Computer Science and Artificial Intelligence Laboratory

Technical Report

MIT-CSAIL-TR-2005-075
AIM-2005-033

November 17, 2005

Analysis of Perceptron-Based Active Learning

Sanjoy Dasgupta, Adam Tauman Kalai, Claire Monteleoni

massachusetts institute of technology, cambridge, ma 02139 usa — www.csail.mit.edu

Analysis of Perceptron-Based Active Learning

Sanjoy Dasgupta[1,⋆], Adam Tauman Kalai[2], and Claire Monteleoni[3,⋆⋆]

[1] UCSD CSE, 9500 Gilman Drive #0114, La Jolla, CA 92093
dasgupta@cs.ucsd.edu
[2] TTI-Chicago, 1427 East 60th Street, Second Floor, Chicago, IL 60637
kalai@tti-c.org
[3] MIT CSAIL, 32 Vassar Street, Cambridge, MA 02139
cmontel@csail.mit.edu

Abstract. We start by showing that in an active learning setting, the Perceptron algorithm needs $\Omega(\frac{1}{\epsilon^2})$ labels to learn linear separators within generalization error ϵ. We then present a simple selective sampling algorithm for this problem, which combines a modification of the perceptron update with an adaptive filtering rule for deciding which points to query. For data distributed uniformly over the unit sphere, we show that our algorithm reaches generalization error ϵ after asking for just $\tilde{O}(d \log \frac{1}{\epsilon})$ labels. This exponential improvement over the usual sample complexity of supervised learning has previously been demonstrated only for the computationally more complex query-by-committee algorithm.

1 Introduction

In many machine learning applications, unlabeled data is abundant but labeling is expensive. This distinction is not captured in the standard PAC or online models of supervised learning, and has motivated the field of *active learning*, in which the labels of data points are initially hidden, and the learner must pay for each label it wishes revealed. If query points are chosen randomly, the number of labels needed to reach a target generalization error ϵ, at a target confidence level $1 - \delta$, is similar to the sample complexity of supervised learning. The hope is that there are alternative querying strategies which require significantly fewer labels.

To date, the single most dramatic demonstration of the potential of active learning is perhaps Freund et al.'s analysis of the query-by-committee (QBC) learning algorithm [7]. In their *selective sampling* model, the learner observes a stream of unlabeled data and makes spot decisions about whether or not to ask for a point's label. They show that if the data is drawn uniformly from the surface of the unit sphere in $\mathbb{R}^d$, and the hidden labels correspond perfectly to a homogeneous (i.e., through the origin) linear separator from this same distribution, then it is possible to achieve generalization error ϵ after seeing $\tilde{O}(\frac{d}{\epsilon} \log \frac{1}{\epsilon})$

⋆ Funded by the NSF, under grant IIS-0347646.
⋆⋆ Work done primarily while at TTI-Chicago.

P. Auer and R. Meir (Eds.): COLT 2005, LNAI 3559, pp. 249–263, 2005.
© Springer-Verlag Berlin Heidelberg 2005

points and requesting just $\tilde{O}(d \log \frac{1}{\epsilon})$ labels:[1] an exponential improvement over the usual $\tilde{O}(\frac{d}{\epsilon})$ sample complexity of learning linear separators in a supervised setting.[2] This remarkable result is tempered somewhat by the complexity of the QBC algorithm, which involves random sampling from intermediate version spaces; the complexity of the update step scales (polynomially) with the number of updates performed.

In this paper, we show how a simple modification of the perceptron update can be used to achieve the same sample complexity bounds (within $\tilde{O}$ factors), under the same streaming model and the same uniform input distribution. Unlike QBC, we do not assume a distribution over target hypotheses, and our algorithm does not need to store previously seen data points, only its current hypothesis.

Our algorithm has the following structure.

```
Set initial hypothesis v_0 ∈ R^d
For t = 0, 1, 2, ...
  Receive unlabeled point x_t
  Make a prediction SGN(v_t · x_t)
  Filtering step: Decide whether to ask for x_t's label
  If label y_t is requested:
    Update step: Set v_{t+1} based on v_t, x_t, y_t
    Adjust filtering rule
  else:  v_{t+1} = v_t
```

Update Step. It turns out that the regular perceptron update, that is,

$$\text{if } (x_t, y_t) \text{ is misclassified then } v_{t+1} = v_t + y_t x_t$$

cannot yield an error rate better than $\Omega(1/\sqrt{l_t})$, where l_t is the number of labels queried up to time t, no matter what filtering scheme is used. In particular:

Theorem 1. *Consider any sequence of data points $x_0, x_1, x_2, \ldots$ which is perfectly classified by some linear separator $u \in \mathbb{R}^d$. If θ_t is the angle between u and v_t, then for any $t \geq 0$, if $\theta_{t+1} \leq \theta_t$ then $\sin \theta_t \geq 1/(5\sqrt{l_t + \|v_0\|^2})$.*

This holds regardless of how the data is produced. When the points are distributed uniformly over the unit sphere, $\theta_t \geq \sin \theta_t$ (for $\theta_t \leq \frac{\pi}{2}$) is proportional to the error rate of v_t.

So instead we use a slightly modified update rule:

$$\text{if } (x_t, y_t) \text{ is misclassified then } v_{t+1} = v_t - 2(v_t \cdot x_t)x_t$$

(where x_t is assumed normalized to unit length). Note that the update can also be written as $v_{t+1} = v_t + 2y_t|v_t \cdot x_t|x_t$, since updates are only made on mistakes,

[1] In this paper, the $\tilde{O}$ notation is used to suppress terms in $\log d, \log \log \frac{1}{\epsilon}$ and $\log \frac{1}{\delta}$.

[2] This label complexity can be seen to be optimal by counting the number of spherical caps of radius ϵ that can be packed onto the surface of the unit sphere in $\mathbb{R}^d$.

in which case $y_t \neq \texttt{SGN}(v_t \cdot x_t)$, by definition. Thus we are scaling the standard perceptron's additive update by a factor of $2|v_t \cdot x_t|$ to avoid oscillations caused by points close to the hyperplane represented by the current hypothesis. The same rule, but without the factor of two, has been used in previous work [3] on learning linear classifiers from noisy data, in a batch setting. We are able to show that our formulation has the following generalization performance in a supervised (non-active) setting.

Theorem 2. *When the modified Perceptron algorithm is applied in a sequential supervised setting, with data points x_t drawn independently and uniformly at random from the surface of the unit sphere in $\mathbb{R}^d$, then with probability $1-\delta$, after $O(d(\log \frac{1}{\epsilon} + \log \frac{1}{\delta}))$ mistakes, its generalization error is at most ϵ.*

This contrasts favorably with the $\tilde{O}(\frac{d}{\epsilon^2})$ mistake bound of the Perceptron algorithm, and a more recent variant, on the same distribution [2, 12]. As a lower bound for standard Perceptron, Theorem 1 also applies in the supervised case, as it holds for all filtering rules, including viewing all the labels. The bound on labels, $\Omega(\frac{1}{\epsilon^2})$, lower bounds mistakes as well, since the number of labels is minimized when every label yields a mistake, and thus an update.

The PAC sample complexity of the problem under the uniform distribution is $\tilde{\Theta}(\frac{d}{\epsilon})$ (lower bound [10], and upper bound [11]). Yet since not all examples yield mistakes, mistake bounds can be lower than sample bounds. A similar statement holds in the active learning case: bounds on labels can be lower than sample bounds, since the algorithms are allowed to filter which samples to label.

Filtering Step. Given the limited information the algorithm keeps, a natural filtering rule is to query points x_t when $|v_t \cdot x_t|$ is less than some threshold s_t. The choice of s_t is crucial. If it is too large, then only a miniscule fraction of the points queried will actually be misclassified – almost all labels will be wasted. On the other hand, if s_t is too small, then the waiting time for a query might be prohibitive, and when an update is actually made, the magnitude of this update might be tiny.

Therefore, we set the threshold adaptively: we start s high, and keep dividing it by two until we reach a level where there are enough misclassifications amongst the points queried. This filtering strategy makes possible our main theorem, again for data from the uniform distribution over the unit sphere in $\mathbb{R}^d$.

Theorem 3. *With probability $1-\delta$, if the active modified Perceptron algorithm is given a stream of $\tilde{O}(\frac{d}{\epsilon}\log\frac{1}{\epsilon})$ unlabeled points, it will request $\tilde{O}(d\log\frac{1}{\epsilon})$ labels, make $\tilde{O}(d\log\frac{1}{\epsilon})$ errors (on all points, labeled or not), and have final error $\leq \epsilon$.*

2 Related Work

Our approach relates to the literature on selective sampling [7, 4]. We have already discussed query-by-committee [7], which is perhaps the strongest positive result in active learning to date. There have been numerous applications of this method and also several refinements (see, for instance, [8, 6]).

Cesa-Bianchi, Gentile, and Zaniboni [4] have recently analyzed an algorithm which conforms to roughly the same template as ours but differs in both the update and filtering rule – it uses the regular perceptron update and it queries points x_t according to a fixed, randomized rule which favors small $|v_t \cdot x_t|$. The authors make no distributional assumptions on the input and they show that in terms of worst-case hinge-loss bounds, their algorithm does about as well as one which queries *all* labels. The actual fraction of points queried varies from data set to data set. In contrast, our objective is to achieve a target generalization error with minimum label complexity, although we do also obtain a mistake bound (on both labeled and unlabeled points) under our distributional assumption.

It is known that active learning does not always give a large improvement in the sample complexity of learning linear separators. For instance, in our setting where data is distributed uniformly over the unit sphere, recent work has shown that if the target linear separator is allowed to be non-homogeneous, then the number of labels required to reach error ϵ is $\Omega(1/\epsilon)$, no matter what active learning scheme is used [5]. This lower bound also applies to learning homogeneous linear separators with respect to an arbitrary distribution.

Many active learning schemes for linear separators (or probabilistic analogues) have been proposed in the literature. Several of these are similar in spirit to our heuristic, in that they query points close to the margin, and seem to have enjoyed some empirical success; e.g., [9]. Finally, there is a rich body of theory on a related model in which it is permissible to create query points synthetically; a recent survey by Angluin [1] summarizes key results.

3 Preliminaries

In our model, all data x_t lie on the surface of the unit ball in $\mathbb{R}^d$, which we will denote as S:

$$S = \left\{ x \in \mathbb{R}^d \;\middle|\; \|x\| = 1 \right\}.$$

Their labels y_t are either -1 or $+1$, and the target function is a half-space $u \cdot x \geq 0$ represented by a unit vector $u \in \mathbb{R}^d$ which classifies all points perfectly, that is, $y_t(u \cdot x_t) > 0$ for all t, with probability one.

For any vector $v \in \mathbb{R}^d$, we define $\hat{v} = \frac{v}{\|v\|}$ to be the corresponding unit vector.

Our lower bound (Theorem 1) holds regardless of how the data are generated; thereafter we will assume that the data points x_t are drawn independently from the uniform distribution over S. This implies that any hypothesis $v \in \mathbb{R}^d$ has error

$$\epsilon(v) = P_{x \in S}[\mathtt{SGN}(v \cdot x) \neq \mathtt{SGN}(u \cdot x)] = \frac{\arccos(u \cdot \hat{v})}{\pi}.$$

We will use a few useful inequalities for θ on the interval $(0, \frac{\pi}{2}]$.

$$\frac{4}{\pi^2} \leq \frac{1 - \cos\theta}{\theta^2} \leq \frac{1}{2}, \tag{1}$$

$$\frac{2}{\pi}\theta \leq \sin\theta \leq \theta \tag{2}$$

Equation (1) can be verified by checking that for θ in this interval, $\frac{1-\cos\theta}{\theta^2}$ is a decreasing function, and evaluating it at the endpoints.

We will also make use of the following lemma.

Lemma 1. *For any fixed unit vector a and any $\gamma \leq 1$,*

$$\frac{\gamma}{4} \leq P_{x \in S}\left[|a \cdot x| \leq \frac{\gamma}{\sqrt{d}}\right] \leq \gamma \tag{3}$$

The proof is deferred to the appendix.

4 A Lower Bound for the Perceptron Update

Consider an algorithm of the following form:

```
Pick some v_0 ∈ R^d
Repeat for t = 0, 1, 2, ...:
  Get some (x, y) for which y(v_t · x) ≤ 0
  v_{t+1} = v_t + yx
```

On any update,

$$v_{t+1} \cdot u = v_t \cdot u + y(x \cdot u). \tag{4}$$

Thus, if we assume for simplicity that $v_0 \cdot u \geq 0$ (we can always just start count when this first occurs) then $v_t \cdot u \geq 0$ always, and the angle between u and v_t is always acute. Denoting this angle by θ_t, we get

$$\|v_t\| \cos\theta_t = v_t \cdot u.$$

The update rule also implies

$$\|v_{t+1}\|^2 = \|v_t\|^2 + 1 + 2y(v_t \cdot x). \tag{5}$$

Thus $\|v_t\|^2 \leq t + \|v_0\|^2$ for all t. In particular, this means that Theorem 1 is an immediate consequence of the following lemma.

Lemma 2. *Assume $v_0 \cdot u \geq 0$ (i.e., start count when this first occurs). Then*

$$\theta_{t+1} \leq \theta_t \quad \Rightarrow \quad \sin\theta_t \geq \min\left\{\frac{1}{3}, \frac{1}{5\|v_t\|}\right\}.$$

Proof. Figure 1 shows the unit circle in the plane defined by u and v_t. The dot product of any point $x \in \mathbb{R}^d$ with either u or v_t depends only upon the projection of x into this plane. The point is misclassified when its projection lies in the shaded region. For such points, $y(u \cdot x)$ is at most $\sin\theta_t$ (point (i)) and $y(v_t \cdot x)$ is at least $-\|v_t\| \sin\theta_t$ (point (ii)).

Combining this with equations (4) and (5), we get

$$v_{t+1} \cdot u \leq v_t \cdot u + \sin\theta_t$$
$$\|v_{t+1}\|^2 \geq \|v_t\|^2 + 1 - 2\|v_t\| \sin\theta_t$$

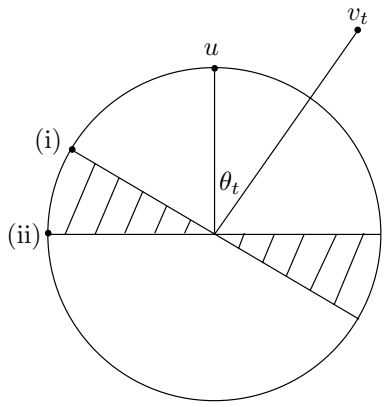

Fig. 1. The plane defined by u and v_t

To establish the lemma, we first assume $\theta_{t+1} \leq \theta_t$ and $\sin\theta_t \leq \frac{1}{5\|v_t\|}$, and then conclude that $\sin\theta_t \geq \frac{1}{3}$.

$\theta_{t+1} \leq \theta_t$ implies

$$\cos^2\theta_t \;\leq\; \cos^2\theta_{t+1} \;=\; \frac{(u\cdot v_{t+1})^2}{\|v_{t+1}\|^2} \;\leq\; \frac{(u\cdot v_t + \sin\theta_t)^2}{\|v_t\|^2 + 1 - 2\|v_t\|\sin\theta_t}.$$

The final denominator is positive since $\sin\theta_t \leq \frac{1}{5\|v_t\|}$. Rearranging,

$$(\|v_t\|^2 + 1 - 2\|v_t\|\sin\theta_t)\cos^2\theta_t \;\leq\; (u\cdot v_t)^2 + \sin^2\theta_t + 2(u\cdot v_t)\sin\theta_t$$

and using $\|v_t\|\cos\theta_t = (u\cdot v_t)$:

$$(1 - 2\|v_t\|\sin\theta_t)\cos^2\theta_t \;\leq\; \sin^2\theta_t + 2\|v_t\|\sin\theta_t\cos\theta_t$$

Again, since $\sin\theta_t \leq \frac{1}{5\|v_t\|}$, it follows that $(1 - 2\|v_t\|\sin\theta_t) \geq \frac{3}{5}$ and that $2\|v_t\|\sin\theta_t\cos\theta_t \leq \frac{2}{5}$. Using $\cos^2 = 1 - \sin^2$, we then get

$$\frac{3}{5}(1 - \sin^2\theta_t) \;\leq\; \sin^2\theta_t + \frac{2}{5}$$

which works out to $\sin^2\theta_t \geq \frac{1}{8}$, implying $\sin\theta_t > \frac{1}{3}$. □

The problem is that the perceptron update can be too large. In $\mathbb{R}^2$ (e.g. Figure 1), when θ_t is tiny, the update will cause v_{t+1} to overshoot the mark and swing too far to the other side of u, unless $\|v_t\|$ is very large: to be precise, we need $\|v_t\| = \Omega(1/\sin\theta_t)$. But $\|v_t\|$ grows slowly, at best at a rate of $\sqrt{t}$. If $\sin\theta_t$ is proportional to the error of v_t, as in the case of data distributed uniformly over the unit sphere, this means that the perceptron update cannot stably maintain an error rate $\leq \epsilon$ until $t = \Omega(1/\epsilon^2)$.

```
Inputs: dimensionality d and desired number of updates
(mistakes) M.
  Let v_1 = x_1 y_1 for the first example (x_1, y_1).
  For t = 1 to M:
    Let (x_t, y_t) be the next example with y(x · v_t) < 0.
    v_{t+1} = v_t − 2(v_t · x_t)x_t .
```

Fig. 2. The (non-active) modified Perceptron algorithm. The standard Perceptron update, $v_{t+1} = v_t + y_t x_t$, is in the same direction (note $y_t = -\mathtt{SGN}(v_t \cdot x_t)$) but different magnitude (scaled by a factor of $2|v_t \cdot x_t|$)

5 The Modified Perceptron Update

We now describe the modified Perceptron algorithm. Using a simple modification to the standard perceptron update yields the fast convergence we will prove subsequently. Unlike with standard Perceptron, this modification ensures that $v_t \cdot u$ is increasing, i.e., the error of v_t is monotonically decreasing. Another difference from the standard update (and other versions) is that the magnitude of $\|v_t\| = 1$, which is convenient for our analysis.

The modified Perceptron algorithm is shown in Figure 2. We now show that the norm of v_t stays at one. Note that $\|v_1\| = 1$ and

$$\|v_{t+1}\|^2 = \|v_t\|^2 + 4(v_t \cdot x_t)^2\|x_t\|^2 - 4(v_t \cdot x_t)^2 = 1$$

by induction. In contrast, for the standard perceptron update, the magnitude of v_t is important and normalized vectors cannot be used.

With the modified update, the error can only decrease, because $v_t \cdot u$ only increases:

$$v_{t+1} \cdot u = v_t \cdot u - 2(v_t \cdot x_t)(x_t \cdot u) = v_t \cdot u + 2|v_t \cdot x_t||x_t \cdot u|.$$

The second equality follows from the fact that v_t misclassified x_t. Thus $v_t \cdot u$ is increasing, and the increase can be bounded from below by showing that $|v_t \cdot x_t||x_t \cdot u|$ is large. This is a different approach from previous analyses.

Blum et al. [3] used an update similar to ours, but without the factor of two. In general, one can consider modified updates of the form $v_{t+1} = v_t - \alpha(v_t \cdot x_t)x_t$. When $\alpha \neq 2$, the vectors v_t no longer remain of fixed length; however, one can verify that their corresponding unit vectors $\hat{v}_t$ satisfy

$$\hat{v}_{t+1} \cdot u = (\hat{v}_t \cdot u + \alpha|\hat{v}_t \cdot x_t||x_t \cdot u|)/\sqrt{1 - \alpha(2-\alpha)(\hat{v}_t \cdot x_t)^2},$$

and thus any choice of $\alpha \in [0, 2]$ guarantees non-increasing error. Blum et al. used $\alpha = 1$ to guarantee progress in the denominator (their analysis did not rely on progress in the numerator) as long as $\hat{v}_t \cdot u$ and $(\hat{v}_t \cdot x_t)^2$ were bounded away from 0. Their approach was used in a batch setting as one piece of a more complex algorithm for noise-tolerant learning. In our sequential framework, we can bound $|\hat{v}_t \cdot x_t||x_t \cdot u|$ away from 0 in expectation, under the uniform distribution, and

hence the choice of $\alpha = 2$ is most convenient, but $\alpha = 1$ would work as well. Although we do not further optimize our choice of the constant α, this choice itself may yield interesting future work, perhaps by allowing it to be a function of the dimension.

5.1 Analysis of (Non-active) Modified Perceptron

How large do we expect $|v_t \cdot x_t|$ and $|u \cdot x_t|$ to be for an error (x_t, y_t)? As we shall see, in d dimensions, one expects each of these terms to be on the order of $d^{-1/2} \sin\theta_t$, where $\sin\theta_t = \sqrt{1-(v_t \cdot u)^2}$. Hence, we might expect their product to be about $(1-(v_t \cdot u)^2)/d$, which is how we prove the following lemma.

Note, we have made little effort to optimize constant factors.

Lemma 3. *For any v_t, with probability at least $\frac{1}{3}$,*

$$1 - v_{t+1} \cdot u \leq (1 - v_t \cdot u)\left(1 - \frac{1}{50d}\right).$$

There exists a constant $c > 0$, such that with probability at least $\frac{63}{64}$, for any v_t,

$$1 - v_{t+1} \cdot u \leq (1 - v_t \cdot u)\left(1 - \frac{c}{d}\right).$$

Proof. We show only the first part of the lemma. The second part is quite similar. We will argue that each of $|v_t \cdot x_t|$,$|u \cdot x_t|$ is "small" with probability at most $1/3$. This means, by the union bound, that with probability at least $1/3$, they are both sufficiently large.

The error rate of v_t is θ_t/π, where $\cos\theta_t = v_t \cdot u$. Also define the error region $\xi_t = \{x \in S \,|\, \mathtt{SGN}(v_t \cdot x) \neq \mathtt{SGN}(u \cdot x)\}$. By Lemma 1, for an x drawn uniformly from the sphere,

$$P_{x \in S}\left[|v_t \cdot x| \leq \frac{\theta_t}{3\pi\sqrt{d}}\right] \leq \frac{\theta_t}{3\pi}.$$

Using $P[A|B] \leq P[A]/P[B]$, we have,

$$P_{x \in S}\left[|v_t \cdot x| \leq \frac{\theta_t}{3\pi\sqrt{d}} \;\middle|\; x \in \xi_t\right] \leq \frac{P_{x \in S}[|v_t \cdot x| \leq \frac{\theta_t}{3\pi\sqrt{d}}]}{P_{x \in S}[x \in \xi_t]} \leq \frac{\theta_t/(3\pi)}{\theta_t/\pi} = \frac{1}{3}$$

Similarly for $|u \cdot x|$, and by the union bound the probability that $x \in \xi$ is within margin $\frac{\theta}{3\pi\sqrt{d}}$ from either u or v is at most $\frac{2}{3}$. Since the updates only occur if x is in the error region, we now have a lower bound on the expected magnitude of $|v_t \cdot x||u \cdot x|$.

$$P_{x \in S}\left[|v_t \cdot x||u \cdot x| \geq \frac{\theta_t^2}{(3\pi\sqrt{d})^2} \,\middle|\, x \in \xi_t\right] \geq \frac{1}{3}.$$

Hence, we know that with probability at least 1/3, $|v_t \cdot x||u \cdot x| \geq \frac{1-(v_t \cdot u)^2}{100d}$, since $\theta_t^2 \geq \sin^2 \theta_t = 1 - (v_t \cdot u)^2$ and $(3\pi)^2 < 100$. In this case,

$$\begin{aligned} 1 - v_{t+1} \cdot u &\leq 1 - v_t \cdot u - 2|v_t \cdot x_t||u \cdot x_t| \\ &\leq 1 - v_t \cdot u - \frac{1 - (v_t \cdot u)^2}{50d} \\ &\leq (1 - v_t \cdot u)\left(1 - \frac{1 + v_t \cdot u}{50d}\right) \end{aligned}$$

□

Finally, we give a high-probability bound, i.e. Theorem 2, stated here with proof.

Theorem 2. *With probability $1 - \delta$, after $M = O(d(\log \frac{1}{\epsilon} + \log \frac{1}{\delta}))$ mistakes, the generalization error of the modified Perceptron algorithm is at most ϵ.*

Proof. By the above lemma, we can conclude that, for any vector v_t,

$$E_{x_t \in \xi_t}[1 - v_{t+1} \cdot u] \leq (1 - v_t \cdot u)\left(1 - \frac{1}{3(50d)}\right).$$

This is because with $\geq 1/3$ probability it goes down by a factor of $1 - \frac{1}{50d}$ and with the remaining $\leq 2/3$ probability it does not increase. Hence, after M mistakes,

$$E[1 - v_M \cdot u] \leq (1 - v_1 \cdot u)\left(1 - \frac{1}{150d}\right)^M \leq \left(1 - \frac{1}{150d}\right)^M,$$

since $v_1 \cdot u \geq 0$. By Markov's inequality,

$$P\left[1 - v_M \cdot u \geq \left(1 - \frac{1}{150d}\right)^M \delta^{-1}\right] \leq \delta.$$

Finally, using (1) and $\cos \theta_M = v_M \cdot u$, we see $P[\frac{4}{\pi^2}\theta_M^2 \geq (1 - \frac{1}{150d})^M \delta^{-1}] \leq \delta$. Using $M = 150d \log(1/\epsilon\delta)$ gives $P[\frac{\theta_M}{\pi} \geq \epsilon] \leq \delta$ as required. □

6 An Active Modified Perceptron

The active version of the modified Perceptron algorithm is shown in Figure 3. The algorithm is similar to the algorithm of the previous section, in its update step. For its filtering rule, we maintain a threshold s_t and we only ask for labels of examples with $|v_t \cdot x_t| \leq s_t$. We decrease this threshold adaptively over time, starting at $s_1 = 1/\sqrt{d}$ and reducing it by a factor of two whenever we have a run of labeled examples on which we are correct.

For Theorem 3, we select values of R, L that yield ϵ error with probability at least $1 - \delta$. The idea of the analysis is as follows:

Inputs: Dimensionality d, maximum number of labels L,
and patience R.
  $v_1 = x_1 y_1$ for the first example (x_1, y_1).
  $s_1 = 1/\sqrt{d}$
  For $t = 1$ **to** L:
    Wait for the next example x : $|x \cdot v_t| \leq s_t$ and query its label.
    Call this labeled example (x_t, y_t).
    If $(x_t \cdot v_t) y_t < 0$, **then**:
      $v_{t+1} = v_t - 2(v_t \cdot x_t) x_t$
      $s_{t+1} = s_t$
    else:
      $v_{t+1} = v_t$
      If predictions were correct on R consecutive labeled examples (i.e. $(x_i \cdot v_i) y_i \geq 0 \ \forall i \in \{t - R + 1, t - R + 2, \ldots, t\}$), **then set** $s_{t+1} = s_t/2$, **else** $s_{t+1} = s_t$.

Fig. 3. An active version of the modified Perceptron algorithm

Definition 1. *We say the tth update is "good" if,*

$$1 - v_{t+1} \cdot u \leq (1 - v_t \cdot u) \left(1 - \frac{c}{d}\right).$$

(The constant c is from Lemma 3.)

1. (Lemma 4) First, we argue that s_t is not too small (we do not decrease s_t too quickly). Assuming this is the case, then 2 and 3 hold.
2. (Lemma 6) We query for labels on at least an expected $1/32$ of all *errors*. In other words, some errors may go undetected because we do not ask for their labels, but the number of mistakes total should not be much more than 32 times the number of updates we actually perform.
3. (Lemma 7) Each update is *good* (Definition 1) with probability at least $1/2$.
4. (Theorem 3) Finally, we conclude that we cannot have too many label queries, updates, or total errors, because half of our updates are good, $1/32$ of our errors are updates, and about $1/R$ of our labels are updates.

We first lower-bound s_t with respect to our error, showing that, with high probability, the threshold s_t is never too small.

Lemma 4. *With probability at least $1 - L\left(\frac{3}{4}\right)^R$, we have:*

$$s_t \geq \sqrt{\frac{1 - (u \cdot v_t)^2}{16d}} \text{ for } t = 1, 2, \ldots, L, \text{ simultaneously.} \tag{6}$$

Before proving this lemma, it will be helpful to show the following lemma. As before, let us define $\xi_t = \{x \in S | (x \cdot v_t)(x \cdot u) < 0\}$.

Lemma 5. *For any* $\gamma \in \left(0, \sqrt{\frac{1-(u\cdot v_t)^2}{4d}}\right]$,

$$P_{x_t \in S}\left[x_t \in \xi_t \;\middle|\; |x_t \cdot v_t| < \gamma\right] \geq \frac{1}{4}$$

Proof. Let x be a random example from S such that $|x \cdot v_t| < \gamma$ and, without loss of generality, suppose that $0 \leq x \cdot v_t \leq \gamma$. Then we want to calculate the probability we err, i.e. $u \cdot x < 0$. We can decompose $x = x' + (x \cdot v_t)v_t$ where $x' = x - (x \cdot v_t)v_t$ is the component of x orthogonal to v_t, i.e. $x' \cdot v_t = 0$. Similarly for $u' = u - (u \cdot v_t)v_t$. Hence,

$$u \cdot x = (u' + (u \cdot v_t)v_t) \cdot (x' + (x \cdot v_t)v_t) = u' \cdot x' + (u \cdot v_t)(x \cdot v_t)$$

In other words, we err iff $u' \cdot x' < -(u \cdot v_t)(x \cdot v_t)$. Using $u \cdot v_t \in [0, 1]$ and since $x \cdot v_t \in [0, \sqrt{(1-(u \cdot v_t)^2)/(4d)}]$, we conclude that if,

$$u' \cdot x' < -\sqrt{\frac{1-(u \cdot v_t)^2}{4d}} \tag{7}$$

then we must err. Also, let $\hat{x}' = \frac{x'}{\|x'\|}$ be the unit vector in the direction of x'. It is straightforward to check that $\|x'\| = \sqrt{1-(x \cdot v_t)^2}$. Similarly, for u we define $\hat{u}' = \frac{u'}{\sqrt{1-(u\cdot v_t)^2}}$. Substituting these into (7), we must err if, $\hat{u}' \cdot \hat{x}' < -1/\sqrt{4d(1-(x \cdot v_t))^2}$, and since $\sqrt{1-(x \cdot v_t)^2} \geq \sqrt{1-1/(4d)}$, it suffices to show that,

$$P_{x \in S}\left[\hat{u}' \cdot \hat{x}' < \frac{-1}{\sqrt{4d(1-1/(4d))}} \;\middle|\; 0 \leq x \cdot v_t \leq \gamma\right] \geq \frac{1}{4}$$

What is the probability that this happens? Well, one way to pick $x \in S$ would be to first pick $x \cdot v_t$ and then to pick $\hat{x}'$ uniformly at random from the set $S' = \{\hat{x}' \in S | \hat{x}' \cdot v_t = 0\}$, which is a unit sphere in one fewer dimensions. Hence the above probability does not depend on the conditioning. By Lemma 1, for any unit vector $a \in S'$, the probability that $|\hat{u}' \cdot a| \leq 1/\sqrt{4(d-1)}$ is at most $1/2$, so with probability at least $1/4$ (since the distribution is symmetric), the signed quantity $\hat{u}' \cdot \hat{x}' < -1/\sqrt{4(d-1)} < -1/\sqrt{4d(1-1/(4d))}$. □

We are now ready to prove Lemma 4.

Proof (of Lemma 4). Suppose that condition (6) fails to hold for some t's. Let t be the smallest number such that (6) fails. By our choice of s_1, clearly $t > 1$. Moreover, since t is the smallest such number, and $u \cdot v_t$ is increasing, it must be the case that $s_t = s_{t-1}/2$, that is we just saw a run of R labeled examples (x_i, y_i), for $i = t-R, \ldots, t-1$, with no mistakes, $v_i = v_t$, and

$$s_i = 2s_t < \sqrt{\frac{1-(u \cdot v_t)^2}{4d}} = \sqrt{\frac{1-(u \cdot v_i)^2}{4d}}. \tag{8}$$

Such an event is highly unlikely, however, for any t. In particular, from Lemma 5, we know that the probability of (8) holding for any particular i and the algorithm not erring is at most $3/4$. Thus the chance of having any such run of length R is at most $L(3/4)^R$.

Lemma 5 also tells us something interesting about the fraction of errors that we are missing because we do not ask for labels. In particular,

Lemma 6. *Given that $s_t \geq \sqrt{(1-(u\cdot v_t)^2)/(16d)}$, upon the tth update, each erroneous example is queried with probability at least 1/32, i.e.,*

$$P_{x\in S}\left[|x\cdot v_t| \leq s_t \;\middle|\; x\in\xi_t\right] \geq \frac{1}{32}.$$

Proof. Using Lemmas 5 and 1, we have

$$\begin{aligned} P_{x\in S}\left[x\in\xi_t \wedge |x\cdot v_t|\leq s_t\right] &\geq P_{x\in S}\left[x\in\xi_t \wedge |x\cdot v_t| \leq \sqrt{\frac{1-(u\cdot v_t)^2}{16d}}\right] \\ &\geq \frac{1}{4}P_{x\in S}\left[|x\cdot v_t| \leq \sqrt{\frac{1-(u\cdot v_t)^2}{16d}}\right] \\ &\geq \frac{1}{64}\sqrt{1-(u\cdot v_t)^2} = \frac{1}{64}\sin\theta_t \\ &\geq \frac{\theta_t}{32\pi} \end{aligned}$$

For the last inequality, we have used (2). However, $P_{x\in S}[x\in\xi_t] = \theta_t/\pi$, so we are querying an error $x\in\xi_t$ with probability at least $1/32$, i.e., the above inequality implies,

$$P_{x\in S}\left[|x\cdot v_t|\leq s_t \;\middle|\; x\in\xi_t\right] = \frac{P_{x\in S}\left[x\in\xi_t\wedge|x\cdot v_t|\leq s_t\right]}{P_{x\in S}[x\in\xi_t]} \geq \frac{\theta_t/(32\pi)}{\theta_t/\pi} = \frac{1}{32}.$$

□

Next, we show that the updates are likely to make progress.

Lemma 7. *Assuming that $s_t \geq \sqrt{(1-(u\cdot v_t)^2)/(16d)}$, a random update is good with probability at least 1/2, i.e.,*

$$P_{x_t\in S}\left[(1-v_{t+1}\cdot u)\leq(1-v_t\cdot u)\left(1-\frac{c}{d}\right) \;\middle|\; |x\cdot v_t|\leq s_t\wedge x_t\in\xi_t\right] \geq \frac{1}{2}.$$

Proof. By Lemma 6, each error is queried with probability 1/32. On the other hand, by Lemma 3 of the previous section, 63/64 of all errors are good. Since we are querying at least 2/64 fraction of all errors, at least half of our queried errors must be good. □

We now have the pieces to guarantee the convergence rate of the active algorithm, thereby proving Theorem 3. This involves bounding both the number of labels that we query as well as the number of total errors, which includes updates as well as errors that were never detected.

Theorem 3. *With probability* $1-\delta$*, using* $L = O\big(d\log\big(\frac{1}{\epsilon\delta}\big)(\log\frac{d}{\delta}+\log\log\frac{1}{\epsilon})\big)$ *labels and making a total number of errors of* $O\big(d\log\big(\frac{1}{\epsilon\delta}\big)(\log\frac{d}{\delta}+\log\log\frac{1}{\epsilon})\big)$*, the final error of the active modified Perceptron algorithm will be* ϵ*, when run with the above* L *and* $R = O(\log\frac{d}{\delta}+\log\log\frac{1}{\epsilon})$.

Proof. Let U be the number of updates performed. We know, by Lemma 4 that with probability $1-L(\frac{3}{4})^R$,

$$s_t \geq \frac{\sin\theta_t}{4\sqrt{d}} \geq \frac{\theta_t}{2\pi\sqrt{d}} \tag{9}$$

for all t. Again, we have used (2). By Lemma 7, we know that for each t which is an update, either (9) fails or

$$E[1-u\cdot v_{t+1}|v_t] \leq (1-u\cdot v_t)\left(1-\frac{c}{2d}\right).$$

Hence, after U updates, using Markov's inequality,

$$P\left[1-u\cdot v_L \geq \frac{4}{\delta}\left(1-\frac{c}{2d}\right)^U\right] \leq \frac{\delta}{4}+L\left(\frac{3}{4}\right)^R.$$

In other words, with probability $1-\delta/4-L(3/4)^R$, we also have

$$U \leq \frac{2d}{c}\log\frac{4}{\delta(1-u\cdot v_L)} \leq \frac{2d}{c}\log\frac{\pi^2}{\delta\theta_L^2} = O\left(d\log\frac{1}{\delta\epsilon}\right),$$

where for the last inequality we used (1). In total, $L \leq R\,(U+\log_2 1/s_L)$. This is because once every R labels we either have at least one update or we decrease s_L by a factor of 2. Equivalently, $s_L \leq 2^{U-L/R}$. Hence, with probability $1-\delta/4-L(3/4)^R$,

$$\frac{\theta_L}{2\pi\sqrt{d}} \leq s_L \leq 2^{O(d\log(1/\delta\epsilon))-L/R}$$

Working backwards, we choose $L/R = \Theta(d\log\frac{1}{\epsilon\delta})$ so that the above expression implies $\frac{\theta_L}{\pi} \leq \epsilon$, as required. We choose,

$$R = 10\log\frac{2L}{\delta R} = \Theta\left(\log\frac{d\log\frac{1}{\epsilon\delta}}{\delta}\right) = O\left(\log\frac{d}{\delta}+\log\log\frac{1}{\epsilon}\right).$$

The first equality ensures that $L(3/4)^R \leq \delta/4$. Hence, for the L and R chosen in the theorem, with probability $1-\frac{3}{4}\delta$, we have error $\theta_L/\pi < \epsilon$. Finally, either condition (9) fails or each error is queried with probability at least 1/32. By the multiplicative Chernoff bound, if there were a total of $E > 64U$ errors, then with probability $\geq 1-\delta/4$, at least $E/64 > U$ would have been caught and used as updates. Hence, with probability at most $1-\delta$, we have achieved the target error using the specified number of labels and incurring the specified number of errors. □

7 Future Directions

The theoretical terrain of active learning is largely an unexplored wilderness. The one nontrivial scenario in which active learning has been shown to give an exponential improvement in sample complexity is that of learning a linear separator for data distributed uniformly over the unit sphere. In this paper, we have demonstrated that this particular case can be solved by a much simpler algorithm than was previously known. It is possible that our algorithm can be molded into something of more general applicability, and so it would be interesting to study its behavior under different circumstances, for instance a different distributional assumption. The uniform distribution is an impressive one to learn against because it is difficult in some ways – most of the data is close to the decision boundary, for instance – but a more common assumption would be to make the two classes Gaussian, or to merely stipulate that they are separated by a margin. How would our algorithm fare under these circumstances?

Acknowledgements

Claire Monteleoni would like to thank Adam Klivans, Brendan McMahan, and Vikas Sindhwani, for various discussions at TTI, and David McAllester for the opportunity to visit. The authors thank the anonymous reviewers for helpful comments used in revision.

References

1. D. Angluin. Queries revisited. *In Proc. 12th Int. Conference on Algorithmic Learning Theory*, LNAI,2225:12–31, 2001.
2. E. B. Baum. The perceptron algorithm is fast for nonmalicious distributions. *Neural Computation*, 2:248–260, 1997.
3. A. Blum, A. Frieze, R. Kannan, and S. Vempala. A polynomial-time algorithm for learning noisy linear threshold functions. In *Proc. 37th IEEE Symposium on the Foundations of Computer Science*, 1996.
4. N. Cesa-Bianchi, C. Gentile, and L. Zaniboni. Worst-case analysis of selective sampling for linear-threshold algorithms. In *Advances in Neural Information Processing Systems 17*, 2004.
5. S. Dasgupta. Analysis of a greedy active learning strategy. In *Advances in Neural Information Processing Systems 17*. 2004.
6. S. Fine, R. Gilad-Bachrach, and E. Shamir. Query by committee, linear separation and random walks. *Theoretical Computer Science*, 284(1):25–51, 2002.
7. Y. Freund, H. S. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28(2-3):133–168, 1997.
8. R. Gilad-Bachrach, A. Navot, and N. Tishby. Kernel query by committee (KQBC). Technical Report 2003-88, Leibniz Center, the Hebrew University, 2003.
9. D. D. Lewis and W. A. Gale. A sequential algorithm for training text classifiers. In *Proc. of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval*, 1994.

10. P. M. Long. On the sample complexity of PAC learning halfspaces against the uniform distribution. *IEEE Transactions on Neural Networks*, 6(6):1556–1559, 1995.
11. P. M. Long. An upper bound on the sample complexity of PAC learning halfspaces with respect to the uniform distribution. *Information Processing Letters*, 87(5):229–23, 2003.
12. R. A. Servedio. On PAC learning using winnow, perceptron, and a perceptron-like algorithm. In *Computational Learning Theory*, pages 296 – 307, 1999.

A Proof of Lemma 1

Let $r = \gamma/\sqrt{d}$ and let A_d be the area of a d-dimensional unit sphere, i.e. the surface of a $(d+1)$-dimensional unit ball.

$$P_x\left[|a \cdot x| \leq r\right] = \frac{\int_{-r}^{r} A_{d-2}(1-z^2)^{\frac{d-2}{2}} dz}{A_{d-1}} = \frac{2A_{d-2}}{A_{d-1}} \int_0^r (1-z^2)^{d/2-1} dz \quad (10)$$

First observe,

$$r(1-r^2)^{d/2-1} \leq \int_0^r (1-z^2)^{d/2-1} dz \leq r \quad (11)$$

For $x \in [0, 0.5]$, $1 - x \geq 4^{-x}$. Hence, for $0 \leq r \leq 2^{-1/2}$,

$$(1-r^2)^{d/2-1} \geq 4^{-r^2(d/2-1)} \geq 2^{-r^2 d}.$$

So we can conclude that the integral of (11) is in $[r/2, r]$ for $r \in [0, 1/\sqrt{d}]$. The ratio $2A_{d-2}/A_{d-1}$ can be shown to be in the range $[\sqrt{d/3}, \sqrt{d}]$ by straightforward induction on d, using the definition of the Γ function, and the fact that $A_{d-1} = 2\pi^{d/2}/\Gamma(d/2)$. □