Multiclass Learning Approaches: A Theoretical Comparison with Implications

Amit Daniely
Department of Mathematics
The Hebrew University
Jerusalem, Israel

Sivan Sabato
Microsoft Research
1 Memorial Drive
Cambridge, MA 02142, USA

Shai Shalev-Shwartz
School of CS and Eng.
The Hebrew University
Jerusalem, Israel

Abstract

We theoretically analyze and compare the following five popular multiclass classification methods: One vs. All, All Pairs, Tree-based classifiers, Error Correcting Output Codes (ECOC) with randomly generated code matrices, and Multiclass SVM. In the first four methods, the classification is based on a reduction to binary classification. We consider the case where the binary classifier comes from a class of VC dimension d, and in particular from the class of halfspaces over $\mathbb{R}^d$. We analyze both the estimation error and the approximation error of these methods. Our analysis reveals interesting conclusions of practical relevance, regarding the success of the different approaches under various conditions. Our proof technique employs tools from VC theory to analyze the *approximation error* of hypothesis classes. This is in contrast to most previous uses of VC theory, which only deal with estimation error.

1 Introduction

In this work we consider multiclass prediction: The problem of classifying objects into one of several possible target classes. Applications include, for example, categorizing documents according to topic, and determining which object appears in a given image. We assume that objects (a.k.a. instances) are vectors in $\mathcal{X} = \mathbb{R}^d$ and the class labels come from the set $\mathcal{Y} = [k] = \{1, \ldots, k\}$. Following the standard PAC model, the learner receives a training set of m examples, drawn i.i.d. from some unknown distribution, and should output a classifier which maps $\mathcal{X}$ to $\mathcal{Y}$.

The centrality of the multiclass learning problem has spurred the development of various approaches for tackling the task. Perhaps the most straightforward approach is a reduction from multiclass classification to binary classification. For example, the One-vs-All (OvA) method is based on a reduction of the multiclass problem into k binary problems, each of which discriminates between one class to all the rest of the classes (e.g. Rumelhart et al. [1986]). A different reduction is the All-Pairs (AP) approach in which all pairs of classes are compared to each other [Hastie and Tibshirani, 1998]. These two approaches have been unified under the framework of Error Correction Output Codes (ECOC) [Dietterich and Bakiri, 1995, Allwein et al., 2000]. A tree-based classifier (TC) is another reduction in which the prediction is obtained by traversing a binary tree, where at each node of the tree a binary classifier is used to decide on the rest of the path (see for example Beygelzimer et al. [2007]).

All of the above methods are based on reductions to binary classification. We pay special attention to the case where the underlying binary classifiers are linear separators (halfspaces). Formally, each $w \in \mathbb{R}^{d+1}$ defines the linear separator $h_w(x) = \text{sign}(\langle w, \bar{x} \rangle)$, where $\bar{x} = (x, 1) \in \mathbb{R}^{d+1}$ is the concatenation of the vector x and the scalar 1. While halfspaces are our primary focus, many of our results hold for any underlying binary hypothesis class of VC dimension $d + 1$.

Other, more direct approaches to multiclass classification over $\mathbb{R}^d$ have also been proposed (e.g. Vapnik [1998], Weston and Watkins [1999], Crammer and Singer [2001]). In this paper we analyze the Multiclass SVM (MSVM) formulation of Crammer and Singer [2001], in which each hypothesis is of the form $h_W(x) = \operatorname{argmax}_{i \in [k]}(W\bar{x})_i$, where W is a $k \times (d+1)$ matrix and $(W\bar{x})_i$ is the i'th element of the vector $W\bar{x} \in \mathbb{R}^k$.

We theoretically analyze the prediction performance of the aforementioned methods, namely, OvA, AP, ECOC, TC, and MSVM. The error of a multiclass predictor $h : \mathbb{R}^d \to [k]$ is defined to be the probability that $h(x) \neq y$, where (x, y) is sampled from the underlying distribution $\mathcal{D}$ over $\mathbb{R}^d \times [k]$, namely, $\operatorname{Err}(h) = \mathbb{P}_{(x,y)\sim\mathcal{D}}[h(x) \neq y]$. Our main goal is to understand which method is preferable in terms of the error it will achieve, based on easy-to-verify properties of the problem at hand.

Our analysis pertains to the type of classifiers each method can potentially find, and does not depend on the specific training algorithm. More precisely, each method corresponds to a hypothesis class, $\mathcal{H}$, which contains the multiclass predictors that may be returned by the method. For example, the hypothesis class of MSVM is $\mathcal{H} = \{x \mapsto \operatorname{argmax}_{i\in[k]}(W\bar{x})_i : W \in \mathbb{R}^{k\times(d+1)}\}$.

A learning algorithm, A, receives a training set, $S = \{(x_i, y_i)\}_{i=1}^m$, sampled i.i.d. according to $\mathcal{D}$, and returns a multiclass predictor which we denote by $A(S) \in \mathcal{H}$. A learning algorithm is called an Empirical Risk Minimizer (ERM) if it returns a hypothesis in $\mathcal{H}$ that minimizes the empirical error on the sample. We denote by $h^\star$ a hypothesis in $\mathcal{H}$ with minimal error,[1] that is, $h^\star \in \operatorname{argmin}_{h\in\mathcal{H}} \operatorname{Err}(h)$.

When analyzing the error of $A(S)$, it is convenient to decompose this error as a sum of *approximation error* and *estimation error*:

$$\operatorname{Err}(A(S)) = \underbrace{\operatorname{Err}(h^\star)}_{\text{approximation}} + \underbrace{\operatorname{Err}(A(S)) - \operatorname{Err}(h^\star)}_{\text{estimation}} . \qquad (1)$$

- The **approximation error** is the minimum error achievable by a predictor in the hypothesis class, $\mathcal{H}$. The approximation error does not depend on the sample size, and is determined solely by the allowed hypothesis class[2].
- The **estimation error** of an algorithm is the difference between the approximation error, and the error of the classifier the algorithm chose based on the sample. This error exists both for statistical reasons, since the sample may not be large enough to determine the best hypothesis, and for algorithmic reasons, since the learning algorithm may not output the best possible hypothesis given the sample. For the ERM algorithm, the estimation error can be bounded from above by order of $\sqrt{C(\mathcal{H})/m}$ where $C(\mathcal{H})$ is a complexity measure of $\mathcal{H}$ (analogous to the VC dimension) and m is the sample size. A similar term also bounds the estimation error from below *for any algorithm*. Thus $C(\mathcal{H})$ is an estimate of the best achievable estimation error for the class.

When studying the estimation error of different methods, we follow the standard distribution-free analysis. Namely, we will compare the algorithms based on the worst-case estimation error, where worst-case is over all possible distributions $\mathcal{D}$. Such an analysis can lead us to the following type of conclusion: If two hypothesis classes have roughly the same complexity, $C(\mathcal{H}_1) \approx C(\mathcal{H}_2)$, and the number of available training examples is significantly larger than this value of complexity, then for both hypothesis classes we are going to have a small estimation error. Hence, in this case the difference in prediction performance between the two methods will be dominated by the approximation error and by the success of the learning algorithm in approaching the best possible estimation error. In our discussion below we disregard possible differences in optimality which stem from algorithmic aspects and implementation details. A rigorous comparison of training heuristics would certainly be of interest and is left to future work.

For the approximation error we will provide even stronger results, by comparing the approximation error of classes for *any* distribution. We rely on the following definition.

[1] For simplicity, we assume that the minimum is attainable.

[2] Note that, when comparing different hypothesis classes over the same distribution, the Bayes error is constant. Thus, in the definition of approximation error, we do not subtract the Bayes error.

Definition 1.1. *Given two hypothesis classes, $\mathcal{H}, \mathcal{H}'$, we say that $\mathcal{H}$* essentially contains *$\mathcal{H}'$ if for any distribution, the approximation error of $\mathcal{H}$ is at most the approximation error of $\mathcal{H}'$. $\mathcal{H}$* strictly contains *$\mathcal{H}'$ if, in addition, there is a distribution for which the approximation error of $\mathcal{H}$ is strictly smaller than that of $\mathcal{H}'$.*

Our main findings are as follows (see a full comparison in Table 1). The formal statements are given in Section 3.

- The estimation errors of OvA, MSVM, and TC are all roughly the same, in the sense that $C(\mathcal{H}) = \tilde{\Theta}(dk)$ for all of the corresponding hypothesis classes. The complexity of AP is $\tilde{\Theta}(dk^2)$. The complexity of ECOC with a code of length l and code-distance δ is at most $\tilde{O}(dl)$ and at least $d\delta/2$. It follows that for randomly generated codes, $C(\mathcal{H}) = \tilde{\Theta}(dl)$. Note that this analysis shows that a larger code-distance yields a larger estimation error and might therefore hurt performance. This contrasts with previous "reduction-based" analyses of ECOC, which concluded that a larger code distance improves performance.
- We prove that the hypothesis class of MSVM essentially contains the hypothesis classes of both OvA and TC. Moreover, these inclusions are strict. Since the estimation errors of these three methods are roughly the same, it follows that the MSVM method dominates both OvA and TC in terms of achievable prediction performance.
- In the TC method, one needs to associate each leaf of the tree to a label. If no prior knowledge on how to break the symmetry is known, it is suggested in Beygelzimer et al. [2007] to break symmetry by choosing a random permutation of the labels. We show that whenever $d \ll k$, for any distribution $\mathcal{D}$, with high probability over the choice of a random permutation, the approximation error of the resulting tree would be close to $1/2$. It follows that a random choice of a permutation is likely to yield a poor predictor.
- We show that if $d \ll k$, for any distribution $\mathcal{D}$, the approximation error of ECOC with a randomly generated code matrix is likely to be close to $1/2$.
- We show that the hypothesis class of AP essentially contains the hypothesis class of MSVM (hence also that of OvA and TC), and that there can be a substantial gap in the containment. Therefore, as expected, the relative performance of AP and MSVM depends on the well-known trade-off between estimation error and approximation error.

	TC	**OvA**	**MSVM**	**AP**	**random ECOC**
Estimation error	dk	dk	dk	dk^2	dl
Approximation error	$\geq$ MSVM $\approx 1/2$ when $d \ll k$	$\geq$ MSVM	$\geq$ AP	smallest	incomparable $\approx 1/2$ when $d \ll k$
Testing run-time	$d\log(k)$	dk	dk	dk^2	dl

Table 1: Summary of comparison

The above findings suggest that in terms of performance, it may be wiser to choose MSVM over OvA and TC, and especially so when $d \ll k$. We note, however, that in some situations (e.g. $d = k$) the prediction success of these methods can be similar, while TC has the advantage of having a testing run-time of $d\log(k)$, compared to the testing run-time of dk for OvA and MSVM. In addition, TC and ECOC may be a good choice when there is additional prior knowledge on the distribution or on how to break symmetry between the different labels.

1.1 Related work

Allwein et al. [2000] analyzed the multiclass error of ECOC as a function of the binary error. The problem with such a "reduction-based" analysis is that such analysis becomes problematic if the underlying binary problems are very hard. Indeed, our analysis reveals that the underlying binary problems would be too hard if $d \ll k$ and the code is randomly generated. The experiments in Allwein et al. [2000] show that when using kernel-based SVM or AdaBoost as the underlying classifier, OvA is inferior to random ECOC. However, in their experiments, the number of classes is small relative to the dimension of the feature space, especially if working with kernels or with combinations of weak learners.

Crammer and Singer [2001] presented experiments demonstrating that MSVM outperforms OvA on several data sets. Rifkin and Klautau [2004] criticized the experiments of Crammer and Singer [2001], Allwein et al. [2000], and presented another set of experiments demonstrating that all methods perform roughly the same when the underlying binary classifier is very strong (SVM with a Guassian kernel). As our analysis shows, it is not surprising that with enough data and powerful binary classifiers, all methods should perform well. However, in many practical applications, we will prefer not to employ kernels (either because of shortage of examples, which might lead to a large estimation error, or due to computational constraint), and in such cases we expect to see a large difference between the methods.

Beygelzimer et al. [2007] analyzed the *regret* of a specific training method for trees, called Filter Tree, as a function of the regret of the binary classifier. The regret is defined to be the difference between the learned classifier and the Bayes-optimal classifier for the problem. Here again we show that the regret values of the underlying binary classifiers are likely to be very large whenever $d \ll k$ and the leaves of the tree are associated to labels in a random way. Thus in this case the regret analysis is problematic. Several authors presented ways to learn better splits, which corresponds to learning the association of leaves to labels (see for example Bengio et al. [2011] and the references therein). Some of our negative results do not hold for such methods, as these do not randomly attach labels to tree leaves.

Daniely et al. [2011] analyzed the properties of multiclass learning with various ERM learners, and have also provided some bounds on the estimation error of multiclass SVM and of trees. In this paper we both improve these bounds, derive new bounds for other classes, and also analyze the approximation error of the classes.

2 Definitions and Preliminaries

We first formally define the hypothesis classes that we analyze in this paper.

Multiclass SVM (MSVM): For $W \in \mathbb{R}^{k\times(d+1)}$ define $h_W : \mathbb{R}^d \to [k]$ by $h_W(x) = \text{argmax}_{i\in[k]}(W\bar{x})_i$ and let $\mathcal{L} = \{h_W : W \in \mathbb{R}^{k\times(d+1)}\}$. Though NP-hard in general, solving the ERM problem with respect to $\mathcal{L}$ can be done efficiently in the realizable case (namely, whenever exists a hypothesis with zero empirical error on the sample).

Tree-based classifiers (TC): A tree-based multiclass classifier is a full binary tree whose leaves are associated with class labels and whose internal nodes are associated with binary classifiers. To classify an instance, we start with the root node and apply the binary classifier associated with it. If the prediction is 1 we traverse to the right child. Otherwise, we traverse to the left child. This process continues until we reach a leaf, and then we output the label associated with the leaf. Formally, a tree for k classes is a full binary tree T together with a bijection $\lambda : \text{leaf}(T) \to [k]$, which associates a label to each of the leaves. We usually identify T with the pair (T, λ). The set of internal nodes of T is denoted by $N(T)$. Let $\mathcal{H} \subset \{\pm 1\}^{\mathcal{X}}$ be a binary hypothesis class. Given a mapping $C : N(T) \to \mathcal{H}$, define a multiclass predictor, $h_C : \mathcal{X} \to [k]$, by setting $h_C(x) = \lambda(v)$ where v is the last node of the root-to-leaf path $v_1, \dots v_m = v$ such that v_{i+1} is the left (resp. right) child of v_i if $C(v_i)(x) = -1$ (resp. $C(v_i)(x) = 1$). Let $\mathcal{H}_T = \{h_C \mid C : N(T) \to \mathcal{H}\}$. Also, let $\mathcal{H}_{\text{trees}} = \cup_{T \text{ is a tree for } k \text{ classes}} \mathcal{H}_T$. If $\mathcal{H}$ is the class of linear separators over $\mathbb{R}^d$, then for any tree T the ERM problem with respect to $\mathcal{H}_T$ can be solved efficiently in the realizable case. However, the ERM problem is NP-hard in the non-realizable case.

Error Correcting Output Codes (ECOC): An ECOC is a code $M \in \mathbb{R}^{k\times l}$ along with a bijection $\lambda : [k] \to [k]$. We sometimes identify λ with the identity function and M with (M, λ)[3]. Given a code M, and the result of l binary classifiers represented by a vector $u \in \{-1, 1\}^l$, the code selects a label via $\tilde{M} : \{-1, 1\}^l \to [k]$, defined by $\tilde{M}(u) = \lambda\left(\arg\max_{i\in[k]} \sum_{j=1}^{l} M_{ij}u_j\right)$. Given binary classifiers $h_1, \dots, h_l$ for each column in the code matrix, the code assigns to the instance $x \in \mathcal{X}$ the label $\tilde{M}(h_1(x), \dots, h_l(x))$. Let $\mathcal{H} \subset \{\pm 1\}^{\mathcal{X}}$ be a binary hypothesis class. Denote by

[3]The use of λ here allows us to later consider codes with random association of rows to labels.

$\mathcal{H}_M \subseteq [k]^{\mathcal{X}}$ the hypotheses class $\mathcal{H}_M = \{h : \mathcal{X} \to [k] \mid \exists (h_1, \dots, h_l) \in \mathcal{H}^l \text{ s.t. } \forall x \in \mathcal{X}, h(x) = \tilde{M}(h_1(x), \dots, h_l(x))\}$.

The *distance* of a binary code, denoted by $\delta(M)$ for $M \in \{\pm 1\}^{k \times l}$, is the minimal *hamming distance* between any two pairs of rows in the code matrix. Formally, the hamming distance between $u, v \in \{-1, +1\}^l$ is $\Delta_{\mathrm{h}}(u, v) = |\{r : u[r] \neq v[r]\}|$, and $\delta(M) = \min_{1 \leq i < j \leq k} \Delta_{\mathrm{h}}(M[i], M[j])$. The ECOC paradigm described in [Dietterich and Bakiri, 1995] proposes to choose a code with a large distance.

One vs. All (OvA) and All Pairs (AP): Let $\mathcal{H} \subset \{\pm 1\}^{\mathcal{X}}$ and $k \geq 2$. In the OvA method we train k binary problems, each of which discriminates between one class and the rest of the classes. In the AP approach all pairs of classes are compared to each other. This is formally defined as two ECOCs. Define $M^{\mathrm{OvA}} \in \mathbb{R}^{k \times k}$ to be the matrix whose (i, j) elements is 1 if $i = j$ and -1 if $i \neq j$. Then, the hypothesis class of OvA is $\mathcal{H}_{\mathrm{OvA}} = \mathcal{H}_{M^{\mathrm{OvA}}}$. For the AP method, let $M^{\mathrm{AP}} \in \mathbb{R}^{k \times \binom{k}{2}}$ be such that for all $i \in [k]$ and $1 \leq j < l \leq k$, the coordinate corresponding to row i and column (j, l) is defined to be -1 if $i = j$, 1 if $i = l$, and 0 otherwise. Then, the hypothesis class of AP is $\mathcal{H}_{\mathrm{AP}} = \mathcal{H}_{M^{\mathrm{AP}}}$.

Our analysis of the estimation error is based on results that bound the sample complexity of multiclass learning. The *sample complexity* of an algorithm A is the function m_A defined as follows: For $\epsilon, \delta > 0$, $m_A(\epsilon, \delta)$ is the smallest integer such that for every $m \geq m_A(\epsilon, \delta)$ and every distribution $\mathcal{D}$ on $\mathcal{X} \times \mathcal{Y}$, with probability of $> 1 - \delta$ over the choice of an i.i.d. sample S of size m,

$$\mathrm{Err}(A(S_m)) \leq \min_{h \in \mathcal{H}} \mathrm{Err}(h) + \epsilon \,. \qquad (2)$$

The first term on the right-hand side is the approximation error of $\mathcal{H}$. Therefore, the sample complexity is the number of examples required to ensure that the estimation error of A is at most ϵ (with high probability). We denote the sample complexity of a class $\mathcal{H}$ by $m_{\mathcal{H}}(\epsilon, \delta) = \inf_A m_A(\epsilon, \delta)$, where the infimum is taken over all learning algorithms.

To bound the sample complexity of a hypothesis class we rely on upper and lower bounds on the sample complexity in terms of two generalizations of the VC dimension for multiclass problems, called the *Graph dimension* and the *Natarajan dimension* and denoted $d_G(\mathcal{H})$ and $d_N(\mathcal{H})$. For completeness, these dimensions are formally defined in the appendix.

Theorem 2.1. *Daniely et al. [2011] For every hypothesis class $\mathcal{H}$, and for every ERM rule,*

$$\Omega\left(\frac{d_N(\mathcal{H}) + \ln(\frac{1}{\delta})}{\epsilon^2}\right) \leq m_{\mathcal{H}}(\epsilon, \delta) \leq m_{\mathit{ERM}}(\epsilon, \delta) \leq O\left(\frac{\min\{d_N(\mathcal{H}) \ln(|\mathcal{Y}|), d_G(\mathcal{H})\} + \ln(\frac{1}{\delta})}{\epsilon^2}\right)$$

We note that the constants in the O, Ω notations are universal.

3 Main Results

In Section 3.1 we analyze the sample complexity of the different hypothesis classes. We provide lower bounds on the Natarajan dimensions of the various hypothesis classes, thus concluding, in light of Theorem 2.1, a lower bound on the sample complexity of *any* algorithm. We also provide upper bounds on the graph dimensions of these hypothesis classes, yielding, by the same theorem, an upper bound on the estimation error of ERM. In Section 3.2 we analyze the approximation error of the different hypothesis classes.

3.1 Sample Complexity

Together with Theorem 2.1, the following theorems estimate, up to logarithmic factors, the sample complexity of the classes under consideration. We note that these theorems support the rule of thumb that the Natarajan and Graph dimensions are of the same order of the number of parameters. The first theorem shows that the sample complexity of MSVM depends on $\tilde{\Theta}(dk)$.

Theorem 3.1. $d(k-1) \leq d_N(\mathcal{L}) \leq d_G(\mathcal{L}) \leq O(dk \log(dk))$.

Next, we analyze the sample complexities of TC and ECOC. These methods rely on an underlying hypothesis class of binary classifiers. While our main focus is the case in which the binary hypothesis class is halfspaces over $\mathbb{R}^d$, the upper bounds on the sample complexity we derive below holds for any binary hypothesis class of VC dimension $d + 1$.

Theorem 3.2. *For every binary hypothesis class of VC dimension $d+1$, and for any tree T, $d_G(\mathcal{H}_T) \le d_G(\mathcal{H}_{\text{trees}}) \le O(dk\log(dk))$. If the underlying hypothesis class is halfspaces over $\mathbb{R}^d$, then also*

$$d(k-1) \le d_N(\mathcal{H}_T) \le d_G(\mathcal{H}_T) \le d_G(\mathcal{H}_{\text{trees}}) \le O(dk\log(dk)).$$

Theorems 3.1 and 3.2 improve results from Daniely et al. [2011] where it was shown that $\lfloor \frac{d}{2} \rfloor \lfloor \frac{k}{2} \rfloor \le d_N(\mathcal{L}) \le O(dk\log(dk))$, and for every tree $d_G(\mathcal{H}_T) \le O(dk\log(dk))$. Further it was shown that if $\mathcal{H}$ is the set of halfspaces over $\mathbb{R}^d$, then $\Omega\left(\frac{dk}{\log(k)}\right) \le d_N(\mathcal{H}_T)$.

We next turn to results for ECOC, and its special cases OvA and AP.

Theorem 3.3. *For every $M \in \mathbb{R}^{k\times l}$ and every binary hypothesis class of VC dimension d, $d_G(\mathcal{H}_M) \le O(dl\log(dl))$. Moreover, if $M \in \{\pm 1\}^{k\times l}$ and the underlying hypothesis class is halfspaces over $\mathbb{R}^d$, then*

$$d \cdot \delta(M)/2 \;\le\; d_N(\mathcal{H}_M) \;\le\; d_G(\mathcal{H}_M) \;\le\; O(dl\log(dl))\ .$$

We note if the code has a large distance, which is the case, for instance, in random codes, then $\delta(M) = \Omega(l)$. In this case, the bound is tight up to logarithmic factors.

Theorem 3.4. *For any binary hypothesis class of VC dimension d, $d_G(\mathcal{H}_{OvA}) \le O(dk\log(dk))$ and $d_G(\mathcal{H}_{AP}) \le O(dk^2\log(dk))$. If the underlying hypothesis class is halfspaces over $\mathbb{R}^d$ we also have:*

$$d(k-1) \le d_N(\mathcal{H}_{OvA}) \le d_G(\mathcal{H}_{OvA}) \le O(dk\log(dk)) \quad \textit{and}$$
$$d\binom{k-1}{2} \le d_N(\mathcal{H}_{AP}) \le d_G(\mathcal{H}_{AP}) \le O(dk^2\log(dk)).$$

3.2 Approximation error

We first show that the class $\mathcal{L}$ essentially contains $\mathcal{H}_{\text{OvA}}$ and $\mathcal{H}_T$ for any tree T, assuming, of course, that $\mathcal{H}$ is the class of halfspaces in $\mathbb{R}^d$. We find this result quite surprising, since the sample complexity of all of these classes is of the same order.

Theorem 3.5. *$\mathcal{L}$ essentially contains $\mathcal{H}_{\text{trees}}$ and $\mathcal{H}_{OvA}$. These inclusions are strict for $d \ge 2$ and $k \ge 3$.*

One might suggest that a small increase in the dimension would perhaps allow us to embed $\mathcal{L}$ in $\mathcal{H}_T$ for some tree T or for OvA. The next result shows that this is not the case.

Theorem 3.6. *Any embedding into a higher dimension that allows $\mathcal{H}_{OvA}$ or $\mathcal{H}_T$ (for some tree T for k classes) to essentially contain $\mathcal{L}$, necessarily embeds into a dimension of at least $\tilde{\Omega}(dk)$.*

The next theorem shows that the approximation error of AP is better than that of MSVM (and hence also better than OvA and TC). This is expected as the sample complexity of AP is considerably higher, and therefore we face the usual trade-off between approximation and estimation error.

Theorem 3.7. *$\mathcal{H}_{AP}$ essentially contains $\mathcal{L}$. Moreover, there is a constant $k^* > 0$, independent of d, such that the inclusion is strict for all $k \ge k^*$.*

For a random ECOC of length $o(k)$, it is easy to see that it does not contain MSVM, as MSVM has higher complexity. It is also not contained in MSVM, as it generates non-convex regions of labels.

We next derive absolute lower bounds on the approximation errors of ECOC and TC when $d \ll k$. Recall that both methods are built upon binary classifiers that should predict $h(x) = 1$ if the label of x is in L, for some $L \subset [k]$, and should predict $h(x) = -1$ if the label of x is not in L. As the following lemma shows, when the partition of $[k]$ into the two sets L and $[k] \setminus L$ is arbitrary and balanced, and $k \gg d$, such binary classifiers will almost always perform very poorly.

Lemma 3.8. *There exists a constant $C > 0$ for which the following holds. Let $\mathcal{H} \subseteq \{\pm 1\}^{\mathcal{X}}$ be any hypothesis class of VC-dimension d, let $\mu \in (0, 1/2]$, and let $\mathcal{D}$ be any distribution over $\mathcal{X} \times [k]$ such that $\forall i\ \mathbb{P}_{(x,y)\sim\mathcal{D}}(y = i) \le \frac{10}{k}$. Let $\phi : [k] \to \{\pm 1\}$ be a randomly chosen function which is sampled according to one of the following rules: (1) For each $i \in [k]$, each coordinate $\phi(i)$ is chosen independently from the other coordinates and $\mathbb{P}(\phi(i) = -1) = \mu$; or (2) ϕ is chosen uniformly among all functions satisfying $|\{i \in [k] : \phi(i) = -1\}| = \mu k$.*

Let $\mathcal{D}_\phi$ be the distribution over $\mathcal{X} \times \{\pm 1\}$ obtained by drawing (x, y) according to $\mathcal{D}$ and replacing it with $(x, \phi(y))$. Then, for any $\nu > 0$, if $k \geq C \cdot \left(\frac{d+\ln(\frac{1}{\delta})}{\nu^2}\right)$, then with probability of at least $1 - \delta$ over the choice of ϕ, the approximation error of $\mathcal{H}$ with respect to $\mathcal{D}_\phi$ will be at least $\mu - \nu$.

As the corollaries below show, Lemma 3.8 entails that when $k \gg d$, both random ECOCs with a small code length, and balanced trees with a random labeling of the leaves, are expected to perform very poorly.

Corollary 3.9. *There is a constant $C > 0$ for which the following holds. Let (T, λ) be a tree for k classes such that $\lambda : \text{leaf}(T) \to [k]$ is chosen uniformly at random. Denote by k_L and k_R the number of leaves of the left and right sub-trees (respectively) that descend from root, and let $\mu = \min\{\frac{k_1}{k}, \frac{k_2}{k}\}$. Let $\mathcal{H} \subseteq \{\pm 1\}^{\mathcal{X}}$ be a hypothesis class of VC-dimension d, let $\nu > 0$, and let $\mathcal{D}$ be any distribution over $\mathcal{X} \times [k]$ such that $\forall i \ \mathbb{P}_{(x,y)\sim\mathcal{D}}(y = i) \leq \frac{10}{k}$. Then, for $k \geq C \cdot \left(\frac{d+\ln(\frac{1}{\delta})}{\nu^2}\right)$, with probability of at least $1 - \delta$ over the choice of λ, the approximation error of $\mathcal{H}_T$ with respect to $\mathcal{D}$ is at least $\mu - \nu$.*

Corollary 3.10. *There is a constant $C > 0$ for which the following holds. Let (M, λ) be an ECOC where $M \in \mathbb{R}^{k \times l}$, and assume that the bijection $\lambda : [k] \to [k]$ is chosen uniformly at random. Let $\mathcal{H} \subseteq \{\pm 1\}^{\mathcal{X}}$ be a hypothesis class of VC-dimension d, let $\nu > 0$, and let $\mathcal{D}$ be any distribution over $\mathcal{X} \times [k]$ such that $\forall i \ \mathbb{P}_{(x,y)\sim\mathcal{D}}(y = i) \leq \frac{10}{k}$. Then, for $k \geq C \cdot \left(\frac{dl\log(dl)+\ln(\frac{1}{\delta})}{\nu^2}\right)$, with probability of at least $1 - \delta$ over the choice of λ, the approximation error of $\mathcal{H}_M$ with respect to $\mathcal{D}$ is at least $1/2 - \nu$.*

Note that the first corollary holds even if only the top level of the binary tree is balanced and splits the labels randomly to the left and the right sub-trees. The second corollary holds even if the code itself is not random (nor does it have to be binary), and only the association of rows with labels is random. In particular, if the length of the code is $O(\log(k))$, as suggested in Allwein et al. [2000], and the number of classes is $\tilde{\Omega}(d)$, then the code is expected to perform poorly.

For an ECOC with a matrix of length $\Omega(k)$ and $d = o(k)$, we do not have such a negative result as stated in Corollary 3.10. Nonetheless, Lemma 3.8 implies that the prediction of the binary classifiers when $d = o(k)$ is just slightly better than a random guess, thus it seems to indicate that the ECOC method will still perform poorly. Moreover, most current theoretical analyses of ECOC estimate the error of the learned multiclass hypothesis in terms of the average error of the binary classifiers. Alas, when the number of classes is large, Lemma 3.8 shows that this average will be close to $\frac{1}{2}$.

Finally, let us briefly discuss the tightness of Lemma 3.8. Let $x_1, \ldots, x_{d+1} \in \mathbb{R}^d$ be affinely independent and let $\mathcal{D}$ be the distribution over $\mathbb{R}^d \times [d+1]$ defined by $\mathbb{P}_{(x,y)\sim\mathcal{D}}((x, y) = (x_i, i)) = \frac{1}{d+1}$. Is is not hard to see that for every $\phi : [d + 1] \to \{\pm 1\}$, the approximation error of the class of half-spaces with respect to $\mathcal{D}_\phi$ is zero. Thus, in order to ensure a large approximation error *for every distribution*, the number of classes must be at least linear in the dimension, so in this sense, the lemma is tight. Yet, this example is very simple, since each class is concentrated on a single point and the points are linearly independent. It is possible that in real-world distributions, a large approximation error will be exhibited even when $k < d$.

We note that the phenomenon of a large approximation error, described in Corollaries 3.9 and 3.10, does not reproduce in the classes $\mathcal{L}$, $\mathcal{H}_{OvA}$ and $\mathcal{H}_{AP}$, since these classes are symmetric.

4 Proof Techniques

Due to lack of space, the proofs for all the results stated above are provided in the appendix. In this section we give a brief description of our main proof techniques.

Most of our proofs for the estimation error results, stated in Section 3.1, are based on a similar method which we now describe. Let $L : \{\pm 1\}^l \to [k]$ be a multiclass-to-binary reduction (e.g., a tree), and for $\mathcal{H} \subseteq \{\pm 1\}^{\mathcal{X}}$, denote $L(\mathcal{H}) = \{x \mapsto L(h_1(x), \ldots, h_l(x)) \mid h_1, \ldots, h_l \in \mathcal{H}\}$. Our upper bounds for $d_G(L(\mathcal{H}))$ are mostly based on the following simple lemma.

Lemma 4.1. *If $\text{VC}(\mathcal{H}) = d$ then $d_G(L(\mathcal{H})) = O(ld\ln(ld))$.*

The technique for the lower bound on $d_N(L(\mathcal{W}))$ when $\mathcal{W}$ is the class of halfspaces in $\mathbb{R}^d$ is more involved, and quite general. We consider a binary hypothesis class $\mathcal{G} \subseteq \{\pm 1\}^{[d]\times[l]}$ which consists of functions having an arbitrary behaviour over $[d] \times \{i\}$, and a very uniform behaviour on other inputs (such as mapping all other inputs to a constant). We show that $L(\mathcal{G})$ N-shatters the set $[d] \times [l]$. Since $\mathcal{G}$ is quite simple, this is usually not very hard to show. Finally, we show that the class of halfspaces is richer than $\mathcal{G}$, in the sense that the inputs to $\mathcal{G}$ can be mapped to points in $\mathbb{R}^d$ such that the functions of $\mathcal{G}$ can be mapped to halfspaces. We conclude that $d_N(L(\mathcal{W})) \geq d_N(L(\mathcal{G}))$.

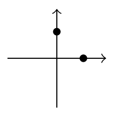
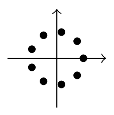
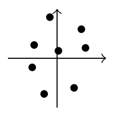
To prove the approximation error lower bounds stated in Section 3.2, we use the techniques of VC theory in an unconventional way. The idea of this proof is as follows: Using a uniform convergence argument based on the VC dimension of the binary hypothesis class, we show that there exists a small labeled sample S whose approximation error for the hypothesis class is close to the approximation error for the distribution, for all possible label mappings. This allows us to restrict our attention to a finite set of hypotheses, by their restriction to the sample. For these hypotheses, we show that with high probability over the choice of label mapping, the approximation error on the sample is high. A union bound on the finite set of possible hypotheses shows that the approximation error on the distribution will be high, with high probability over the choice of the label mapping.

5 Implications

The first immediate implication of our results is that whenever the number of examples in the training set is $\tilde{\Omega}(dk)$, MSVM should be preferred to OvA and TC. This is certainly true if the hypothesis class of MSVM, $\mathcal{L}$, has a zero approximation error (the realizable case), since the ERM is then solvable with respect to $\mathcal{L}$. Note that since the inclusions given in Theorem 3.5 are strict, there are cases where the data is realizable with MSVM but not with $\mathcal{H}_{\text{OvA}}$ or with respect to any tree.

In the non-realizable case, implementing the ERM is intractable for all of these methods. Nonetheless, for each method there are reasonable heuristics to approximate the ERM, which should work well when the approximation error is small. Therefore, we believe that MSVM should be the method of choice in this case as well due to its lower approximation error. However, variations in the optimality of algorithms for different hypothesis classes should also be taken into account in this analysis. We leave this detailed analysis of specific training heuristics for future work. Our analysis also implies that it is highly unrecommended to use TC with a randomly selected λ or ECOC with a random code whenever $k > d$. Finally, when the number of examples is much larger than dk^2, the analysis implies that it is better to choose the AP approach.

To conclude this section, we illustrate the relative performance of MSVM, OvA, TC, and ECOC, by considering the simplistic case where $d = 2$, and each class is concentrated on a single point in $\mathbb{R}^2$. In the leftmost graph below, there are two classes in $\mathbb{R}^2$, and the approximation error of all algorithms is zero. In the middle graph, there are 9 classes ordered on the unit circle of $\mathbb{R}^2$. Here, both MSVM and OvA have a zero approximation error, but the error of TC and of ECOC with a random code will most likely be large. In the rightmost graph, we chose random points in $\mathbb{R}^2$. MSVM still has a zero approximation error. However, OvA cannot learn the binary problem of distinguishing between the middle point and the rest of the points and hence has a larger approximation error.

			
MSVM	✓	✓	✓
OvA	✓	✓	✗
TC/ECOC	✓	✗	✗

Acknowledgements: Shai Shalev-Shwartz was supported by the John S. Cohen Senior Lectureship in Computer Science. Amit Daniely is a recipient of the Google Europe Fellowship in Learning Theory, and this research is supported in part by this Google Fellowship.

References

E. L. Allwein, R.E. Schapire, and Y. Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of Machine Learning Research*, 1:113–141, 2000.

S. Ben-David, N. Cesa-Bianchi, D. Haussler, and P. Long. Characterizations of learnability for classes of $\{0, \ldots, n\}$-valued functions. *Journal of Computer and System Sciences*, 50:74–86, 1995.

S. Bengio, J. Weston, and D. Grangier. Label embedding trees for large multi-class tasks. In *NIPS*, 2011.

A. Beygelzimer, J. Langford, and P. Ravikumar. Multiclass classification with filter trees. *Preprint, June*, 2007.

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

A. Daniely, S. Sabato, S. Ben-David, and S. Shalev-Shwartz. Multiclass learnability and the erm principle. In *COLT*, 2011.

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

Trevor Hastie and Robert Tibshirani. Classification by pairwise coupling. *The Annals of Statistics*, 26(1):451–471, 1998.

Ryan Rifkin and Aldebaro Klautau. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5:101–141, 2004.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning internal representations by error propagation. In David E. Rumelhart and James L. McClelland, editors, *Parallel Distributed Processing – Explorations in the Microstructure of Cognition*, chapter 8, pages 318–362. MIT Press, 1986.

G. Takacs. *Convex polyhedron learning and its applications*. PhD thesis, Budapest University of Technology and Economics, 2009.

V. N. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

J. Weston and C. Watkins. Support vector machines for multi-class pattern recognition. In *Proceedings of the Seventh European Symposium on Artificial Neural Networks*, April 1999.