector machines with learned kernels. In *COLT*, pages 169–183, 2006.
- [22] I. Tsang, J. Kwok, and P. Cheung. Core vector machines: fast svm training on very large data sets. *JMLR*, 5:363–392, 2005.
- [23] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *JMLR*, 6:1453–1484, 2005.
- [24] V. Vapnik. *Statistical Learning Theory*. Wiley Interscience, 1998.
- [25] T. Zhang. Statistical analysis of some multi-category large margin classification methods. *JMLR*, 2004.
- [26] A. Zien and C. S. Ong. Multiclass multiple kernel learning. In *ICML*, 2007.

Appendix A. Proof of Lemma 4.1

We relate the multiclass 0/1 loss of the SBC reduction to the 0/1 loss of the underlying binary classifier. Let $\{(z_r, y_r)\}_{r=1}^c$ be the c copies of x , where $z_r = z_r(x)$ is the r th extension of x and y_r is its binary label. Denote by $\ell_B(h_{\alpha, \mu}, (z_r, y_r))$ the binary 0/1 loss of w on the replica (z_r, y_r) . It is easy to see that

$$\forall (x, y), \quad \ell_M(h_{\alpha, \mu}, (x, y)) \leq \sum_{r=1}^c \ell_B(h_{\alpha, \mu}, (z_r, y_r)) . \quad (\text{A.1})$$

Assume that $P(x, y)$ is the (unknown) underlying distribution of the data and let $\ell_M(h_{\alpha, \mu})$ be the true average 0/1 multiclass error of the SBC classifier. Let $dP(x, y)$ be a probability mass of the multiclass example (x, y) , sampled from some distribution \mathcal{D}_M . We define the distribution \mathcal{D}_2 over the binary replications (z_r, y_r) in the following way. For each example (x, y) , the probability mass of each of its replicas is $\frac{1}{c}dP(x, y)$. Then the true average binary 0/1 loss of $h_{\alpha, \mu}$ is $\ell_B(h_{\alpha, \mu}) = \int \frac{1}{c} \sum_{r=1}^c \ell_B(h_{\alpha, \mu}, (z_r, y_r)) dP(x, y)$. Using (A.1) we have

$$\begin{aligned} \ell_M(h_{\alpha, \mu}) &\leq \int \sum_{r=1}^c \ell_B(h_{\alpha, \mu}, (z_r, y_r)) dP(x, y) \\ &= c \cdot \ell_B(h_{\alpha, \mu}) . \end{aligned}$$

We bound $\ell_B(h_{\alpha, \mu})$ using the recent bound of [21] for ‘learned kernels’. To apply this bound the training examples should be independent. However, the examples in the set S' , containing all replicas of the training multiclass examples, are dependent. This is so because for

each example $(x, y) \in S$, all its replicas are included in S' . To this end we consider the set S_2 consisting of m independent replicas by drawing uniformly at random one of the c replicas of each example $(x, y) \in S$. The proof is concluded by the application of Theorem 2 of [21] for the independent examples from the set S_2 .

Appendix B. Proof of Lemma 4.2

The proof is inspired by the proofs of consistency and inconsistency in [16]. We have

$$\mathbf{E}_{(x, y) \sim \mathcal{D}} \{\ell(\mathbf{f}(x))\} = \mathbf{E}_{x \sim \mathcal{D}_x} \mathbf{E}_{y \sim \mathcal{D}_{y|x}} \{\ell(\mathbf{f}(x)) | x\} . \quad (\text{B.1})$$

Therefore the right-hand side of (B.1) is minimized iff the expectation $\mathbf{E}_{y \sim \mathcal{D}_{y|x}} \{\ell(\mathbf{f}(x)) | x\}$ is minimized for any $x \in \mathcal{X}$. Let $\mathbf{P}(i|x) \triangleq \mathbf{P}_{y \sim \mathcal{D}_{y|x}}(y = i|x)$. Using the definition of the SBC loss we obtain

$$\begin{aligned} \mathbf{E}_{y \sim \mathcal{D}_{y|x}} \{\ell(\mathbf{f}(x)) | x\} &= \\ &= \sum_{i=1}^c \left(\sum_{\substack{j=1 \\ j \neq i}}^c (1 + f_j(x))_+ + (1 - f_i(x))_+ \right) \mathbf{P}(i|x) \\ &= \sum_{i=1}^c (1 - \mathbf{P}(i|x)) (1 + f_i(x))_+ + \\ &\quad (1 - f_i(x))_+ \mathbf{P}(i|x) . \end{aligned} \quad (\text{B.2})$$

Suppose there exists $x \in \mathcal{X}$ such that $p_{\max}(x) \leq \frac{1}{2}$. Therefore for all $1 \leq i \leq c$, it holds that $1 - \mathbf{P}_{y \sim \mathcal{D}_{y|x}}(y = i|x) \geq \frac{1}{2}$ and the minimum of (B.2) is achieved when all components of $\mathbf{f}(x)$ are -1 . In this case for the example x the algorithm \mathcal{A} can choose the label which is not the most probable one. Hence the algorithm \mathcal{A} is inconsistent.

Suppose that for all $x \in \mathcal{X}$ it holds that $p_{\max}(x) > \frac{1}{2}$. Let $r(x) \triangleq \arg \max_{1 \leq i \leq c} \{\mathbf{P}_{(x, y) \sim \mathcal{D}}(y = i|x)\}$. For all $1 \leq i \leq c$, such that $i \neq r(x)$ it holds that $\mathbf{P}_{y \sim \mathcal{D}_{y|x}}(y = i|x) < \frac{1}{2}$. Therefore the minimum of (B.2) is achieved when all components, except the $r(x)$ th component, of $\mathbf{f}(x)$ are -1 and the $r(x)$ th component of $\mathbf{f}(x)$ is 1 . In this case for the example x the algorithm \mathcal{A} chooses the most probable label. Hence the algorithm \mathcal{A} is consistent.