A Random Sampling based Algorithm for Learning Intersections of Half-spaces

Santosh Vempala
Mathematics Department, MIT
E-mail: vempala@math.mit.edu

Abstract

We present an algorithm for learning the intersection of half-spaces in n dimensions. Over nearly-uniform distributions it runs in polynomial time for up to $O(\log n/\log\log n)$ half-spaces, or more generally, for any number of half-spaces whose normal vectors lie in an $O(\log n/\log\log n)$ dimensional subspace. Over less restricted "non-concentrated" distributions it runs in polynomial time for a constant number of half-spaces. These results improve and generalize earlier work of Blum and Kannan [4] for the uniform distribution. The algorithm uses random projection in a novel way to identify a relevant low-dimensional subspace.

1 Introduction

In this paper, we consider the problem of learning the intersection of k half-spaces in n dimensions from labeled examples. For $k = 1$, this corresponds to learning a single half-space (also called a *perceptron*), one of the oldest problems in machine learning [9, 13]. Learning a single perceptron is equivalent to linear programming and hence can be solved in polynomial time. Other solutions to this problem, notably the perceptron algorithm, have also been studied in the literature [3]. While perceptrons themselves are highly interesting and in fact have directly found applications, it is often the case that one needs a more complex concept class to accurately model some phenomenon. The intersection of k half-spaces is a natural generalization of a perceptron. Besides its intuitive geometric appeal, in principle any convex concept could be approximated by an intersection of half-spaces. It also corresponds to a simple neural network which is used in many machine learning applications.

What is the complexity of learning the intersection of k half-spaces? There are several negative results about this [2, 5, 8, 12]. If we make no assumptions on the distribution from which examples are presented to us, the problem cannot be solved in polynomial time even for $k = 2$ unless $RP = NP$ [5, 12]. Further, if k is part of the input, then it is NP-hard under randomized reductions to learn the intersection k half-spaces using fewer than $kn^{1-\epsilon}$ half-spaces, for any $\epsilon > 0$.

On the other hand, for special distributions there are some positive results. Baum [1] gave an algorithm for learning the intersection of two homogenous half-spaces (a half-space is homogenous if the hyperplane defining it passes through the origin) over any distribution $\mathcal{D}$ that is origin-symmetric, i.e., for any $x \in R^n$, $\mathcal{D}(x) = \mathcal{D}(-x)$ (any point x and its reflection through the origin are equally likely). Recently, Blum and Kannan [4] gave a polynomial time algorithm for the problem for any constant number of half-spaces for the uniform distribution on the unit ball in n dimensions. Their algorithm does not explicitly find the half-spaces, instead it finds a prediction algorithm which can be evaluated in polynomial time (for a constant number of half-spaces) and is probably approximately correct. The running time and number of examples required by the algorithm are doubly exponential in k.

We present a randomized algorithm for the problem. Besides being simpler, our algorithm improves on the previous one in three ways:

- It is faster: the running time and number of examples required are (singly) exponential in k and polynomial in n. Hence we can learn the intersection of up to $O(\log n/\log\log n)$ half-spaces in polynomial time.
- The concept that it reports is shorter: our algorithm explicitly finds a set of $O(k)$ half-spaces.
- It can handle more general distributions (for a small number of half-spaces): specifically any distribution on the unit ball that is not "concentrated", i.e., whose probability density is at least $1/poly(n)$ and at most $poly(n)$ everywhere.

Although our algorithm is quite different from that of Blum and Kannan it is inspired by the same idea: the true complexity of the problem is determined *not* by the dimension n or the number of half-spaces k, but by the dimension of the subspace spanned by their normal vectors to the defining hyperplanes. Indeed our algorithm can learn the intersection of any number of half-spaces so long as their normals span a subspace of dimension $O(\log n/\log\log n)$.

To explain the idea, let us assume that the half-spaces are homogeneous, i.e., the hyperplanes defining them pass through the origin (this is without loss of generality as shown in Section 2.5). Let the half-spaces be $w_1 \cdot x \leq 0, w_2 \cdot x \leq 0, \ldots, w_k \cdot x \leq 0$. The intersection of these half-spaces is the positive region P. For each half-space $w_i \cdot x \leq 0$, w_i is the normal vector to the hyperplane defining the half-space (lying in the region opposite the positive region with respect to this hyperplane). Our goal will be to find a set of normal vectors that are very close in angle to $w_1, \ldots, w_k$. For this we consider the set of all normal vectors that define hyperplanes through the origin which *do not intersect* the positive region P. Formally, it is,

$$P^* = \{v \in \mathbf{R}^n \mid v \cdot x \leq 0 \,\forall x \in P\},$$

and is called the dual cone or *polar* cone of P [7] (see figure 1). It can be defined for any subset of $\mathbf{R}^n$ and for any closed cone C, it satisfies $(C^*)^* = C$. In our case, it follows that P^* is the the cone at the origin formed by the vectors

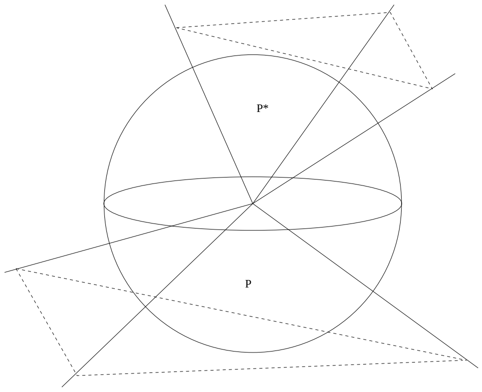

Figure 1: The polar cone

$w_1, \ldots, w_k$, i.e.,

$$P^* = \{v = \sum_{j=1}^{k} \alpha_j w_j \,|\, \forall j,\, \alpha_j \geq 0\}.$$

The first step of the algorithm is to find a good approximation to P^*. Although the minimum dimension of a subspace containing P^* is at most k (it could be less) we first find a good "approximation" to P^* in n dimensions. This is done by simply choosing a large sample of examples and considering the dual C^* of their conical hull C, i.e., the set of homogeneous hyperplanes that do not intersect the convex hull of the examples. In the second step we apply a simple procedure based on random sampling to identify a k-dimensional subspace close to the subspace containing P^*. Then we project our n-dimensional approximation of P^* to this relevant subspace. Let this projection be $\hat{C}^*$. The next step is to choose vectors from $\hat{C}^*$ to guarantee that for each w_i there is at least one vector in the sample close to it (in angle). We could do this by simply considering all points of a sufficiently fine grid enclosing $\hat{C}^*$. Finally, we prune the sample using a greedy heuristic. The half-spaces defined by the vectors in the pruned sample constitute the concept output by the algorithm. In other words, we label a point positive if it lies in the intersection of these half spaces and negative otherwise.

For most of the paper, we assume that the half-spaces we are trying to learn are homogeneous, i.e., the corresponding hyperplanes pass through the origin. In the next section, we introduce the framework and notation. Then we describe the algorithm in detail. Section 2.1 is devoted to proving a property we need of a large sample of examples. Section 2.2 outlines a proof of the sampling procedure used to find the relevant subspace. Section 2.4 discusses the final pruning step. Section 2.5 shows how to reduce the non-homogeneous case to the homogeneous one (this only works for the restricted distributions we can handle and not in general).

1.1 Preliminaries

Let us recall the terminology used in the literature. An *example* is a point in $\mathbf{R}^n$. A *concept* is a subset of $\mathbf{R}^n$. An example that belongs to a concept is a *positive* example for the concept, and an example that lies outside the concept is a *negative* example. Given a set of labelled examples drawn from an unknown distribution $\mathcal{D}$ in $\mathbf{R}^n$, and labelled according to an unknown *target* concept the learning task is to learn the target concept. What this means is that given an error parameter ϵ and a confidence parameter δ, with probability at least $1-\delta$ the algorithm has to find a concept that has error at most ϵ on $\mathcal{D}$.

For us the target concept is the intersection of l half-spaces in $\mathbf{R}^n$, such that the normal vectors to the hyperplanes bounding these half-spaces span a subspace of dimension k. For most of the paper we will assume that the hyperplanes bounding the half-spaces pass through the origin. Each point $x \in R^n$ is labeled *positive* or *negative* according to the following rule:

$$\begin{aligned} label(x) &= + \quad if \quad Wx \leq 0 \\ label(x) &= - \qquad otherwise. \end{aligned}$$

Here W is a real matrix of rank k with l rows and n columns. Each row w_i represents a half-space $w_i \cdot x \leq 0$. Hence a point x is labelled positive if it lies in the intersection of the k half-spaces and it is labeled negative otherwise. Formally, the positive region P is:

$$\{x \in \mathbf{R}^n | x \in B_n, Wx \leq 0\}$$

Examples presented to the algorithm are drawn from some unknown distribution on the unit ball in n dimensions, B_n. We assume that the distribution is *non-concentrated*, meaning that its probability density is at least $1/p$ and at most p everywhere in the unit ball. We also assume w.l.o.g. that P occupies at least a ϵ fraction of the unit ball (otherwise we could simply output the concept that labels every point negative).

Let P^* denote the dual cone of cone formed by the positive region.

$$P^* = \{v \in \mathbf{R}^n \,|\, v \cdot x \leq 0 \,\forall x \in P\} = \{v = \sum_{i=1}^{l} \alpha_i w_i, \,|\, \forall i,\, \alpha_i \geq 0\}.$$

Note that if we could project down to the subspace spanned by P^*, then the projections $w'_1, w'_2, \ldots, w'_l$ of $w_1, \ldots, w_k$ would give half-spaces that separate the positives from the negatives in the lower-dimensional space.

We say that a convex cone K in $\mathbf{R}^k$ is ϵ-*enclosed* by another convex cone K' if $K \subseteq K'$, and for every point $x \in K'$ there is some point $y \in K$ such that the angle between the vectors x and y is at most ϵ. Conversely, we say that K' ϵ-*encloses* K.

The *projection length* of a convex body K along a vector (or direction) v is the length of the 1-dimensional projection of K onto v. Intuitively, it is the width of the body in the direction v. Formally, it is

$$\max_{x \in K} x \cdot v - \min_{x \in K} x \cdot v.$$

1.2 The Algorithm

The input to the algorithm is an error parameter ϵ, a confidence parameter δ, and a set of labeled examples. The output of the algorithm is a set of m half-spaces.

Here we give a high-level description of the algorithm. Details and proofs of individual steps are in later sections. The parameters ϵ_i will be specified later.

1. **Approximate the dual cone.** Let S be the sample of positive examples presented to the algorithm. Let C be the cone formed by the vectors in S and C^* be the dual of C, i.e., the set of normal vectors to hyperplanes that do not intersect C.

$$C^* = \{y \in R^n \mid \forall x \in C, y \cdot x \leq 0\}.$$

   The cardinality of S will be chosen so that C^* ϵ_1-encloses P^*.

2. **Identify the "irrelevant" subspace.** We do this by finding a set of $n-k$ orthogonal vectors $\{x_1, x_2, \ldots, x_{n-k}\}$ as follows: Choose a set of random unit vectors and let x_1 be the one among them such that the projection length of $C^* \cap B_n$ in the direction of x_1 is minimum. Now pick unit vectors orthogonal to x_1 and let x_2 be the one among them with the minimum projection length of $C^* \cap B_n$. In this way at step i we find a vector x_i that is orthogonal to $x_1, \ldots, x_{i-1}$ and the projection length of C^* along x_i is small. After $n-k$ steps, the vectors $\{x_1, \ldots, x_{n-k}\}$ span a subspace that approximates the irrelevant subspace. The size of the sample is chosen so that each vector x_i is a small angle (ϵ_2) away from being orthogonal to P^*.

3. **Reduce dimensionality.** Compute the matrix M that projects to the subspace orthogonal to that spanned by $\{x_1, x_2, \ldots, x_{n-k}\}$, i.e., for any x in $\mathbf{R}^n$, Mx lies in the subspace spanned by C^*. Let $\hat{C}^*$ be the projection of C^*.

4. **Sample the dual.** Consider all lattice points spaced at ϵ_3 units in a box that encloses $\hat{C}^*$. In the original space, for each w_i there is a vector corresponding to some lattice point that makes an angle less than ϵ_3 with w_i.

5. **Prune the sample.** Let U be the set of lattice points from the previous step. Let S_1 be a new set of labelled examples. Greedily choose a subset of U of size m as follows: let u_1 be the vector from U such that $u_1 \cdot x \leq 0$ separates the largest number of negative examples from the positive examples in S_1. Discard the negative examples that are separated in this

way by u_1. Let u_2 be the vector from U that separates the largest number of remaining negative examples in S_1 from the positives. Similarly, let u_i be the vector from U that separates the largest number of remaining negatives. The number of steps m is chosen so that the number of remaining negatives after m steps is less than an $\epsilon/2$ fraction of the initial number. Output the vectors $u_1, u_2, \ldots, u_m$. Given an unlabeled point x, we label it positive if $u_i \cdot (Qx) \leq 0$ for $i = 1, \ldots, m$, and negative otherwise.

In the last step, observe that $u_i \cdot (Qx) = (u_i Q) \cdot x$ and so we can view $(u_i Q) \cdot x \leq 0$ as a half-space in $\mathbf{R}^n$.

2 It's analysis

Our main theorem is a performance guarantee for this algorithm for a suitable choice of the ϵ_i's. In the statement below $p \geq 1$ is a parameter of the distribution. If $\mathcal{D}$ has parameter p then the probability density everywhere in the unit ball is between $\frac{1}{p}$ and p.

Theorem 1 *The above algorithm PAC-learns with parameter ϵ and δ the intersection of l half-spaces whose normals lie in a k-dimensional subspace, and has a bound of*

$$poly(n) l \cdot (\frac{8pk \log n}{\epsilon})^{2k} \log \frac{1}{\epsilon\delta}$$

on the running time and the number of examples required.

For nearly-uniform distributions, i.e., when p is a constant, this gives us a polynomial-time algorithm for k up to about $\log n / \log\log n$.

2.1 A large sample of examples

In this section, we derive an upper bound on the number of examples required by the algorithm. Let S be the set of examples. Then $|S|$ should be large enough to guarantee that P^* is ϵ_1-enclosed by the dual to the cone formed by S.

Lemma 1 *Let $\mathcal{D}$ be a non-concentrated distribution on B_n with parameter p. Let S be a random sample of positive examples from this distribution and C be the cone at the origin formed by the points in S. Let C^* be the dual cone of C. If*

$$|S| \geq 16pnl(\frac{1}{\epsilon_1})^{k+1} \log \frac{1}{\epsilon\delta}),$$

then with probability at least $1-\delta$, the cone P^* is ϵ_1-enclosed by C^*.

Proof. Let Q^* be the maximal cone that ϵ_1-encloses P^*. Note that for each vector x in P^*, all the vectors within angle ϵ_1 from x are contained in Q^*. Let $Q = (Q^*)^*$ be the dual cone of Q^*.

We will prove that P^* is ϵ_1-enclosed by C^* by showing that the latter is contained in Q^*. For this, it is sufficient to show that for each point h on the boundary of Q^*, there is a supporting plane H of C^* which separates h from P^*. We will show that this occurs with high probability.

Since h is on the boundary of Q^*, by the definition of the dual cone, $h \cdot x = 0$ gives a supporting hyperplace of Q such that Q lies in the half-space $h \cdot x \leq 0$. Consider the convex region $P' = P \cap B_n \cap (h \cdot x \geq 0)$ and take $y \in P'$. Now $y \cdot h \geq 0$. On the other hand since the dual of P^* is P and $y \in P$, for any $w \in P^*$, we have $y \cdot w \leq 0$. Thus, $y \cdot x = 0$ would give us a hyperplane that separates h from P^*.

Next we bound the probability of choosing such a "witness" y in our sample S. This is at least $\frac{1}{p}$ times the volume of the unit ball occupied by P'. Now $vol(P') = vol(B_n \cap P \cap (h \cdot x \geq 0))$ and so we need to calculate the probability that y is chosen from the polyhedron defined by $w_i \cdot x \leq 0$ for $i = 1, 2, \ldots l$ and $h \cdot x \geq 0$. Thus we can focus only on the "relevant" $(k+1)$-dimensional subspace spanned by $w_1, w_2, \ldots, w_l$ and h.

Now observe that for any unit vector z in Q, all the vectors within angle ϵ_1 of z are contained in P. Suppose not, say there exists a vector z' on the boundary of P whose angle with z is smaller than ϵ_1. Then there is a supporting plane of P at z' with normal $h(z')$, so that upon moving $h(z')$ by an angle less than ϵ_1, the plane intersects Q, i.e., the new normal vector lies outside Q^*. This contradicts the maximality of Q^* — for any point in P^* (in our case, the normal vector $h(z')$), all points within angle ϵ_1 of it are contained in Q^*.

Next, note that $h \cdot x = 0$ is a supporting plane of Q, and let q be a point in Q for which $h \cdot q = 0$. Then, by the previous argument, the cone of all points within angle ϵ_1 of q is contained in P. Further, a half-cone of this angle is contained in P'. Thus, the volume of P' is $\Omega(\epsilon_1^{k+1}\mathsf{Vol}(P))$, and the probability that a single example is chosen from P' is $\Omega(\epsilon_1^{k+1}/p)$.

This proves it for a single point h. To prove it with high probability for every point in the boundary of Q^*, we use the VC-dimension of the intersection of up to $l+1$ half-spaces. The VC theorem implies that if we consider a sample of size

$$16\frac{ln}{\epsilon}\log\frac{1}{\epsilon\delta}),$$

then with high probability, every set of $l+1$ half-spaces whose intersection has probability mass at least ϵ will see at least one example in the sample. We set ϵ in the VC theorem to be ϵ_1^{k+1}/p to complete the proof. $\square$

2.2 Identifying the (ir)relevant subspace

In this section we analyze the procedure to approximately identify the irrelevant subspace and hence the subspace spanned by P^* .

The first step is to find a vector x_1 such that the projection of C^* along x_1 is small. For this we do the following. Pick a set of random unit vectors, and find the projection length of $C^* \cap B_n$ along these vectors. Let x_1 be the direction along which the projection length is minimum. In the next lemma we estimate the probability that x_1 is nearly orthogonal to P^*.

Fact 1 *The volume of the n-dimensional ball of radius r is equal to $\frac{2r^n \pi^{n/2}}{n\Gamma(n/2)}$ and its surface area is $\frac{2r^{n-1}\pi^{n/2}}{\Gamma(n/2)}$.*

Lemma 2 *Let $v_1, v_2, \ldots v_k$ be orthogonal unit vectors in $\mathbf{R}^n$. Then the probability that a random unit vector x satisfies $|v_i \cdot x| \le \beta$ for $i = 1, \ldots, k$ is*

$$(1 - k\beta^2)^{\frac{n-k}{2}} (\beta^2(n-k))^{k/2}$$

when x is chosen from a uniform distribution on the unit sphere.

Proof. Let $v_{k+1}, v_{k+2}, \ldots, v_n$ be unit vectors such that $v_1, v_2, \ldots, v_n$ form an orthonormal basis for $\mathbf{R}^n$. Write the $n-1$ dimensional sphere as $x_1^2 + \ldots + x_n^2 = 1$. Since we are choosing points uniformly, the desired probability is equal to the fraction of the sphere's surface area such that $|x_i| \le \beta$ for $i = 1, \ldots, k$. We bound the area of this region from below by setting $x_{k+1}^2 + \ldots + x_n^2$ equal to $1 - k\beta^2$ and letting the other coordinates vary between $-\beta$ and β. This describes a "band" which has a $n-k$ dimensional ball of radius $\sqrt{1 - k\beta^2}$ for its base and a thickness of 2β in the other k dimensions. Now denote the surface area of a $n+1$-dimensional sphere of radius r by $A_n(r)$. Then the desired probability is at least

$$\begin{aligned}
\frac{A_{n-k}(\sqrt{1-k\beta^2})(2\beta)^k}{A_n(1)} &= (\frac{4}{\pi})^{\frac{k}{2}} (1-k\beta^2)^{\frac{n-k-1}{2}} \beta^k \frac{\Gamma(\frac{n}{2})}{\Gamma(\frac{n-k}{2})} \\
&> (1-k\beta^2)^{\frac{n-k}{2}} \beta^k \frac{n^{n/2}}{(n-k)^{(n-k)/2}} \\
&> (1-k\beta^2)^{\frac{n-k}{2}} \beta^k \frac{(n-k)^{n/2}}{(n-k)^{(n-k)/2}} \\
&> (1-k\beta^2)^{\frac{n-k}{2}} \beta^k (n-k)^{k/2},
\end{aligned}$$

as desired. (Here we used Stirling's approximation to write $\Gamma(\frac{n}{2}) \sim \sqrt{\pi n}(\frac{n}{2e})^{n/2}$). □

Corollary 2 *If a sample of $\Omega(n/\alpha^k)$ random unit vectors is used then with high probability one of them will have a dot product of magnitude less than $\frac{\alpha}{\sqrt{n-k}}$ with each of a fixed set of k orthogonal unit vectors.*

Proof. Put $\beta = \frac{\alpha}{\sqrt{n-k}}$ into the expression given Lemma 2. If we choose $\lceil n/\alpha^k \rceil$ random unit vectors then

$$\begin{aligned}
\mathsf{P}[\exists x_i : |x_i \cdot v_j| < \frac{\alpha}{\sqrt{n-k}}, j = 1, 2, \ldots, k] &= 1 - (1 - \alpha^k)^{n/\alpha^k} \\
&\ge 1 - (e^{-\alpha^k})^{n/\alpha^k} \\
&= 1 - e^{-n}.
\end{aligned}$$

$\square$

In the algorithm, we set x_1 to be the vector in the sample that minimizes the projection length of $C^* \cap B_n$. Next, we sample only from vectors orthogonal to x_1, record the vector x_2 with the minimum projection length and repeat this to find $x_2, \ldots, x_{n-k}$.

For any $\alpha > 0$, let $P^*(\alpha)$ be the minimal subcone of P^* such that along any vector in the span of $P^*(\alpha)$, the projection length of $P^*(\alpha) \cap B_n$ is at least α. Note that the dimension of $P^*(\alpha)$ could be smaller than that of P^*, but is at least 1. Further, for any point x in P^*, there is a point in y in $P^*(\alpha)$ such that $||x - y|| \leq \alpha$.

Lemma 3 *Assume that C^* $\alpha/2$-encloses P^*. Let $v_1, \ldots, v_r$ be an orthonormal basis of $P^*(\alpha)$. Then using a sample of n/α^k random unit vectors at each step, we can find a set of vectors $x_1, \ldots, x_{n-k}$ that are almost orthogonal to span$(P^*(\alpha))$, i.e., $|v_i \cdot x_j| \leq \frac{\alpha}{\sqrt{n+1-j-r}}$ for $i = 1, \ldots, r$ and $j = 1, \ldots, n-k$. Furthermore any unit vector w in P^* has a projection $\hat{w}$ in $\hat{C}^*$ such that $w \cdot \hat{w} \geq 1 - \alpha^2 k \log n$.*

Proof. From Lemma 2, the vector x_1 satisfies $v_i \cdot x_1 \leq \frac{\alpha}{\sqrt{n-k}}$. For the purpose of analysis we can view the second step of the algorithm as first projecting C^* to the subspace orthogonal to x_1 and then sampling from all unit vectors in that subspace. The projection of v_i is $v_i - (v_i \cdot x_1)x_1$. So again from the lemma we have

$$|v_i \cdot x_2| = |(v_i - (v_i \cdot x_1)x_1) \cdot x_2| \leq \frac{\alpha}{\sqrt{n-1-k}}$$

At the t^{th} step, the projection of v_i is $v_i - \sum_{j=1}^{t-1}(v_i \cdot x_j)x_j$ and it follows that $|v_i \cdot x_j| \leq \frac{\alpha}{\sqrt{n+1-t-k}}$.

To prove the second part, for a unit vector $w \in P^*$, $\hat{w} = w - \sum_{j=1}^{n-r}(w \cdot x_j)x_j$. So $w \cdot \hat{w} = 1 - \sum_{j=1}^{n-r}(w \cdot x_j)^2$. We rewrite w as $\sum_{i=1}^{r} \lambda_i v_i + w'$ where $\sum_i \lambda_i^2 + ||w'||^2 = 1$ and $||w'|| < \alpha$. Then,

$$\begin{aligned}
|(w - w') \cdot x_j| &= |\sum_{i=1}^{r} \lambda_i (v_i \cdot x_j)| \\
&\leq \frac{\alpha}{\sqrt{n+1-j-r}} \sum_{i=1}^{r} \lambda_i \\
&\leq \alpha \sqrt{\frac{r(1 - ||w'||^2}{n+1-j-r}}.
\end{aligned}$$

Hence,

$$\begin{aligned} w \cdot \hat{w} &\geq 1 - \sum_{j=1}^{n-r} ((w - w') \cdot x_j)^2 - ||w'||^2 \\ &\geq 1 - r(1 - ||w'||^2) \sum_{j=1}^{n-r} \frac{1}{n+1-j-r}) - ||w'|^2 \\ &\geq 1 - \alpha^2 r \ln n. \end{aligned}$$

□

The assumption of the lemma can be satisfied by setting $\epsilon_1 = \alpha/2$ in the first step of the algorithm. By setting $\alpha = \frac{\epsilon_2}{\sqrt{2k \log n}}$ we get that each $w \in P^*$ has a projection $\hat{w} \in \hat{C}^*$ at an angle of no more than ϵ_2. Thus the number of random unit vectors used at each iteration is $n(2k \log n/\epsilon_2)^k$.

2.3 Sampling the dual

Now we need to find a sample in the projected dual so that there is a point in the sample within ϵ_3 of within each w_i. This can be done by simply finding a sample that has a point close to every point of $\hat{C}^* \cap B_k$. And it is easy to see that the size of our sample, which is just the intersection of $\hat{D}_c \cap B_k$ with the integer lattice scaled down by ϵ_3, is bounded by $(\frac{1}{\epsilon_3})^k$. Call this set of vectors U.

2.4 Pruning the sample of normal vectors

Here we show that the number of planes output by the algorithm is $m = O(l)$. Let S_1 be a fresh sample of examples. From standard VC-dimension arguments it is enough to find a set of half-spaces that have an error less than ϵ on a sample whose size is the VC-dimension. So we choose

$$|S_1| \geq \frac{16nl}{\epsilon} \log \frac{1}{\epsilon\delta}.$$

Let $z_1, z_2, \ldots, z_l$ be the vectors in U that are closest in angle to $w'_1, w'_2, \ldots, w'_l$ respectively (which are the projections of $w_1, \ldots, w_l$). Let P_Z be the region $z_1 \cdot x \leq 0, \ldots, z_l \cdot x \leq 0$. So P_Z is the positive region according to these vectors. Assume that the combined error of these half-spaces with respect to the actual set of half-spaces is bounded by $\epsilon/2$. In other words, there is a set of l half-spaces in U that correctly classify a $(1 - \frac{\epsilon}{2})$ fraction of the distribution. This can be achieved by setting $\epsilon_2 = \epsilon_3 \leq \frac{\epsilon}{4kp}$ in the previous steps of the algorithm.

Our procedure to prune U is the following: pick the best u from U, i.e., the vector u such that the half-space $u \cdot x \leq 0$ separates the maximum number of negatives from the positives in S_1. Call it u_1. Then pick the best u for the remaining negative examples and so on. From standard set cover guarantees [6], it follows that a greedy set of l half-spaces must separate at least half as many as the best set of l half-spaces, and a set of $l \log r$ greedy half-spaces separate at least $1 - \frac{1}{r}$ fraction of what the best set of l half-spaces can separate. We formalize this below.

Theorem 3 *With probability at least $1-\delta$, a greedily chosen set of $2l \log r$ half-spaces will correctly classify at least $(1-\frac{1}{r})(1-\frac{\epsilon}{2})$ fraction of the distribution.*

Note that the error probability in the statement above comes only from the possible error in there existing $z_1, \ldots, z_l$ that are close to the true vectors $w_1, \ldots, w_l$. The greedy pruning step itself is deterministic.

Setting r to be less than $\frac{\epsilon}{3}$ (say) gives us a set of $O(l)$ planes that correctly classify $1-\epsilon$ of the distribution with probability $1-\delta$.

Theorem 1 now follows from Lemmas 1,3 and Theorem 3.

2.5 The non-homogeneous case

The discussion so far has assumed that the half-spaces we are trying to learn are homogeneous. Of course this may not be the case in general, and here we show a simple reduction from the general (non-homogeneous) case to the homogeneous case by going to a representation in one more dimension, i.e., in $\mathbf{R}^{n+1}$. The key issue will be to make sure that we can do this while keeping the distribution in the new space non-concentrated. We make two observations for this.

Our first observation is that all the algorithm needs is a distribution on the unit *sphere* (rather than the ball) in $\mathbf{R}^n$ such that:

Any convex region of the sphere with more than $1/p(n)$ fraction of the area of the sphere should have probability mass between $1/q(n)$ and $q(n)$ for some polynomials $p(n), q(n)$ (analogously, any c_1 fraction should have probability mass between c_2 and $1/c_2$ for constants c_1, c_2).

Our second observation is that we only need the above condition to hold for the part of the unit sphere that is in the positive region. (Negative examples are used only in the last step.)

The following mapping from the unit ball B_n in $\mathbf{R}^n$ to the unit sphere S_n in $\mathbf{R}^{n+1}$ satisfies these conditions.

$$\begin{aligned} y_i &= \frac{n x_i}{\sqrt{n^2|x|^2+1}} \quad \mathit{for} \quad i = 1, \ldots n \\ y_{n+1} &= \frac{1}{\sqrt{n^2|x|^2+1}} \end{aligned}$$

This corresponds to blowing up B_n by a factor of n then placing it on the plane $y_{n+1} = 1$ and then mapping it stereographically to the unit sphere (scaling the length of the image). See figure 2 for an illustration.

The new coordinate y_{n+1} lies between 1 and $\frac{1}{\sqrt{n^2+1}}$, i.e., points are mapped only to the part of the sphere in the half-space $y_{n+1} \geq \frac{1}{\sqrt{n^2+1}}$. Suppose the

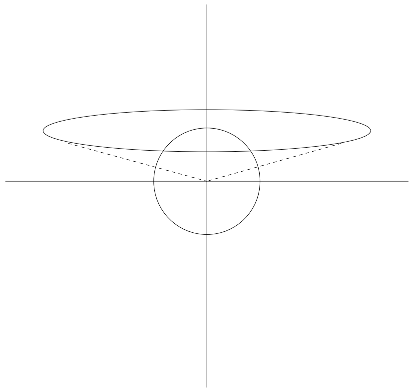

Figure 2: The mapping to $\mathbf{R}^{n+1}$

positive region in $\mathbf{R}^n$ is given by the set of (possibly non-homogeneous) half-spaces $w_i \cdot x \leq b_i$ for $i = 1, \ldots, l$. Then in $\mathbf{R}^{n+1}$ the positive region is given by the half-spaces

$$(w_i, -b_i n) \cdot y \leq 0$$

$$y_{n+1} \geq \frac{1}{\sqrt{n^2 + 1}}$$

This is then approximated by the homogeneous half-spaces $(w_i, -b_i n) \cdot y \leq 0$ and $y_{n+1} \geq 0$ and we can run the algorithm to learn them. The "blow-up" factor n could be replaced by any $poly(n)$. Only (about) half the sphere is used.

2.6 Open problems

Can the complexity of the algorithm be improved to polynomial in $\frac{1}{\epsilon}$? Indeed an algorithm that is fully polynomial, i.e., polynomial in k as well, might be possible for non-concentrated distributions.

One possibility for improvement is to use the SVD (singular value decomposition) in place of random projection to identify the relevant subspace.

Can we learn the intersection of k hyperplanes when the examples are drawn from the uniform distribution on the vertices of a hypercube? This does not induce a non-concentrated distribution on the sphere and so the methods in this paper cannot be applied directly. One reason that the problem is interesting for this distribution is that it includes as a special case the problem of learning DNF-formulae. Some progress has been made for the hypercube in [11] for the case when k is a constant using Fourier analysis.

References

[1] E. B. Baum. Polynomial time algorithms for learning neural nets. In *Proceedings of the Third Annual Workshop on Computational Learning Theory*, pages 258-272. Morgan Kaufmann (1990).

[2] Eric B. Baum. On learning a union of half spaces. *Journal of Complexity*, 6(1):67-101, March 1990.

[3] A. Blum, A. Frieze, R. Kannan and S. Vempala. A polynomial-time algorithm for learning noisy linear threshold functions. In *Proceedings of the 37th Symposium on the Foundations of Computer Science*, 330-338, 1996.

[4] A. Blum and R. Kannan. Learning an intersection of k halfspaces over a uniform distribution. In *Proceedings of the 34th Symposium on the Foundations of Computer Science*, 312-320, 1993.

[5] A. Blum and R. Rivest. Training a 3-node neural network is NP-hard. *Neural Networks*, 5:117-127, 1992.

[6] D. Hochbaum. *Approximation Algorithms for NP-Hard Problems*, Thomson, 1996.

[7] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*, Springer, 1988.

[8] P. M. Long and M. K. Warmuth. Composite geometric concepts and polynomial predictability. In *Proceedings of the Third Annual Workshop on Computational Learning Theory*, pages 273-287. Morgan Kaufmann (1990).

[9] Marvin Minsky and Seymour Papert. *Perceptrons: An Introduction to Computational Geometry.* The MIT Press, 1969.

[10] R. Kannan, L. Lovász and M. Simonovits. Random walks and and an $O^*(n^5)$ volume algorithm for convex sets. *Random Structures and Algorithms* **11** (1997), 1-50.

[11] A. Klivans, R. O'Donnell and R. Servedio. Learning intersections and thresholds of half-spaces. *Proc. of the 43rd Symposium on the Foundations of Computer Science*, 177-186, 2002.

[12] N. Megiddo. On the complexity of polyhedral separability. Technical Report RJ 5252, IBM Almaden Research Center, August 1996.

[13] F. Rosenblatt. *Principles of Neurodynamics.* Spartan Books, 1962.