Active Boosted Learning (ActBoost)

Kirill Trapeznikov
Boston University

Venkatesh Saligrama
Boston University

David Castañón
Boston University

Abstract

Active learning deals with the problem of selecting a small subset of examples to label, from a pool of unlabeled data, for training a good classifier. We develop an active learning algorithm in the boosting framework. In contrast to much of the recent efforts, which has focused on selecting the most ambiguous unlabeled example to label based on the current learned classifier, our algorithm selects examples to maximally reduce the volume of the version space of feasible boosted classifiers. We show that under suitable sparsity assumptions, this strategy achieves the generalization error performance of a boosted classifier trained on the entire data set while only selecting logarithmically many unlabeled samples to label. We also establish a partial negative result, in that with out imposing structural assumptions it is difficult to guarantee generalization error performance. We explicitly characterize our convergence rate in terms of the sign pattern differences produced by the weak learners on the unlabeled data. We also present a convex relaxation to account for the non-convex sparse structure and show that the computational complexity of the resulting algorithm scales polynomially in the number of weak learners. We test ActBoost on several datasets to illustrate its performance and demonstrate its robustness to initialization.

1 Introduction

An active learner [Cohn et al., 1994] selects which examples to label from a pool of unlabeled data for training a good classifier. Active learning is based on the premise that while unlabeled data is abundant, labeling it is expensive. This paper develops a novel active learner based on boosting. Boosting is a method for combining decisions of weak classifiers to form a strong classifier. A strong classifier is parameterized by a probability weight vector on the weak classifiers. The significance of boosted classifiers is that if the weak learning assumption holds, a probability weight vector and the associated strong classifier can be constructed with training error essentially equal to zero [Freund & Schapire, 1996].

This feature of boosted classifiers motivates considering an active boosted learner based on the version space approach. In our context, the version space is the set of all probability vectors that correctly classify the current labeled training set. Our active boosted learning (ActBoost) algorithm, at each time, selects examples to approximately bisect the version space. We establish that the number of labeled examples n necessary to reduce the volume of the version space to a fraction ϵ of the initial volume scales as $n = O(\log \frac{1}{\epsilon})$. ActBoost randomly samples from the version space and chooses an example with maximum disagreement among the sampled boosted classifiers. We utilize the Hit and Run algorithm from [Lovász & Vempala, 2004] to uniformly sample from a convex body. ActBoost has polynomial computational complexity in the number of weak learners. Hit and Run has been used for SVM active learning by [Gilad-Bachrach et al., 2005].

Nevertheless, we show that in the context of boosting, if all the weak hypotheses are allowed to contribute to the final ensemble, reduction of version space does not guarantee improvement in generalization performance. This motivates imposition of sparsity. Sparsity in the number of boosted weak learners has been employed in the literature [Taylor et al., 2010] and has been shown to improve generalization. We show that under this assumption, our strategy achieves the generalization error performance of a boosted classifier trained on the entire data set while only selecting logarithmically many unlabeled samples to label.

Active learning has been extensively studied by several researchers (see [Settles, 2010]). In contrast to

Appearing in Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS) 2011, Fort Lauderdale, FL, USA. Volume 15 of JMLR: W&CP 15. Copyright 2011 by the authors.

boosted classifiers, version spaces can turn out to be empty sets after a few iterations for important classifier classes such as SVMs. This has motivated researchers to consider strategies for finding the most ambiguous/uncertain examples based on the current learned classifier (see [Campbell et al., 2000, Tong & Koller, 2001, Nguyen & Smeulders, 2004, Dasgupta & Hsu, 2008, Guo & Greiner, 2007]).

Active learning in the boosting framework with the most ambiguous example perspective has also been studied before [Abe & Mamitsuka, 1998, Tur et al., 2003]. [Abe & Mamitsuka, 1998] describe Query-by-Boosting (QBB) method using the Adaboost algorithm of [Freund & Schapire, 1996]. QBB labels an example that is most ambiguous among all examples for the Adaboost classifier trained on the current labeled set. Unfortunately, no performance guarantees for QBB can be provided since it suffers from initialization bias as we will describe later. In contrast we select examples that most closely bisects the current space of feasible boosted classifiers. We require no initialization because our algorithm does not concentrate on the current boundary but explores the entire example space. We compare the performance of our algorithm on several datasets to QBB algorithm from [Abe & Mamitsuka, 1998] and to a random labeling strategy. On all datasets, our algorithm performs better or on par with QBB. However when initialization bias is introduced into the experiments, QBB performance deteriorates while our algorithm remains robust.

The idea of reducing the version space in active learning has been considered by several researchers ([Seung et al., 1992, Gilad-Bachrach et al., 2005, Abe & Mamitsuka, 1998, Freund et al., 1997, Tong & Koller, 2001, Nowak, 2009]) for the so called realizable case. The Query By Committee (QBC) of [Freund et al., 1997] is a version space approach described usually in a streaming scenario. The QBC selects an unlabeled instance to label if two random classifiers chosen from the version space disagree. Freund et. al. describe a PAC framework and show that if the set of classifiers have a finite VC dimension, the classifiers and the unlabeled data points are chosen from a known prior distribution, and the so called information gain condition is satisfied, the number of examples labeled is small. Our algorithm is similar in spirit to QBC but with important differences. We do not employ a PAC framework or the streaming scenario. We do not assume the information gain conditions or bounds on VC dimension. Instead we draw upon margin based generalization bounds for boosted classifier. Our analysis draws upon the concept of Generalized Binary Search of Nowak [Nowak, 2009]. We extend [Nowak, 2009] to the case of continuous classifier spaces, more precisely the space of probability vectors on weak learners. We explicitly relate our convergence rate to a function of the hamming distances of the weak classifiers.

The paper is organized in the following manner. In Section, 2 we define our algorithm. We perform theoretical analysis in Section 3. We present our numerical experiments in Section 4.

2 Active Boosted Learning

We denote by $\mathcal{X} = \{x_i\}_{i=1}^{B}$, the pool of unlabeled data. We let $\mathcal{H} = \{h_j(\cdot)\}_{j=1}^{N}$ denote a finite set of weak classifiers[1]. Each weak classifier is a binary valued function $h_j : \mathcal{X} \to \{+1, -1\}$. Associated with each example $x_i \in \mathcal{X}$ is a binary label $y_i \in \{-1, 1\}$, which is revealed by querying an Oracle. We introduce the simplex of all positive weight vectors Q:

$$Q = \left\{ (q_1, q_2, \ldots, q_N) \mid \sum_{j=1}^{N} q_j = 1,\, q_j \geq 0,\, j = 1, \ldots, N \right\} \tag{1}$$

The space of classifiers under consideration is the set of all weighted combinations of weak classifiers, namely,

$$q(x) := \text{sgn}(\sum_{j=1}^{N} h_j(x) q_j) = \text{sgn}(\mathbf{h}(x)^T q)$$

where, $\mathbf{h}(x) = [h_1(x)\ h_2(x)\ ..\ h_N(x)]^T$ and $q = [q_1, q_2, \ldots, q_N]$. We refer to this set of classifiers as boosted classifiers.

Our goal is to select a small subset of examples $x_i \in \mathcal{X}$ to label so that boosted training on this labeled set leads to a strong classifier, namely that $\sum \mathbb{1}_{[q(x_i) \neq y_i]}$ on the unlabeled data is small. Let L^t be the set of labeled training examples at iteration t. Our version space is the set of all classifiers parameterized by the vector q that correctly classify the labeled training set at an iteration t:

$$Q^t = \left\{ q \in Q \mid y_i \sum_{j=1}^{N} h_j(x_i) q_j \geq 0,\ \forall\ i \in L^t \right\}$$

Note that under the weak learning assumption [Freund & Schapire, 1996], it is well known that there exists a set of weights, q_j, such that the above set is not empty.

At every iteration, a new unlabeled example is labeled and added to L^t, and the version space Q^t decreases such that: $Q = Q^0 \supset Q^1\ \ldots\ Q^t \supset Q^{t+1}$. We want

[1]ActBoost can be extended to a suitably parameterized continuous space of weak classifiers but we consider a finite set for technical simplicity

to maximally reduce the version space at every iteration. To do so we draw upon recent ideas from generalized binary search [Nowak, 2009]. Let U^t be a set of unlabeled examples at iteration t. The goal is to to pick an $x \in U^t$, so that one half of the volume of Q^t labels this point $+1$ and the other -1. Once the label y is revealed half of Q^t will be eliminated. If the space Q is discrete then the query strategy is: $x^t = argmin_{x \in U^t} | \sum_{q \in Q^t} q(x)|$. In our case, Q being a continuous space of weight vectors our query strategy is to pick:

$$\begin{aligned} x^t &= \arg\min_{x \in U^t} \left| \int_{q \in Q^t} q(x) dq \right| && (2) \\ &:= \arg\min_{x \in U^t} |Vol(Q^t_+) - Vol(Q^t_-)| && (3) \end{aligned}$$

where, dq represents the Lebesgue measure on the space of normalized weight vectors; $Vol(Q^t_+) = \int_{q \in Q^t} \mathbb{1}_{[q(x)=1]} dq$ and $Vol(Q^t_-) = \int_{q \in Q^t} \mathbb{1}_{[q(x)=-1]} dq$. The integral is well defined since Q^t is a bounded polyhedron and the indicator functions described here are Lebesgue measurable.

Randomly Sampling a Polyhedron: The expression in (2) is hard to evaluate so we approximate it by uniformly sampling from Q^t. If we draw D random samples $q^1, q^2, .., q^D$ then approximation is the following:

$$x^t = \arg\min_{x \in U^t} |\sum_{d=1}^{D} q^d(x)| \qquad (4)$$

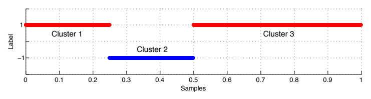
This sampling from the version space is related to QBC algorithm of Freund et. al. [Freund et al., 1997, Seung et al., 1992] with one principle difference, namely, our goal is to find the best example to approximate the volume(s) of the version space[2], while [Freund et al., 1997] employs it to determine whether or not an instance is to be labeled.

Since our version space is a bounded polyhedron we can uniformly sample by employing the Hit and Run algorithm [Lovász & Vempala, 2004]. The "Hit" step generates a random direction and draws a line in that direction through the interior point. The "Run" step generates a new interior point by uniformly sampling along the interval defined by the line in the "Hit" step and the boundary of the polyhedron. As two steps are repeated, the generated interior points converge to a uniform sample from the polyhedron. For a given labeled set, we can employ a phase I optimization approach [Boyd & Vandenberghe, 2004] to find a feasible point and a cutting plane or barrier method [Boyd & Vandenberghe, 2004] to find a point near the center of the polyhedron. Recently, Kannan et. al. [Kannan & Narayanan, 2009] have developed techniques which produces a uniformly random sample with complexity scaling as $O(|L_t|N^2)$.

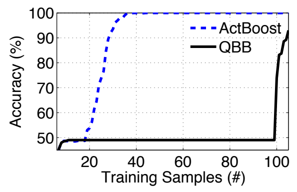
[2]Note that the random variable $q^d(x)$ is Bernoulli for each x and one can obtain a precise approximate characterization for sufficiently large D by using a combination of Chernoff and Union bounds.

Algorithm 1 Active Boosted Learning (ActBoost)

```
INPUT: $H$ {problem matrix}, $T$ {number of iterations}
$L^0 \leftarrow \emptyset$, $U^0 \leftarrow \mathcal{X}$, $Q^0 \leftarrow \{q | \mathbf{1}^T q = 1,\ q \geq 0\}$, $t \leftarrow 0$
while $t \leq T$ do
  $q^0 \leftarrow init(Q^t)$ {compute initial classifier in version space: $q^0 \in Q^t$}
  for $d = 1$ to $D$ do
    $q^d \leftarrow sample(q^0, Q^t)$ {draw a uniform random samples from version space}
  end for
  $x^t \leftarrow \arg\min_{x \in U^t} | \sum_{d=1}^{D} q^d(x)|$ {find the closest bisecting example}
  $y^t \leftarrow label(x^t)$, $Q^{t+1} \leftarrow Q^t \cap \{q | y^t \mathbf{h}(x^t)^T q \geq 0\}$ {label and update the version space}
  $L^{t+1} \leftarrow L^t \cup \{x^t, y^t\}$, $U^{t+1} \leftarrow U^t \setminus x^t$, $t \leftarrow t + 1$
end while
OUTPUT: $\{x_i, y_i\} \in L^T$ {labeled set of examples}
```



(a) 1D Linear Clusters Dataset. Red: $+1$, Blue: -1 class



(b) Accuracy on $1D$ linear clusters

Figure 1: Initialization bias is tested on a $1D$ linear set (in 1(a)) consisting of three clusters ActBoost and QBB are initialized with examples drawn only from the first two clusters. Both ActBoost and QBB are trained using Adaboost on their respective labeled examples. ActBoost is robust to initialization bias while QBB does not find the third cluster until all the first two clusters are labeled.

2.1 ActBoost vs. Query-by-Boosting(QBB)

As we described earlier recent effort in active learning has focused on finding examples that are most ambiguous for the currently trained classifier. QBB is such an algorithm based on boosting. In contrast the ActBoost algorithm is based on finding an example that approximately bisects the version space. We use $1D$ example with three clusters in Figure 1(b) to point out an important attribute of ActBoost, namely, that it does not suffer from initialization bias.

QBB in [Abe & Mamitsuka, 1998] labels an example with the smallest margin with respect to the current classifier: $x_{QBB} = \arg\min_{x_i \in U^t} |\sum_{j=1}^{N} h_j(x_i) q_j^t|$ (where q^t are the weights of the adaboost classifier trained on the labeled data).

Initial training set L^0 is forced to sample from only the first two clusters. QBB is a margin based algorithm and only queries examples in the vicinity of the current boundary estimate. Since the initialization is skewed to omit the third cluster, the second boundary is not detected until all the samples in the first two clusters are labeled (Figure 1(b)). ActBoost is not affected and starts to query from the third cluster much sooner. Margin based classifiers fail to explore the space far from the boundary. They heavily rely on the current classifier estimate and become sensitive to initialization.

3 Theory

All proofs of theorem and lemma statements appear in the supplementary section. Our main result is based on establishing that ActBoost achieves exponential reduction of version space volume. Nevertheless, Theorem 1 shows that if the set of weak learners is negation complete, there is no labeling strategy that can guarantee generalization performance. Consequently, structural assumptions on the target boosted classifier must be imposed. Under sparsity assumptions, which is practically well-motivated, we show that version space reduction results in generalization.

At stage t suppose L^t is the set of labeled examples and the version space $Q^t = \{q \in Q | \mathbf{h}(x)^T q \geq 0, x \in L^t\}$. We suppose that our weak hypothesis set $\mathcal{H} = \{h_j\}_{j=1}^{N}$ is negation complete, i.e., corresponding to each weak learner the hypothesis set admits its complement.

Theorem 1. *Suppose the weak hypothesis set is negation complete and the weak learning hypothesis is satisfied. Then for any unlabeled example $x_k \notin L^t$ there exists $q_+, q_- \in Q^t$ such that both $\mathbf{h}(x_k)^T q_+ > 0$ and $\mathbf{h}(x_k)^T q_- < 0$ are satisfied. Consequently, regardless of the labeling strategy and at any stage t it follows that,* $\max_{q,q' \in Q^t} |q(x) - q'(x)| = 1$.

Thus the version space always contains two classifiers that disagree on any unlabeled example. This implies that one cannot obtain useful information on the unlabeled samples based on the labeled samples. This situation calls for imposing additional constraints on the hypothesis set. Generalization ability of boosting has been shown to depend on two aspects: the margin [Schapire et al., 1997] and the sparsity in the hypotheses weight vector q [Taylor et al., 2010] . A boosted subset of weak hypothesis achieves better generalization error than an ensemble of a complete set. More so, [Koltchinskii & Panchenko, 2005] showed that generalization can be improved if the weights for a subset of weak hypothesis decay to zero at an exponential rate. And experimentally, majority of boosting algorithms when trained on common datasets also favor sparse ensembles. Consequently, sparsity and margin assumptions on the boosted classifier trained on the entire data set appears to arise naturally and we will impose such conditions to relate volume reduction rate to error rates in Section 3.2. In the following section we quantify volume reduction rates when the initial version space is the entire simplex. We then extend these results to the sparse setting and relate version space volume to error rates.

3.1 Volume Convergence Rate

In this section we prove that the ActBoost algorithm has an exponential rate of convergence, namely, that to reduce the version space to an ϵ fraction of its original volume requires labeling $O\left(\log \frac{1}{\epsilon}\right)$ unlabeled points. To establish this fact we introduce the concept of coherence and neighborliness for continuous spaces of weight vectors based on concepts developed by [Nowak, 2009] for the generalized binary search problem.

For convenience we introduce the matrix, $\mathbf{H}$ to denote,

$$\mathbf{H} = [h_{ij}],\ h_{ij} = h_j(x_i);\ i = 1, 2, \ldots, B,\ j = 1, 2, \ldots, N$$

In other words the jth column of the matrix $\mathbf{H}$ denotes the sign-pattern for the jth weak learner on all the data points, while the ith row denotes the sign pattern of all the weak learners on data point x_i. We say that x_i and x_j are K-neighborly if the difference in the sign patterns across all classifiers is smaller than K:

$$d_h(x_i, x_j) = \sum_{k=1}^{N} \mathbb{1}_{[h_k(x_i) \neq h_k(x_j)]} \leq K$$

The significance of K-neighborliness of two data points is described in the following important lemma. The lemma provides a connection between the discrete world of Hamming distances to volumes on continuous spaces.

Lemma 1. *If x and x' are K-neighbors then, for any $Q' \subset Q$, where Q is as in Equation 1,*

$$\frac{1}{2} \int_{Q'} \mathbb{1}_{[q(x) \neq q(x')]} dq \leq \frac{2K+1}{2N} Vol(Q) \tag{5}$$

where dq is the Lebesgue measure on Q.

K-Connected Graph: We introduce a graph based on the neighborliness property introduced above for later use. We form a graph G with nodes, $x \in \mathcal{X}$. Two nodes x_i, x_j, have an edge if they are K-neighborly. A graph is said to be K-connected if the K-NNG[3] graph so formed is connected. Note that K-connectedness is a property of the matrix $\mathbf{H}$.

Example: We note that 1D stumps are 1-connected. Figure 2 depicts stump classifiers which are our weak classifiers. A 1D stump is defined as follows: $h_j(x_i) = sgn(x_i - t_j)$, where t_j is a threshold.

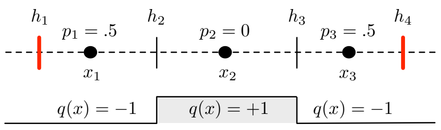


Figure 2: Stump Classifiers

Finally, we need the notion of coherence. Let $Q' \subset Q$, and denote by

$$\rho^*(\mathcal{X}, Q') = \min_{\sum_i p_i = 1,\, p_i \geq 0} \sup_{q \in Q'} \left| \sum_{x_i \in \mathcal{X}} q(x_i) p_i \right| \quad (6)$$

where, p_i denotes the weight of the ith data point. Note that by construction $0 \leq \rho^*(\mathcal{X}, Q') \leq 1$ and if $Q_1 \subset Q_2$ then $\rho^*(\mathcal{X}, Q_1) \leq \rho^*(\mathcal{X}, Q_2)$.

The significance of the above definition becomes clear in the following lemma.

Lemma 2. *Suppose the graph induced by the matrix* $\mathbf{H}$ *is K-connected and* $\rho^*(\mathcal{X}, Q') < 1$ *for some* $Q' \subset Q$. *Then for any* ρ *s.t.* $\rho^* \leq \rho < 1$, *one the following statements must hold:*

(1) $| \int_{Q'} q(x) dq \,| \leq \rho \, Vol(Q')$

(2) $Vol(Q') < \frac{(2K+1)}{N\rho} Vol(Q)$

To build intuition into why the algorithm described in Section 2 results in reducing the size of the feasible set note that if at iteration t we have $| \int_{Q^t} q(x) dq| = |Vol(Q^t_+) - Vol(Q^t_-)| \leq \rho_t Vol(Q^t)$ for some $\rho_t < 1$, where Q^t_+ is the region where $q(x) = 1$ and Q^t_- for $q(x) = -1$. Without loss of generality suppose $+1$ is the true label for x. So Q^t_- will be eliminated if x is labeled. Then, $Q^{t+1} = Q^t_+$. Substituting $Vol(Q^t_-) = Vol(Q^t) - Vol(Q^t_+)$ we obtain:

$$|Vol(Q^{t+1}) - (Vol(Q^t) - Vol(Q^t_+))| \leq \rho_t Vol(Q^t) \quad (7)$$

$$\implies Vol(Q^{t+1}) \leq \frac{(1+\rho_t)}{2} Vol(Q^t) \quad (8)$$

However, it turns out that the Lemma 2 is still insufficient and we need a regularity (smoothness) condition.

[3]K-NNG: Nearest Neighbor Graph where only K-neighborly vertices are connected

Regularity Condition: Suppose $\tilde{Q} \subset Q$ such that $Vol(\tilde{Q}) \geq (1-\eta) Vol(Q)$ for some fixed constant $\eta > 0$, then for any two data points, x and x' there is an $\alpha > 0$ such that,

$$\int_{\tilde{Q}} \mathbb{1}_{[q(x) \neq q(x')]} dq \geq \alpha \int_{Q} \mathbb{1}_{[q(x) \neq q(x')]} dq \quad (9)$$

Basically the regularity condition states that the disagreement volume, $V_d(Q) = \int_Q \mathbb{1}_{[q(x) \neq q(x')]} dq$ on the original simplex Q (see Eq. 1) cannot change arbitrarily if a small subset of Q is removed. In other words, if $\Delta Q = Q \setminus \tilde{Q}$ has small volume then the disagreement volume, $V_d(\Delta Q)$ has to be small as well. We are now ready to state our main theorem.

Theorem 2. *Consider the algorithm of Section 2 where an unlabeled data point* $x \in U^t$ *is picked at stage* t *as the minimizer to Equation 2. Further assume that the graph induced by the matrix* $\mathbf{H}$ *is K-connected and the assumption given by Equation 9 is satisfied. Then to reduce the volume of the version space to a fraction* ϵ *of its original volume requires* $n = \frac{\log \epsilon}{\log \lambda}$ *iterations where* $\lambda = \max\{\frac{1+\rho^*}{2}, \frac{1}{2}(1 + (1-\alpha)\frac{2K+1}{N})\}$.

Dealing with Constant Offsets: Constant offsets lead to $\rho^* = 1$, which implies no reduction in version space. To see this consider the $1D$ case in Figure 2 with stump classifiers. Outer stumps h_1, h_4 (bold red) are offsets and are problematic because they appear as columns of either all 1's or all -1's in the problem matrix $\mathbf{H}$. Note that for this situation $\rho^* = 1$ since all the weight can be assigned to q_1 or q_4. However, if the stumps h_1 and h_4 are removed it can be easily seen that $\rho^* = 0$. This example generalizes to stumps in multiple dimensions. This scenario makes sense: if the target classifier is the outer stump then in our search process no query helps in significantly reducing the version space, and every example has to be labeled. However, completely removing the outer stumps may degrade the classification ability of the hypothesis set. Instead, we assume that the good boosted classifier does not concentrate all of its weight on the problematic stumps. Let I_p be the problematic set, and the augmented version space $Q' = Q \cap \{q | \sum_{j \in I_p} q_j \leq \eta\}$ for some fraction η. Maximization of q in (6) will be over the augmented version space Q', and q will no longer be able to put all the weight on the outer stumps resulting in ρ^* strictly less than zero. This constraint is equivalent to an assumption that the target classifier will not be trivial and will not exclusively consist of outer stumps.

3.2 Error Convergence

In this section we will discuss how reduction in version space Q is related to generalization error. Based on our earlier arguments we impose sparsity and margin constraints on the target boosted classifier, namely,

$$\exists q \in S = \{q \in Q \mid \|q\|_0 \leq p\} \text{ such that } y_i q^T h(x_i) \geq \theta$$

for all $x_i \in \mathcal{X}$ and for some $\theta > 0$ and where $\|\cdot\|_0$ is the so called ℓ_0 norm and characterizes the number of non-zero elements in the vector q. Under the margin constraints [Schapire et al., 1997] has shown that the generalization error scales as $O(\frac{\log|\mathcal{X}|\log p}{\theta^2|\mathcal{X}|})^{\frac{1}{2}}$. Consequently, our problem boils down to labeling sufficiently many points such that every element in our version space satisfies the margin constraint.

Our goal is to reduce the sparse version space such that $S \supset S^1 \supset \ldots \supset S^t$. Note that the set S is made of $\binom{N}{p}$ p-sparse disjoint simplices. For notational convenience we denote $\{s_1, s_2, \ldots, s_{\binom{N}{p}}\} = S \subset Q$

Our modified query strategy is to find an example to bisect the sparse version space:

$$x^* = \arg\min_{x \in U^t} \left| \sum_{r=1}^{\binom{N}{p}} \int_{q \in s_r} q(x)\ d\mathcal{P}(q) \right| \tag{10}$$

Note that our modified algorithm accounts for sparsity of the target boosted classifier but does not assume knowledge of θ. Analogous to the setup for volume reduction on the entire simplex in Section 3.1 we define ρ^* as

$$\rho^*(\mathcal{X}, S') = \min_{\sum_i p_i = 1,\, p_i \geq 0} \sup_{q \in S} \left| \sum_{x_i \in \mathcal{X}} q(x_i) p_i \right|,\ for\ S' \subset S.$$

and $\lambda = \max\{\frac{1+\rho^*}{2}, \frac{1}{2}(1 + (1-\alpha)\frac{2K+1}{N})\}$. Define,

$$f(\theta, p) = \inf_{q^* \in S} Vol(\{q \in S \mid \|q - q^*\|_1 \leq \theta/2\})$$

where volume is taken with respect to the lebesgue measure on the p sparse subspace.

Theorem 3. *Consider the strategy where an unlabeled data point $x \in U^t$ is picked at stage t as the minimizer to Equation (10) and suppose the regularity conditions of Theorem 2 are satisfied for the sparse set S. Let the number of stages n and hence the number of labeled samples satisfy*

$$n \geq \frac{\log\binom{N}{p} + \log\frac{1}{f(\theta,p)}}{\log\frac{1}{\lambda}}.$$

Then for all $q \in S^n$, it follows that

$$Prob(q(x) \neq y) \leq O\left(\frac{\log|\mathcal{X}|\log p}{\theta^2|\mathcal{X}|} + \frac{\log(1/\delta)}{|\mathcal{X}|}\right)^{\frac{1}{2}}$$

with probability $1-\delta$ for a $\delta > 0$ and where $|\mathcal{X}|$ is the size of the unlabeled data pool.

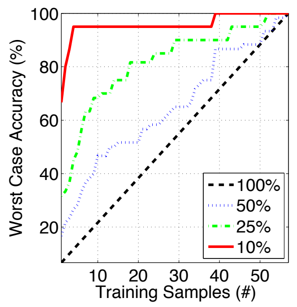


Figure 3: Accuracy of the worst case classifier vs. # labeled examples. For each curve, only the specified fraction of hypotheses is allowed in the version space. The true sparsity of the target classifier is 3%.

Figure 3 illustrates implication of our modified algorithm on Fig. 1(a) for increasing knowledge of sparsity.

Convex Surrogate: To reduce the sparse subset instead of the full version space is combinatorially hard because the integral in Equation (10) has to be enumerated for every p-sparse segment and p is also unknown. If we convexify this problem, then ActBoost amounts to reducing the convex hull of the sparse subspace at every iteration. To see this let $S^t = \{q \in Q^t \mid \|q_0\| = p\}$. Instead of selecting an example to bisect a non-convex subspace S^t, suppose we reduce the convex hull of S^t: $\mathcal{C}(S^t)$. Note that $Q = \mathcal{C}(S)$ and for a set of labeled examples: $Q^t \supset \mathcal{C}(S^t)$. At $t+1$, if we select an example to reduce the version space Q^t, it will also reduce $\mathcal{C}(S^{t+1})$:

$$\min_{x \in U^t} Vol(\mathcal{C}(S^{t+1})) \leq \min_{x \in U^t} Vol(Q^{t+1}) \tag{11}$$

While we can reduce the convex hull of S^t, we cannot guarantee that the subspace S^t is also reduced by the same amount. However, our simulations demonstrate that reducing the full version space results in generalization.

3.3 LP for Bounding Coherence

In Theorem 1 we saw that coherence (see Eq. 6) controls the convergence. In this section we present a linear programming solution for computing the coherence. First, we have the following result.

Lemma 3. *Let $\rho^*(Q', \mathcal{X})$ be as in (6) and suppose the set of weak classifiers is balanced, namely, for each $h_j(\cdot) \in \mathcal{H}$ there is a corresponding element $h_k(\cdot) = -h_j(\cdot) \in \mathcal{H}$. Then,*

$$\textbf{(1)}\ \rho^*(Q', \mathcal{X}) < 1 \implies \exists \lambda \geq 0,\ \lambda \neq 0,\ \lambda^T H = 0$$
$$\textbf{(2)}\ \exists q \in Q',\ q \geq 0,\ Hq \geq 0 \implies \rho^*(Q', \mathcal{X}) = 1$$

Note that an example of balanced set of classifiers are stumps described in the previous section. While the above lemma characterizes when $\rho^* < 1$, it does not directly help in finding a bound. To this end we consider the following LP and its corresponding dual.

$$\max_{v,q} v,\ s.t.\ \mathbf{H}q \geq v,\ \mathbf{1}^T q = 1,\ q \geq 0 \Leftrightarrow$$
$$\min_{\pi,\lambda} \pi,\ s.t.\ \lambda^T \mathbf{H} \leq \pi,\ \mathbf{1}^T \lambda = 1,\ \lambda \geq 0$$

If the value of the primal or dual is less than zero then $\rho^*(\mathcal{X}, Q')$ is less than one. The dual variable λ is the probability distribution on the examples. Notice that the constraint $\mathbf{1}^T\lambda = 1$ imposes sparsity. Only the examples with a corresponding non-zero λ_i are active constraints in the primal. The examples with zero λ_i can be removed and the solution of the problem will not change. In the expression (6), we can put equal weights p_i only on the examples with non-zero λ_i. Suppose the cardinality of the support of λ is l, then a bound on ρ^* follows:

$$\rho^* \leq \max_{q \in Q} |\sum_{i|\lambda_i > 0} q(x_i)\frac{1}{l}| \implies \rho^* \leq |\frac{l-1}{l} - \frac{1}{l}| = 1 - \frac{2}{l}$$

where we have used the fact that $\rho^* < 1$. This implies that $q(x)$ cannot have the same sign for all $x \in \mathcal{X}$. Note that as the sparsity of λ increases, ρ^* decreases.

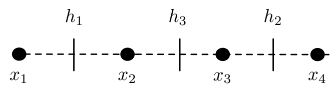
3.4 Undersampling vs. Oversampling

The convergence rate in Theorem 1 depends on how large a K is necessary to ensure K-connectedness. To build intuition, consider a 1D stump example with three stumps h_1, h_2, h_3 and four training samples x_1, x_2, x_3, x_4. Each row $\mathbf{h}(x_i)$ in the matrix in Figure 4(a) captures how each example is classified. Note that $K = 1$ is sufficient to ensure connectedness.

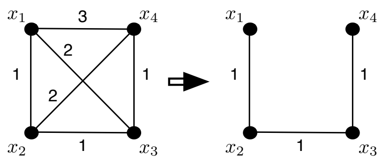
Suppose we oversample our weak hypotheses by adding three more stumps in bold red (Figure 4(b)). Now performing the same graph construction and reduction, then we need $K = 3$ to ensure connectedness. Since the value K increases this directly leads to a reduction of aggregate convergence rate λ. This implies that if we demand too much resolution in our target classifier and do not have an appropriately dense training set then the number of iterations increases exponentially with the disagreement factor K. Notice that the opposite case of oversampled training set does not hurt as it increases the set of possible queries. Note that increase in K here is a direct outcome of redundant weak learners. Consequently, these weak learners can be removed in simple cases but for more complicated cases this reduction may be non-trivial.

4 Experiments

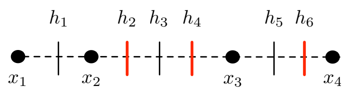
Set Up: We compare three algorithms random query strategy (RANDOM), QBB from [Abe & Mamitsuka, 1998] and ActBoost. RANDOM uniformly samples a random x^*_{random} from the unlabeled pool U^t at time t. QBB relies on the Adaboost solution $q^t_a = adaboost(L^t)$, computed on the current labeled set L^t: $x^*_{QBB} = \arg\min_{x \in U^t} |\mathbf{h}(x)^T q^t_a|$. All three algorithms are then trained using Adaboost on their respective labeled sets. Note ActBoost operates on the full version space in the simulation without knowledge of the sparsity.



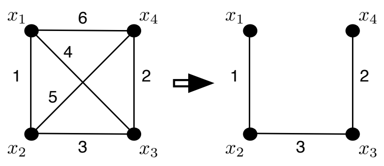
	h_1	h_2	h_3
x_1	-	-	-
x_2	+	-	-
x_3	+	+	-
x_4	+	+	+



(a) Undersampled



	h_1	h_2	h_3	h_4	h_5	h_6
x_1	-	-	-	-	-	-
x_2	+	-	-	-	-	-
x_3	+	+	+	+	-	-
x_4	+	+	+	+	+	+



(b) Oversampled

Figure 4: The parameter K required for maintaining graph connectedness in the case of undersampled and oversampled weak learners. For each case shown: $1D$ dataset with stumps, the problem matrix $\mathbf{H}$ and the neighborly graph reduction. Oversampling weak learners results in increase in K

The ActBoost algorithm has several parameters: number of samples D drawn by the Hit and Run algorithm, the number of iterations to generate one sample for the Hit and Run algorithm, initial size of the labeled set L^0 (necessary for comparison with QBB). Figure 5(a) demonstrates performance vs D. As expected, more samples result in better performance as the query comes closer to bisecting the version space. Changing the number of iterations for the Hit and Run algorithm does not have much affect on performance confirming that the sampling has a quick mixing time.

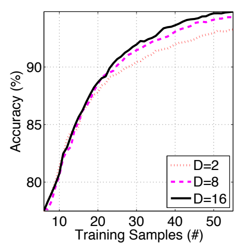
For each simulation an unlabeled training pool $\mathcal{X}$ of 200 is sampled uniformly from a dataset. Each simulation is averaged over 100 trials. For all simulations, the number of sampled classifiers is fixed at $D = 8$.[4] We chose to use Adaboost to train a classifier on the labeled set at every iteration. We then compared the performance of QBB, ActBoost and Random. Observe that the goal of active learning is to achieve performance of an entire training set while labeling and learning on only a fraction of examples. Following this philosophy, at each iteration, we compute the error against the entire training pool $\mathcal{X}$. Similar performance evaluation on 'query data' for QBB is employed in [Abe & Mamitsuka, 1998]. Since the labeled set of points is relatively small compared with the amount of unlabeled data, this also serves as a characterization of the generalization performance.

We use stumps for weak hypotheses as defined previously. For a given dataset of size B in $\mathbb{R}^{D_x}$, there are $2(B+1)D_x$ possible weak hypotheses. However, if the data examples are categorical or integers then the number of stumps can be significantly reduced by eliminating redundancies. ActBoost is tested on several datasets from the UC Irvine Machine Learning Repository (except for synthetic).

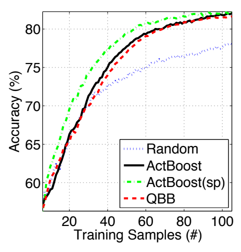
Unbiased Initialization: The initial set L^0 is resampled at every trial to avoid any initialization bias (which is addressed next). BOX and BANANA are two dimensional datasets (see suppl.), MUSHROOM is a 22 dimensional dataset. On the $2D$ datasets, we illustrate the advantage of learning in sparse version space (Fig 5(c) 5(b)). Instead of the entire space, ActBoost(sp) operates only on the 10% of the weak hypotheses. This subset also contains the true sparse support which is determined by training with an entire pool labeled. As our theoretical results suggest, ActBoost(sp) achieves better performance than ActBoost. The experiments in Fig 5 only show marginal advantage of ActBoost since the initialization bias is removed by averaging. The simulations also support the negative result of Theorem 1: a reduction of version space of the full ensemble does not guarantee a decrease in generalization error.

Biased Initialization: We further illustrate the robustness of ActBoost to initialization bias (see Section 2.1). We transform a multi-class dataset into a binary dataset consisting of three clusters. IRIS is a four dimensional dataset of three classes. Classes 1 and 3 are combined into one class to form a binary dataset consisting of three clusters. DERMATOLOGY has 34 dimensions and 5 classes. Similarly, we designate the first three classes as clusters and combine the first and third to form two classes. SOY has 35 dimensions and 19 classes. We use classes 4, 8, 14 as clusters to form a binary dataset. Gaussian Clusters is a synthetic two dimensional dataset consisting of three clusters. (see suppl.) For each experiment (Fig 6), we force the initial set L^0 to be sampled from only the first two clusters. This approach simulates the worst case initialization bias. For all datasets, ActBoost remains robust while QBB does not locate the third cluster until the first two are exhausted.

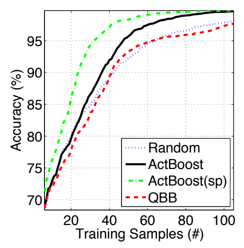
[4]$D = 16$ shows small improvement, but $D = 8$ is used to speed up computation



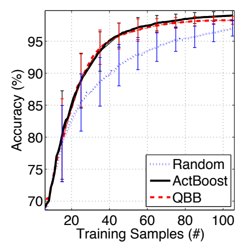
(a) Samples D



(b) Banana

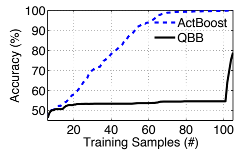


(c) Box

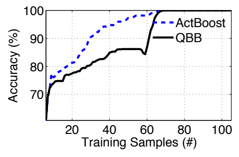


(d) Mushroom

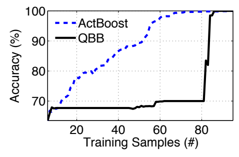
Figure 5: Accuracy vs. # labeled examples. As a function of ActBoost classifier samples D (5(a)): more samples provide a better approximation of the GBS metric integral, resulting in better performance. $2D$ datasets: BANANA (5(b)) and BOX (5(c)). ActBoost(sp) learns in the sparse version space and performs better than ActBoost which operates on the full version space. Multivariate Dataset: MUSHROOM (5(d)). ActBoost does not show significant performance improvement over QBB.



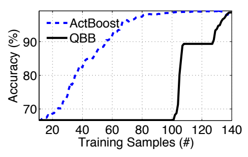
(a) Gauss Clusters



(b) Dermatology



(c) Soy



(d) Iris

Figure 6: Robustness to Initialization Bias: binary datasets consisting of three clusters are formed from multi-class datasets (except for gauss clusters). Both algorithms are initialized only from the first two clusters; ActBoost is not affected.

References

[Abe & Mamitsuka, 1998] Abe, N. & Mamitsuka, H. (1998). Query learning strategies using boosting and bagging. In *Proceedings of the 15th International Conference on Machine Learning* (pp. 1–9).

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

[Campbell et al., 2000] Campbell, C., Cristianini, N., & Smola, A. (2000). Query learning with large margin classifiers. In *Proceedings 17th International Conference on Machine Learning* (pp. 111–118).

[Cohn et al., 1994] Cohn, D., Ladner, R., & Waibel, A. (1994). Improving generalization with active learning. In *Machine Learning* (pp. 201–221).

[Dasgupta & Hsu, 2008] Dasgupta, S. & Hsu, D. (2008). Hierarchical sampling for active learning. In *Proceedings of the 25th International Conference on Machine Learning* (pp. 208–215).

[Freund & Schapire, 1996] Freund, Y. & Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *Proceedings of the 13th International Conference on Machine Learning* (pp. 148–156).

[Freund et al., 1997] Freund, Y., Seung, H. S., Shamir, E., & Tishby, N. (1997). Selective sampling using the query by committee algorithm. In *Machine Learning*.

[Gilad-Bachrach et al., 2005] Gilad-Bachrach, R., Navot, A., & Tishby, N. (2005). Query by committee made real. In *Advances in Neural Information Processing Systems*.

[Guo & Greiner, 2007] Guo, Y. & Greiner, R. (2007). Optimistic active learning using mutual information. In *Proceedings of the 20th international joint conference on Artifical intelligence* (pp. 823–829).

[Kannan & Narayanan, 2009] Kannan, R. & Narayanan, H. (2009). Random walks on polytopes and an affine interior point method for linear programming. In *Proceedings of the 41st annual ACM symposium on Theory of computing* (pp. 561–570).

[Koltchinskii & Panchenko, 2005] Koltchinskii, V. & Panchenko, D. (2005). Complexities of convex combinations and bounding the generalization error in classification. *The Annals of Statistics*, 33(4), pp. 1455–1496.

[Lovász & Vempala, 2004] Lovász, L. & Vempala, S. (2004). Hit-and-run from a corner. In *Proceedings of the 36th annual ACM Symposium on Theory of Computing* (pp. 310–314).

[Nguyen & Smeulders, 2004] Nguyen, H. T. & Smeulders, A. (2004). Active learning using pre-clustering. In *Proceedings of the 21st international conference on Machine learning* (pp.$\tilde{7}$9).

[Nowak, 2009] Nowak, R. D. (2009). The geometry of generalized binary search. In *Advances in Neural Information Processing Systems*.

[Schapire et al., 1997] Schapire, R. E., Freund, Y., Bartlett, P., & Lee, W. S. (1997). Boosting the margin: A new explanation for the effectiveness of voting methods.

[Settles, 2010] Settles, B. (2010). *Active Learning Literature Survey*. Technical report, University of Wisconsin–Madison.

[Seung et al., 1992] Seung, H. S., Opper, M., & Sompolinsky, H. (1992). Query by committee. In *Proceedings of the 5th Workshop on Computational Learning Theory* (pp. 287–294).

[Taylor et al., 2010] Taylor, Y., Zhen, X., Xiang, J., Ramadge, P. J., & Schapire, R. E. (2010). Speed and sparsity of regularized boosting. *AISTATS2009*.

[Tong & Koller, 2001] Tong, S. & Koller, D. (2001). Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, (pp. 45–66).

[Tur et al., 2003] Tur, G., Schapire, R., & Hakkani-Tur, D. (2003). Active learning for spoken language understanding. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*.