Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Label Efficient Learning by Exploiting Multi-Class Output Codes

Maria Florina Balcan
School of Computer Science
Carnegie Mellon University
ninamf@cs.cmu.edu

Travis Dick
School of Computer Science
Carnegie Mellon University
tdick@cs.cmu.edu

Yishay Mansour
Microsoft Research and
Tel Aviv University
mansour@tau.ac.il

Abstract

We present a new perspective on the popular multi-class algorithmic techniques of one-vs-all and error correcting output codes. Rather than studying the behavior of these techniques for supervised learning, we establish a connection between the success of these methods and the existence of label-efficient learning procedures. We show that in both the realizable and agnostic cases, if output codes are successful at learning from labeled data, they implicitly assume structure on how the classes are related. By making that structure explicit, we design learning algorithms to recover the classes with low label complexity. We provide results for the commonly studied cases of one-vs-all learning and when the codewords of the classes are well separated. We additionally consider the more challenging case where the codewords are not well separated, but satisfy a boundary features condition that captures the natural intuition that every bit of the codewords should be significant.

1 Introduction

Motivation: Large scale multi-class learning problems with an abundance of unlabeled data are ubiquitous in modern machine learning. For example, an in-home assistive robot needs to learn to recognize common household objects, familiar faces, facial expressions, gestures, and so on in order to be useful. Such a robot can acquire large amounts of unlabeled training data simply by observing its surroundings, but it would be prohibitively time consuming (and frustrating) to ask its owner to annotate any significant portion of this raw data. More generally, in many modern learning problems we often have easy and cheap access to large quantities of unlabeled training data (e.g., on the internet) but obtaining high-quality labeled examples is relatively expensive. More examples include text understanding, recommendation systems, or wearable computing (Thrun 1996; Thrun and Mitchell 1995b; 1995a; Mitchell et al. 2015). The scarcity of labeled data is especially pronounced in problems with many classes, since supervised learning algorithms require labeled examples from every class. In such settings, algorithms should make the best use of unlabeled data in order to minimize the need for expensive labeled examples.

Overview: We approach label-efficient learning by making the implicit assumptions of popular multi-class learning algorithms explicit and showing that they can also be exploited when learning from limited labeled data. We focus on a family of techniques called *output codes* that work by decomposing a given multi-class problem into a collection of binary classification tasks (Mohri, Rostamizadeh, and Talwalkar 2012; Dietterich and Bakiri 1995; Langford and Beygelzimer 2005; Beygelzimer, Langford, and Ravikumar 2009). The novelty of our results is to show that the existence of various low-error output codes constrains the distribution of unlabeled data in ways that can be exploited to reduce the label complexity of learning. We consider both the consistent setting, where the output code achieves zero error, and the agnostic setting, where the goal is to compete with the best output code. The most well known output code technique is one-vs-all learning, where we learn one binary classifier for distinguishing each class from the union of the rest. When output codes are successful at learning from labeled data, it often implies geometric structure in the underlying problem. For example, if it is possible to learn an accurate one-vs-all classifier with linear separators, it implies that no three classes can be collinear, since then it would be impossible for a single linear separator to distinguish the middle class from the union of the others. In this work, we exploit this implicitly assumed structure to design label-efficient algorithms for the commonly assumed cases of one-vs-all and error correcting output codes, as well as a novel boundary features condition that captures the intuition that every bit of the codewords should be significant.

Our results: Before discussing our results, we briefly review the output code methodology. For a problem with L classes, a domain expert designs a code matrix $C \in \{\pm 1\}^{L \times m}$ where each column partitions the classes into two meaningful groups. The number of columns m is chosen by the domain expert. For example, when recognizing household objects we could use the following true/false questions to define the partitions: “is it made of wood?”, “is it sharp?”, “does it have legs?”, “should I sit on it?”, and so on. Each row of the code matrix describes one of the classes in terms of these partitions (or semantic features). For example, the class “table” could be described by the vector $(+1, -1, +1, -1)$, which is called the class’ codeword. Once the code matrix has been designed, we train an output code by learning a binary classifier for each of the binary partitions (e.g., predicting whether an object is made of wood or not). To predict the class of a new example, we predict its codeword in $\{\pm 1\}^m$ and output the class with the nearest codeword under the Hamming distance. Two popular special cases of output codes are one-vs-all learning, where C is the identity matrix (with -1 in the off-diagonal entries), and error correcting output codes, where the Hamming distance between the codewords is large.

In each of our results we assume that there exists a consistent or low-error linear output code classifier and we impose constraints on the code matrix and the distribution that generates the data. We present algorithms and analysis techniques for a wide range of different conditions on the code matrix and data distribution to showcase the variety of implicit structures that can be exploited. For the code matrix, we consider the case when the codewords are well separated (i.e., the output code is error correcting), the case of one-vs-all (where the code matrix is the identity), and a natural boundary features condition. These conditions can loosely be compared in terms of the Hamming distance between codewords. In the case of error correcting output codes, the distance between codewords is large (at least $d + 1$ when the data is d-dimensional), in one-vs-all the distance is always exactly 2, and finally in the boundary features condition the distance can be as small as 1. In the latter cases, the lower Hamming distance requirement is balanced by other structure in the code matrix. For the distribution, we either assume that the data density function satisfies a thick level set condition or that the density is upper and lower bounded on its support. Both regularity conditions are used to ensure that the geometric structure implied by the consistent output code will be recoverable based on a sample of data.

Error correcting output codes: We first showcase how to exploit the implicit structure assumed by the commonly used and natural case of linear output codes where the Hamming distance between codewords is large. In practice, output codes are designed to have this property in order to be robust to prediction errors for the binary classification tasks (Dietterich and Bakiri 1995). We suppose that the output code makes at most β errors when predicting codewords and has codewords with Hamming distance at least $2\beta + d + 1$ in a d-dimensional problem. The key insight is that when the code words are well separated, this implies that points belonging to different classes must be geometrically separated as well. This suggests that tight clusters of data will be label-homogeneous, so we should be able to learn an accurate classifier using only a small number of label queries per cluster. The main technical challenge is to show that our clustering algorithm will not produce too many clusters (in order to keep the label complexity controlled), and that with high probability, a new sample from the distribution will have the same label as its nearest cluster. We show that when the data density satisfies a thick-level set condition (requiring that its level sets do not have bridges or cusps that are too thin), then a single-linkage clustering algorithm can be used to recover a small number of label-homogeneous clusters.

One-vs-all: Next, we consider the classic one-vs-all setting for data in the unit ball. This is an interesting setting because of the popularity of one-vs-all classification and because it significantly relaxes the assumption that the codewords are well separated (in a one-vs-all classifier, the Hamming distance between codewords is exactly 2). The main challenge in this setting is that there need not be a margin between classes and a simple single-linkage style clustering might group multiple classes into the same cluster. To overcome this challenge, we show that the classes are probabilistically separated in the following sense: after projecting onto the surface of the unit ball, the level sets of the projected density are label-homogeneous. Equivalently, the high-density regions belonging to different classes must be separated by low-density regions. We exploit this structure by estimating the connected components of the ϵ level set using a robust single-linkage clustering algorithm.

The boundary features condition: We introduce an interesting and natural condition on the code matrix capturing the intuition that every binary learning task should be significant. This condition has the weakest separation requirement, allowing the codewords to have a Hamming distance of only 1. This setting is our most challenging, since it allows for the classes to be very well connected to one another, which prevents clustering or level set estimation from being used to find a small number of label-homogeneous clusters. Nevertheless, we show that the implicit geometric structure implied by the output code can be exploited to learn using a small number of label queries. In this case, rather than clustering the unlabeled sample, we apply a novel hyperplane-detection algorithm that uses the *absence* of data to learn local information about the boundaries between classes. We then use the implicit structure of the output code to extend these local boundaries into a globally accurate classifier.

Agnostic Setting: Finally, we show that our results for all three settings can be extended to an agnostic learning scenarios, where we do not assume that there exists a consistent output code classifier and the goal is to compete with the best linear output code.

Our results show an interesting trend: when linear output codes are able to learn from labeled data, it is possible to exploit the same underlying structure in the problem to learn using a small number of label requests. Our results hold under several natural assumptions on the output code and general conditions on the data distribution, and employ both clustering and hyperplane detection strategies to reduce the label complexity of learning.

Full proofs of all our results can be found in the full version of the paper (Balcan, Dick, and Mansour 2016), while we present the modeling and technical flow of ideas here.

2 Related Work

Reduction to binary classification is one of the most widely used techniques in applied machine learning for multi-class problems. Indeed, the one-vs-all, one-vs-one, and the error correcting output code approaches (Dietterich and Bakiri 1995) all follow this structure (Mohri, Rostamizadeh, and Talwalkar 2012; Langford and Beygelzimer 2005; Beygelzimer, Langford, and Ravikumar 2009; Daniely, Schapira, and Shahaf 2012; Allwein, Schapire, and Singer 2000).

There is no prior work providing error bounds for output codes using unlabeled data and interaction. There has been a long line of work for providing provable bounds for semi-supervised learning (Balcan, Blum, and Yang 2004; Balcan and Blum 2010; Blum and Mitchell 1998; Chapelle, Schlkopf, and Zien 2010) and active learning (Balcan, Beygelzimer, and Lanford 2006; Dasgupta 2011; Balcan and Urner 2015; Hanneke 2014). These works provide bounds on the benefits of unlabeled data and interaction for significantly different semi-supervised and active learning methods that are based different assumptions, often focusing on binary classification, thus the results are largely incomparable. Another line of recent work considers the multi-class setting and uses unlabeled data to consistently estimate the risk of classifiers when the data is generated from a known family of models (Donmez, Lebanon, and Balasubramanian 2010; Balasubramanian, Donmez, and Lebanon 2011a; 2011b). Their results do not immediately imply learning algorithms and they consider generative assumptions, while in contrast our work explicitly designs learning algorithms under commonly used discriminative assumptions.

Another work related to ours is that of Balcan, Blum, and Mansour (2013), where labels are recovered from unlabeled data. The main tool that they use in order to recover the labels is the assumption that there are multiple views and an underlying ontology that are known, and restrict the possible labeling. Similarly, Steinhardt and Liang (2016) show how to use the method of moments to estimate the risk of a model from unlabeled data under the assumption that the data has three independent views. Our work is more widely applicable, since it applies when we have only a single view.

The output-code formalism is also used by Palatucci et al. (2009) for the purpose of zero shot learning. They demonstrate that it is possible to exploit the semantic relationships encoded in the code matrix to learn a classifier from labeled data that can predict accurately even classes that *did not appear in the training set*. These techniques make very similar assumptions to our work but require that the code matrix C is known and the problem that they solve is different.

3 Preliminaries

We consider multiclass learning problems over an instance space $\mathcal{X} \subset \mathbb{R}^d$ where each point is labeled by $f^* : \mathcal{X} \to \{1, \ldots, L\}$ to one out of L classes and the probability of observing each outcome $x \in \mathcal{X}$ is determined by a data distribution P on $\mathcal{X}$. The density function of P is denoted by $p : \mathcal{X} \to [0, \infty)$. In all of our results we assume that there exists a consistent (but unknown) linear output-code classifier defined by a code matrix $C \in \{\pm 1\}^{L \times m}$ and m linear separators $h_1, \ldots, h_m$. We denote class i's code word by C_i and define $h(x) = (\mathrm{sign}(h_1(x)), \ldots, \mathrm{sign}(h_m(x)))$ to be the code word for point x. We let $d_{\mathrm{Ham}}(c, c')$ denote the Hamming distance between any codewords $c, c' \in \{\pm 1\}^m$. To simplify notation, we assume that $\mathcal{X}$ has diameter ≤ 1.

Our goal is to learn a hypothesis $\hat{f} : \mathcal{X} \to \{1, \ldots, L\}$ minimizing $\mathrm{err}_P(\hat{f}) = \Pr_{X \sim P}(\hat{f}(x) \neq f^*(x))$ from an unlabeled sample drawn from the data distribution P together with a small set of actively queried labeled examples.

Finally, we use the following notation throughout the paper: for any set A in a metric space $(\mathcal{X}, d)$, the σ-interior of A is the set $\mathrm{int}_\sigma(A) = \{x \in A : B(x, \sigma) \subset A\}$. The notation $\tilde{O}(\cdot)$ suppresses logarithmic terms.

4 Error Correcting Output Codes

We first consider the implicit structure when there exists a consistent linear *error correcting* output code classifier:

Assumption 1. *There exists a code matrix $C \in \{\pm 1\}^{L \times m}$ and linear functions $h_1, \ldots, h_m$ such that: (1) there exists $\beta \ge 0$ such that any point x from class y satisfies $d_{\mathrm{Ham}}(h(x), C_y) \le \beta$, (2) The Hamming distance between the codewords of C is at least $2\beta + d + 1$; and (3) at most d of the separators $h_1, \ldots, h_m$ intersect at any point.*

Part (1) of this condition is a bound on the number of linear separators that can make a mistake when the output code predicts the codeword of a new example, part (2) formalizes the requirement of having well separated codewords, and part (3) requires that the hyperplanes be in general position, a very mild condition that can be satisfied by adding an arbitrarily small perturbation to the linear separators.

Despite being very natural, Assumption 1 conveniently implies that there is a margin between classes.

Lemma 1. *Under Assumption 1, there exists $g > 0$ s.t. any points x and x' with different labels satisfy $\|x - x'\| > g$.*

Lemma 1 suggests a clustering based approach. Any single-linkage style clustering algorithm that only merges clusters closer than distance g will produce label-homogeneous clusters, so we can query the label of a single point per cluster. See Algorithm 1 for pseudocode.

```
Input: Sample $S = \{x_1, \ldots, x_n\}$, $r_c > 0$, $\epsilon > 0$.
1. Let $\{\hat{A}_1\}_{i=1}^N$ be the connected components of the
   graph $G$ with vertex set $S$ and an edge between $x_i$
   and $x_j$ if $\|x_i - x_j\| \le r_c$.
2. In decreasing order of size, query the label of each
   $\hat{A}_i$ until $\le \frac{\epsilon}{4} n$ points belong to unlabeled clusters.
3. Output $\hat{f}(x)$ = label of nearest labeled cluster to $x$.
```

Algorithm 1: Single-linkage learning.

In order to get a meaningful reduction in label complexity, we need to ensure that most of the samples belong to a small number of clusters. For this purpose, we borrow the following very general and interesting thick level set condition from Steinwart (2015): a density function p has C-thick level sets if there exists a level $\lambda_0 > 0$ and a radius $\sigma_0 > 0$ such that for every level $\lambda \le \lambda_0$ and radius $\sigma < \sigma_0$, (1) the σ-interior of $\{p \ge \lambda\}$ is non-empty and (2) every point in $\{p \ge \lambda\}$ is at most distance $C\sigma$ from the σ-interior. This condition elegantly characterizes a large family of distributions for which single-linkage style clustering algorithms succeed at recovering the high-density clusters and only rules out distributions whose level sets have bridges or cusps that are too thin. The thickness parameter C measures how pointed the boundary of the level sets of p can be. For example, in $\mathbb{R}^d$ if the level set of p is a ball then $C = 1$, while if the level set is a cube, then $C = \sqrt{d}$.

Using the thick level set condition to guarantee that our clustering algorithm will not subdivide the high-density clusters of p, we obtain the following result for Algorithm 1.

Theorem 1. *Suppose that Assumption 1 holds and that the data distribution has C-thick level sets. For any target error $\epsilon > 0$, let N be the number of connected components of $\{p \geq \epsilon/(2\operatorname{Vol}(K))\}$. With probability at least $1-\delta$, running Algorithm 1 with parameter $r_c = g$ on an unlabeled sample of size $n = \tilde{O}(\frac{1}{\epsilon^2}((4C)^{2d}d^{d+1}/r_c^{2d} + N))$ will query at most N labels and output a classifier with error at most ϵ.*

The exponential dependence on the dimension in Theorem 1 is needed to ensure the sample S will be a fine covering of the level set of p w.h.p, which guarantees that Algorithm 1 will not subdivide its connected components into smaller clusters. When the data has low intrinsic dimensionality, the unlabeled sample complexity is only exponential in the intrinsic dimension. In particular, under the common assumption that the distribution is a doubling measure, the unlabeled sample complexity is exponential only in the doubling dimension. A probability measure P has doubling dimension D if for all points x in the support of P and every radius $r > 0$, we have that $P(B(x, 2r)) \leq 2^D P(B(x, r))$ (see, for example, (Dasgupta and Sinha 2013)).

Theorem 2. *Suppose that Assumption 1 holds the data distribution P has doubling dimension D, and the support of P has N connected components. With probability at least $1-\delta$, running Algorithm 1 with parameter $r_c = g$ on a sample of size $n = \tilde{O}\big(d/r_c^{2D} + N/\epsilon^2\big)$ will query at most N labels and have error at most ϵ.*

The unlabeled sample complexity and parameter settings in Theorem 1 depend on the gap g. Such a scale parameter must appear in our results, since Assumption 1 is scale-invariant, yet our algorithm exploits scale-dependent geometric properties of the problem. If we have a conservatively small estimate $\hat{g} \leq g$, then the conclusion of Theorem 1 and Theorem 2 continue to hold if the connection radius and unlabeled sample complexity are set using the estimate $\hat{g}$. Nevertheless, in some cases we may not have an estimate of g. The following result shows that if we have an estimate of the number of high-density clusters, and these clusters have roughly balanced probability mass, then we are still able to take advantage of the geometric structure even when the distance g is unknown. The idea is to construct a hierarchical clustering of S using single linkage, and then to use a small number of label queries to find a good pruning. See Algorithm 2 for pseudocode.

Theorem 3. *Suppose Assumption 1 holds and the density p has C-thick level sets. For any $0 < \epsilon \leq 1/2$, suppose that $\{A_i\}_{i=1}^N$ are the connected components of $\{p \geq \epsilon/(2\operatorname{Vol}(K))\}$ and for some $\alpha \geq 1$ we have $P(A_i) \leq \alpha P(A_j)$ for all i, j. With probability $\geq 1-\delta$, running Algorithm 2 with $t = \tilde{O}(\alpha N)$ on an unlabeled sample of size $n = \tilde{O}(\frac{1}{\epsilon^2}(C^{2d}d^{d+1}/g^{2d} + N))$ will have error $\leq \epsilon$.*

```
Input: Sample $S = \{x_1, \ldots, x_n\}$, $t \in \mathbb{N}$.
1. Let $T$ be the hierarchical clustering of $S$ obtained by single-linkage.
2. Query the labels of a random subset of $S$ of size $t$.
3. Let $\{\hat{B}_i\}_{i=1}^M$ be the coarsest pruning of $T$ such that each $\hat{B}_i$ contains labels from one class.
4. Output $\hat{f}(x)$ = label of nearest $\hat{B}_i$ to $x$.
```

Algorithm 2: Hierarchical single-linkage learning.

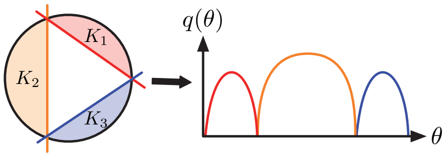

Figure 1: An example problem satisfying Assumption 2 and the projected density q when the density p is uniform on K.

In this section we showed that when there exists linear error correcting correcting output code with low error, then it is possible to reduce the label complexity of learning to the number of high-density clusters, which are the connected components of $\{p \geq \epsilon\}$.

5 One-Versus-All on the Unit Ball

In this section we show that even when the codewords are not well separated, we can still exploit the implicit structure of output codes to reduce the label complexity of learning by clustering the data. Specifically, we consider the implicit structure of a linear one-vs-all classifier over the unit ball:

Assumption 2. *The instance space $\mathcal{X}$ is the unit ball and there exist L linear separators $h_1, \ldots, h_L$ such that: (1) point x belongs to class i iff $h_i(x) > 0$, and (2) for all i, $h_i(x) = w_i^\top x - b_i$ with $\|w_i\| = 1$ and $b_i \geq b_{\min} > 0$.*

See Figure 1 for an example problem satisfying this condition. Since a one-vs-all classifier is an output code where the code matrix is the identity, the Hamming distance between any pair of codewords is exactly 2. In this setting we do not have a result similar to Lemma 1 ensuring geometric separation of the classes. Instead, we exploit the one-vs-all structure to show the classes are probabilistically separated and use a robust clustering algorithm.

As before, we study this problem under a mild constraint on the data distribution. For each class i, denote the set of points in class i by $K_i = \{x : \|x\| \leq 1, h_i(x) > 0\}$ and let $K = \bigcup_{i=1}^L K_i$. In this section, we assume that the density p is supported on K with upper and lower bounds:

Assumption 3. *There exist constants $0 < c_{\mathrm{lb}} \leq c_{\mathrm{ub}}$ s.t. for $x \in K$ we have $c_{\mathrm{lb}} \leq p(x) \leq c_{\mathrm{ub}}$ and otherwise $p(x) = 0$.*

This distributional constraint is quite general: it only requires that the density is supported on examples for which exactly one linear separators claim the point is positive and the density does not take extreme values.

Our algorithm for this setting projects the data onto the unit sphere $\mathcal{S}^{d-1} = \{x \in \mathbb{R}^d : \|x\| = 1\}$ followed by a robust clustering algorithm. The projection does not introduce errors, since each linear separator carves out a spherical cap for its class, and no two class caps overlap. Given that no class contains the origin, an examples label depends only on its projection to the sphere. We show that projecting to the sphere has the useful property that the projected density goes to zero at the boundary of the classes, leading to probabilistic separation. Algorithm 3 gives pseudocode, using the notation $\theta(u, v)$ for the angle between u and v and $V^d(r)$ for the probability that a uniformly random sample from $\mathcal{S}^{d-1}$ lands in a given spherical cap of angular radius r.

```
Input: Sample $S = \{x_1, \dots, x_n\}$, $r_c > 0$, $\epsilon > 0$.
1. Define $r_a = r_c/2$ and $\tau = \frac{c_{\text{lb}}}{2c_{\text{ub}}} V^d(r_a)\epsilon$.
2. Let $v_i = \frac{x_i}{\|x_i\|}$ be the projection of $x_i$ to the sphere.
3. Mark $v_i$ active if $|\{v_j : \theta(v_i, v_j) \le r_a\}| \ge \tau n$ and
   inactive otherwise for $i \in [n]$.
4. Let $\hat{A}_1, \dots, \hat{A}_N$ be the connected components of
   the graph $G$ whose vertices are the active $v_i$ with an
   edge between $v_i$ and $v_j$ if $\theta(v_i, v_j) < r_c$.
5. In decreasing order of size, query the label of each
   $\hat{A}_i$ until $\le \frac{\epsilon}{4}n$ points belong to unlabeled clusters.
6. Output $\hat{f}(x)$ = label of nearest cluster to $x/\|x\|$.
```

Algorithm 3: Robust single-linkage learning.

Our first result characterizes the density of the projected data (defined relative to the uniform distribution on $\mathcal{S}^{d-1}$).

Lemma 2. *Suppose Assumptions 2 and 3 hold and let $q : \mathcal{S}^{d-1} \to [0, \infty)$ be the density function of the data projected onto the unit sphere. Then $q_{\text{lb}}(v) \le q(v) \le q_{\text{ub}}(v)$, where*

$$q_{\text{lb}}(v) = \begin{cases} c_{\text{lb}} d v_d (1 - (b_i / w_i^\top v)^d) & \textit{if } v \in K_i \\ 0 & \textit{otherwise}, \end{cases}$$

and $q_{\text{ub}}(v) = c_{\text{ub}}/c_{\text{lb}} \cdot q_{\text{lb}}(v)$, where v_d is the volume of the unit ball in d dimensions.

Both bounds are defined piecewise with one piece for each class. Restricted to class i, both $q_{\text{lb}}(v)$ and $q_{\text{ub}}(v)$ are decreasing functions of $\theta(w_i, v)$, which implies that their λ-level sets are spherical caps. Therefore, each class contributes one large connected component to the level set of q that is roughly a spherical cap centered at the point w_i and the density of q goes to zero at the boundary of each class. Our main result is as follows:

Theorem 4. *Suppose Assumptions 2 and 3 hold and that f^* is consistent. There exists an r_c satisfying $r_c = \Omega(\epsilon c_{\text{lb}}/(c_{\text{ub}}^2 b_{\min}))$ such that with probability at least $1 - \delta$, running Algorithm 3 with parameter r_c on an unlabeled sample of size $n = \tilde{O}((c_{\text{ub}}^4 d/(\epsilon^2 c_{\text{lb}}^2 b_{\min}^2))^d)$ will query at most L labels and output a classifier with error at most ϵ.*

If the scale parameter $b_{\min}$ is unknown, Theorem 4 continues to hold if we use an underestimate $\widehat{b_{\min}} \le b_{\min}$.

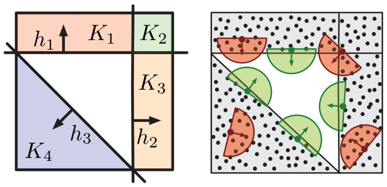

Figure 2: Left: An example boundary features problem. The shaded regions correspond to the four classes. Right: Example half-balls that pass or fail the test in step (2b) of Algorithm 4. The four innermost half-balls are accepted.

There are two main differences between the sample complexity of Theorem 4 and the results from Section 4. First, the unlabeled sample complexity now has an ϵ^{-2d} dependence, rather than only ϵ^{-2}. This is because the distance between the connected components of $\{p \ge \epsilon\}$ goes to zero (in the worst case) as $\epsilon \to 0$, so our algorithm must be able to detect low-density regions of small width. In contrast, Lemma 1 allowed us to establish a non-diminishing gap $g > 0$ between the classes when the codewords were well separated. On the other hand, the label complexity in this setting is better, scaling with L instead of N, since we are able to establish that each class will have one very large cluster containing nearly all of its data.

6 The Boundary Features Condition

In this section we introduce a novel condition on the code matrix called the boundary features condition that captures the intuition that every binary classification task should be significant. Assumption 4 formalizes this intuition.

Assumption 4. *There exists a code matrix $C \in \{\pm 1\}^{L \times m}$, linear functions $h_1, \dots, h_m$, and a scale parameter $R > 0$ so that: (1) for any point x in class y, we have $h(x) = C_y$; (2) for each h_j, there exists a class i such that negating the j^{th} entry of C_i produces a codeword C_i' not in C and there exists a point x on the hyperplane $h_j = 0$ such that every point in $B(x, R)$ has either code word C_i or C_i'; and (3) any pair of points $x, x' \in \mathcal{X}$ such that $h(x)$ and $h(x')$ are not codewords in C and $h(x) \ne h(x')$ must have $\|x - x'\| \ge R$.*

Part (1) requires that the output code classifier is consistent, part (2) guarantees that every linear separator h_j separates at least one class i from a region of space that does not belong to any class, and part (3) requires that points with codewords not in the code matrix must either have the same codeword or be separated by distance R. Part (3) simplifies both our algorithm and analysis and is trivially satisfied in cases where all points in $\mathcal{X}$ that do not belong to any class have the same codeword, as in the one-vs-all setting.

This setting is more challenging than in the previous sections because we are unable to apply clustering-based learning strategies. For example, if the Hamming distance between a pair of codewords is one, then one of the linear separators h_j forms a shared boundary between those classes, potentially allowing them to be well connected.

Instead, our algorithm uses the *absence* of data to directly learn the linear separators $h_1, \dots, h_m$. It searches for balls of radius r whose centers are sample points such that one half of the ball contains very few samples. If a half-ball contains few sample points, then it must be mostly disjoint from the set K. But since its center belongs to the set K, this means that the hyperplane defining the half-ball is a good approximation one of the true hyperplanes. See Figure 2 for examples of half-balls that would pass and fail this test. The collection H of hyperplanes obtained in this way partition the space into cells. Our algorithm queries the labels of the cells containing the most sample points and classifies test points based on the label of their cell in the partition (and if the label is unknown, we output a random label). Pseudocode is given in Algorithm 4 using the following notation: for any center $x \in \mathcal{X}$, radius $r \geq 0$, and direction $w \in \mathcal{S}^{d-1}$, let $B^{1/2}(x, r, w) = \{y \in B(x, r) : w^\top (y - x) > 0\}$ and define $p^{1/2}(r) = \frac{1}{2} c_{\text{lb}} r^d v_d$.

```
Input: Sample $S = \{x_1, \dots, x_n\}$, $r > 0$, $\tau > 0$.
1. Initialize set of candidate hyperplanes $H = \emptyset$.
2. For all samples $\hat{x} \in S$ with $B(\hat{x}, r) \subset \mathcal{X}$:
    (a) Let $\hat{w} = \operatorname{argmin}_{w \in \mathcal{S}^{d-1}} |B^{1/2}(\hat{x}, r, w) \cap S|$.
    (b) If $|B^{1/2}(\hat{x}, r, \hat{w}) \cap S|/n < \tau$, add $(\hat{x}, \hat{w})$ to $H$.
3. Let $\{\hat{C}_i\}_{i=1}^N$ be the partitioning of $\mathcal{X}$ induced by $H$.
4. Query the label of the $L$ cells with the most samples.
5. Output $\hat{f}(x)$ = label of $C_i$ containing $x$.
```

Algorithm 4: Plane-detection algorithm.

Theorem 5. *Suppose Assumptions 3 and 4 hold. For any desired error* $\epsilon > 0$*, with probability at least* $1 - \delta$*, running Algorithm 4 with parameters* $r = R/2$ *and* $\tau = \alpha p^{1/2}(r)/2$ *for a known constant* α *on on a sample of size* $n = \tilde{O}(dm^2 c_{\text{ub}}^2 R^d / (c_{\text{lb}}^2 \epsilon^4))$ *will have error at most* ϵ.

If the scale parameter R is unknown, the conclusions of Theorem 5 still hold if we use an underestimate $\hat{R} \leq R$.

7 Extensions to the Agnostic Setting

Most of our algorithms have two phases: first, we extract a partitioning of the unlabeled data into groups that are likely label-homogeneous, and second, we query the label of the largest groups. We can extend our results for these algorithms to the agnostic setting by querying multiple labels from each group and using the majority label.

Specifically, suppose that the data is generated according to a distribution P over $\mathcal{X} \times [L]$ and there exists a labeling function f^* such that $\Pr_{(x,y) \sim P}(f^*(x) \neq y) \leq \eta$ and our assumptions hold when the unlabeled data is drawn from the marginal $P_{\mathcal{X}}$ but the labels are assigned by f^*. That is, there is a function f^* satisfying our assumptions such that the true label disagrees with f^* with probability at most η. In this setting, the first phase of our algorithms, which deals with only unlabeled data, behaves exactly as in the realizable setting. The only difference is that we will need to query multiple labels from each group of data to ensure that the majority label is the label predicted by f^*. Suppose that the training data is $(x_1, y_1), \dots, (x_n, y_n)$ drawn from P (where the labels y_i are initially unobserved). For $n = \tilde{O}(1/\eta^2)$, we are guaranteed that $y_i \neq f^*(x_i)$ for at most $2\eta n$ points w.h.p. Moreover, if we only need to guess the label of large groups of samples, say those containing at least $8\eta n$ points, then we are guaranteed that within each group at least $1/4$ of the sample points will have labels that agree with f^*. Therefore, after querying $O(\log(1/\delta))$ labeled examples from each group, the majority label will agree with f^*. If we use these labels in the second phase of the algorithm, we would be guaranteed that the error of our algorithm would be at most ϵ had the labels been produced by f^*, and therefore the error under the distribution P is at most $\eta + \epsilon$. The full version of the paper contains agnostic versions of Theorems 1, 4, and 5.

Similarly, modifying Algorithm 2 to require that the each cluster in the pruning have a majority label that accounts for at least $3/4$ of the cluster's data can be used to extend the corresponding result to the agnostic setting.

8 Conclusion and Discussion

In this work we showed how to exploit the implicit geometric assumptions made by output code techniques under the well studied cases of one-vs-all and well separated codewords, and for a novel boundary features condition that captures the intuition that every binary learning task should be significant. We provide label-efficient learning algorithms for both the consistent and agnostic learning settings with guarantees when the data density has thick level sets or upper and lower bounds. In all cases, our algorithms show that the implicit assumptions of output code learning can be used to learn from very limited labeled data.

In this work we focused on linear output codes, which have been used in several practical works. For example Palatucci et al. (2009) use linear output codes to decode thoughts from fMRI data, Berger (1999) used them for text classification, and Crammer and Singer (2000) show that they perform well on MNIST and several UCI datasets. Many other works use non-linear output codes, and it is an interesting direction to extend our results to these cases.

The unlabeled sample complexity of our algorithms is exponential in the dimension because our algorithms require the samples to cover high-density regions. It is common for semi-supervised algorithms to require exponentially more unlabeled data than labeled, e.g. (Singh, Zhu, and Nowak 2008; Castelli and Cover 1995). Our results also show that the unlabeled sample complexity only scales exponentially with the intrinsic dimension, which may be significantly lower than the ambient dimension for real-world problems. An interesting direction for future work is to determine further conditions under which the unlabeled sample complexity can be drastically reduced.

Acknowledgments

This work was supported in part by NSF grants CCF-1422910, CCF-1535967, IIS-1618714, a Sloan Research Fellowship, a Microsoft Research Faculty Fellowship, and a Google Research Award.

References

Allwein, E.; Schapire, R.; and Singer, Y. 2000. Reducing multiclass to binary: A unifying approach for margin classifiers. In *Journal of Machine Learning Research*.

Balasubramanian, K.; Donmez, P.; and Lebanon, G. 2011a. Unsupervised supervised learning ii: Margin-based classification without labels. In *AISTATS*, 137–145.

Balasubramanian, K.; Donmez, P.; and Lebanon, G. 2011b. Unsupervised supervised learning ii: Margin-based classification without labels. In *Journal of Machine Learning Research*, volume 12, 3119–3145.

Balcan, M.-F., and Blum, A. 2010. A discriminative model for semi-supervised learning. In *Journal of the ACM*.

Balcan, M.-F., and Urner, R. 2015. Active learning. In *Survey in the Encyclopedia of Algorithms*.

Balcan, M.-F.; Beygelzimer, A.; and Lanford, J. 2006. Agnostic active learing. In *ICML*.

Balcan, M.-F.; Blum, A.; and Mansour, Y. 2013. Exploiting ontology structures and unlabeled data for learning. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, 1112–1120.

Balcan, M.-F.; Blum, A.; and Yang, K. 2004. Co-training and expansion: Towards bridging theory and practice. In *NIPS*.

Balcan, M.-F.; Dick, T.; and Mansour, Y. 2016. Label efficient learning by exploiting multi-class output codes. *CoRR* abs/1511.03225.

Berger, A. 1999. Error-correcting output coding for text classification. In *IJCAI Workshop on machine learning for information filtering*.

Beygelzimer, A.; Langford, J.; and Ravikumar, P. 2009. Solving multiclass learning problems via error-correcting output codes. *ALT*.

Blum, A., and Mitchell, T. 1998. Combining labeled and unlabeled data with co-training. In *COLT*.

Castelli, V., and Cover, T. 1995. On the exponential value of labeled samples. In *Pattern Recognition Letters*.

Chapelle, O.; Schlkopf, B.; and Zien, A. 2010. *Semi-Supervised Learning*. The MIT Press, 1st edition.

Crammer, K., and Singer, Y. 2000. Improved output coding for classification using continuous relaxation. In *NIPS*.

Daniely, A.; Schapira, M.; and Shahaf, G. 2012. Multiclass learning approaches: A theoretical comparison with implications. In *NIPS*.

Dasgupta, S., and Sinha, K. 2013. Randomized partition trees for exact nearest neighbor search. In *COLT*.

Dasgupta, S. 2011. Two faces of active learning. In *Theoretical Computer Science*.

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

Donmez, P.; Lebanon, G.; and Balasubramanian, K. 2010. Unsupervised supervised learning i: Estimating classification and regression errors without labels. In *Journal of Machine Learning Research*, volume 11, 1323–1351.

Hanneke, S. 2014. Theory of active learning. *Foundations and Trends in Machine Learning* 7(2–3).

Langford, J., and Beygelzimer, A. 2005. Sensitive error correcting output codes. *COLT*.

Mitchell, T.; Cohen, W.; Hruschka, E.; Talukdar, P.; Betteridge, J.; Carlson, A.; Dalvi, B.; Gardner, M.; Kisiel, B.; Krishnamurthy, J.; Lao, N.; Mazaitis, K.; Mohamed, T.; Nakashole, N.; Platanios, E.; Ritter, A.; Samadi, M.; Settles, B.; Wang, R.; Wijaya, D.; Gupta, A.; Chen, X.; Saparov, A.; Greaves, M.; and Welling, J. 2015. Never-ending learning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI-15)*.

Mohri, M.; Rostamizadeh, A.; and Talwalkar, A. 2012. *Foundations of Machine Learning*. MIT press.

Palatucci, M.; Pomerleau, D.; Hinton, G.; and Mitchell, T. 2009. Zero-shot learning with semantic output codes. In *NIPS*.

Singh, A.; Zhu, X.; and Nowak, R. 2008. Unlabeled data: Now it helps, now it doesn't. In *NIPS*.

Steinhardt, J., and Liang, P. 2016. Unsupervised risk estimation with only structural assumptions. (Preprint).

Steinwart, I. 2015. Fully adaptive density based clustering. In *Annals of Statistics*, volume 43, 2132–2167.

Thrun, S., and Mitchell, T. 1995a. Learning one more thing. In *Proc. 14th International Joint Conference on Artificial Intelligence (IJCAI)*, 1217–1225.

Thrun, S., and Mitchell, T. M. 1995b. Lifelong robot learning. *Robotics and Autonomous Systems* 15(1-2):25–46.

Thrun, S. 1996. *Explanation-Based Neural Network Learning: A Lifelong Learning Approach*. Boston, MA: Kluwer Academic Publishers.