An Algorithmic Theory of Learning: Robust Concepts and Random Projection*

Rosa I. Arriaga† Santosh Vempala ‡

Abstract

We study the phenomenon of cognitive learning from an algorithmic standpoint. How does the brain effectively learn concepts from a small number of examples despite the fact that each example contains a huge amount of information? We provide a novel algorithmic analysis via a model of *robust* concept learning (closely related to "margin classifiers"), and show that a relatively small number of examples are sufficient to learn rich concept classes. The new algorithms have several advantages — they are faster, conceptually simpler, and resistant to low levels of noise. For example, a robust half-space can be PAC-learned in linear time using only a constant number of training examples, regardless of the number of attributes. A general (algorithmic) consequence of the model, that "more robust concepts are easier to learn", is supported by a multitude of psychological studies.

1 Introduction

A principal motivation of computational learning theory is to gather insight into cognitive processes. While the exact physical processes underlying learning, indeed any aspect of cognition, are far from being understood, it is also mostly a mystery as to how the brain copes with huge amounts of data, even from a purely theoretical standpoint. How does the brain effectively learn concepts from a relatively small number of examples, when each example consists of a huge amount of information?

There are at least two approaches to explaining this phenomenon. The first, due to Valiant, is *attribute-efficient learning* [41, 31, 32]. In this model, it is assumed that the target concept is simple in a specific manner: it is a function of only a small subset of the set of attributes, called the *relevant* attributes, while the rest are *irrelevant*. From this assumption one can typically argue that the VC-dimension of the resulting concept class is a function of only the number of relevant attributes (k), and hence derive a bound on the number of examples required. Unfortunately, although the model is theoretically clean and appealing, it is not known how to learn anything more complex than a disjunction of variables (without membership queries). Further, it is NP-hard to learn a disjunction of k variables as a disjunction of fewer than $k \log n$ variables (where n is the total number of variables).

In this paper, we study a different approach based on a simple idea which is illustrated in the following example. Imagine a child learning the concept of an "elephant". We point the child to pictures of elephants or to real elephants a few times and say "elephant", and perhaps to a few examples of other animals and say their names (i.e. "*not* elephant"). ¿From then on, the child will almost surely correctly label only elephants as elephants. On the other hand, imagine a child learning the concept of "African elephant" (as opposed to the Indian elephant)

*A preliminary version of this paper appeared in the Proc. of the Symposium on the Foundations of Computer Science, 1999

†Department of Psychology, Harvard University. Email: rarriaga@wjh.harvard.edu

‡Department of Mathematics, M.I.T. Email: vempala@math.mit.edu.

just from examples. It will probably take many more examples, and perhaps even be necessary to explicitly point out the bigger ears of the African elephant.

The crucial difference in the two concepts above is not in the number of attributes, or even in the number of relevant attributes of the examples presented, but in the *robustness* of the concepts. There is a clearer demarcation between elephants and non-elephants than there is between African elephants and Indian elephants. This is the basis of the robust concept model. An alternative perspective of robustness is that it is a measure of how much the attributes of an example can be altered without affecting the concept. The robustness of a concept is measured by a parameter ℓ. The main feature of robust concepts is that the number of examples and the time required to learn a robust concept can be bounded as a function of ℓ and do not depend on the total number of attributes. The model and the paramter ℓ are defined precisely in section 2. As we discuss there, the model is very closely related to *Large Margin* classifiers studied in machine learning, that are in turn the basis for Support Vector Machines [42, 13].

In the robust concept model, the main new observation is that we can employ a general procedure to reduce the dimensionality of examples, *independent of the concept class.* While reducing the dimensionality of examples, we would like to preserve concepts. So, for example, if our original concept class is the set of half-spaces (linear thresholds) in n-dimensional space, we would like to project examples to a k-dimensional space, where $k << n$, and maintain the property that some half-space in the k-dimensional space correctly classifies (most of) the examples. We show that *Random Projection*, the technique of projecting a set of points to a randomly chosen low-dimensional space, is suitable for this purpose. It has been observed that random projection (approximately) preserves key properties of a set of points, e.g., the distances between pairs of points [24]; this has led to efficient algorithms in several other contexts [28, 30, 44, 45]. In section 3, we develop "neuronal" versions of random projection, i.e., we demonstrate that it is easy to implement it using a single layer of perceptrons where the weights of the network are chosen *independently* and from any one of a class of distributions; this class includes discrete distributions such as the picking 1 or -1 with equal probability. Our theorems here can be viewed as an extension/refinement of the work of [24, 16].

Then we address the question of how many examples are needed to efficiently learn a concept with robustness ℓ. We begin with the concept class of half-spaces with n attributes. In this case, it is already known that one needs $O(1/\ell^2)$ examples [11, 42, 17]. Here we show that a simple algorithm based on random projection gives an alternative proof of such a guarantee.

Next we consider other rich concept classes, namely intersections of half-spaces and ellipsoids. Using neuronal random projection, we demonstrate that the examples can first be projected down to a space whose dimension is a function of ℓ, and in some cases an additional parameter of the concept class (e.g. the number of half-spaces when the concept class is intersections of half-spaces etc.), but does not depend on the number of attributes of the examples. This then allows us to bound the number of examples required to learn the concepts as a function of ℓ, independent of the original number of attributes, via an adaptation of the fundamental VC-dimension theorem [43].

The proposed algorithms are fast — their running time is linear in n — since after random projection (which takes time linear in n), all the work happens in the smaller-dimensional space with a small number of sample points. Indeed, this suggests that the algorithms studied here could be used in SVM's in place of current solutions [13, 17] such as quadratic optimization in a dual space called the kernel space.

In section 5.2, we mention the noise tolerance properties of the algorithms, notably that agnostic learning is possible, and (equivalently) that it is possible to find hypotheses that *minimize the number of misclassified points*, for fairly low robustness.

1.1 Related work

The main contribution of this paper is a new perspective on learning via a connection to dimension reduction. This facilitates efficient algorithms which use small sample sizes. It also gives a simple intuitive way to see the $O(1/\epsilon^2)$ sample complexity bounds of margin classifiers (SVM's) [11]. It is related to previous work [39] which showed that generalization error can be bounded in terms of the observed margin of examples (a more refined notion of margin is used there, but is similar in spirit). As we discuss in section 5.1, it seems to fit well with attempts to model cognition on a computational basis [41], and predicts the commonly observed phenomenon that finer distinctions take more examples. From a purely computational viewpoint, these are simple new algorithms for fundamental learning theory problems, that might be practical.

There have been further applications of random projection in learning theory subsequent to this work. Garg, Har-Peled and Roth [18] and Garg and Roth [19] have pursued similar ideas, developing the related notion of *projection profile*. Recently, Balcan, Blum and Vempala [4] have used random projection to give a new interpretation of Kernel functions. Klivans and Servedio [27] have used polynomial threshold functions in the context of robust concepts to get substantially improved time bounds. Specifically, they give faster algorithms for learning intersections (and other functions) of t half-spaces (with some increase in the sample complexity). Finally, Ben-David, Eiron and Simon [6] have used random projection to show a interesting lower bound on learning with half-spaces. They prove that "most" concept classes of even constant VC-dimension cannot be embedded into half-spaces where the dimension of the Euclidean space is small or the margin is large. Thus algorithms based on first transforming to half-spaces cannot gain much in terms of the margin or the dimension.

2 The model

To describe the model, we adopt the terminology used in the literature. We assume that attributes are real valued; an *example* is a point in $\boldsymbol{R}^n$; a *concept* is a subset of $\boldsymbol{R}^n$. An example that belongs to a concept is labelled *positive* for the concept, and an example that lies outside the concept is labelled a *negative* example.

Given a set of labelled examples drawn from an unknown distribution $\mathcal{D}$ in $\boldsymbol{R}^n$, and labelled according to an unknown *target* concept the learning task is to find a hypothesis with low error. A *hypothesis* is a polynomial-time computable function. The error of a hypothesis h with respect to the target concept is the probability that h disagrees with the target function on a random example drawn from $\mathcal{D}$. Thus, if h has error ϵ, then the probability for a random x that $h(x)$ disagrees with the target concept is at most ϵ. So, 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 $\mathcal{D}$ [40].

The basic insight of the new model is the idea of robustness (implicit in earlier work). Intuitively, a concept is "robust" if it is immune to attribute noise. That is, modifying the attributes of an example by some bounded amount does not change its label. Another interpretation is that points with different labels are far apart. This is formalized below:

Definition 1 *For any real number $\ell > 0$, a concept C in conjunction with a distribution $\mathcal{D}$ in $\boldsymbol{R}^n$, is said to be* ℓ-robust, *if*

$$\mathsf{P}_{\mathcal{D}}\left(x \,|\, \exists y : label(x) \neq label(y), ||x-y|| \leq \ell\right) = 0$$

The norm $||x-y||$ is the Euclidean distance between x and y. This can be replaced by other norms, but we use the Euclidean norm in this paper. The probability is over all points x with the property that there is some point y with a different label within a distance ℓ. In other words, a concept is ℓ-robust if there is zero probability of points being within ℓ of the boundary of the concept. The definition could be weakened by requiring only that the above probability should be negligible (e.g. $1/2^n$). When $\mathcal{D}$ is over a discrete subset of $\boldsymbol{R}^n$, then this has a simple interpretation. A ball of radius ℓ around any point x of non-zero probability lies entirely on one side of the concept, i.e., every point in the ball has the same label as x. To avoid scaling issues, we usually consider only distributions whose support is (a subset of) the unit ball in $\boldsymbol{R}^n$, i.e., all examples given to the algorithm will have length at most 1 (alternatively, one could incorporate normalize the distance between examples by their length, but we find our definition more convenient). Given access to examples from a robust concept, and parameters ϵ, δ, a learning algorithm succeeds and is said to (ϵ, δ)-learn if, with probability at least $1 - \delta$, it produces a hypothesis that is consistent with at least $1 - \epsilon$ of the example distribution. Note that strictly speaking this is not PAC-learning since robustness restricts the example distribution.

In what follows, we present tools and algorithms for learning robust concepts. It is worth noting that "robustness" refers only to the target concept; it is not required of all concepts in the class.

2.1 Connection to existing models

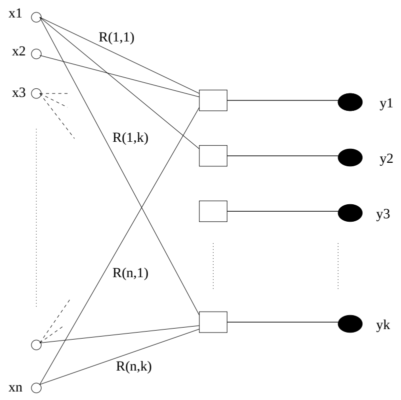
The model is closely related to large margin classifiers used in Support Vector Machines [11]. Indeed, for the concept class of half-spaces, the robustness as defined here is exactly the largest possible margin of a correctly classifying half-space (with the normalization that all the examples are from the unit ball). In general, however, there is a subtle but important difference. Whereas in SVM's the margin is measured in the "lifted" space where concepts have been transformed to half-spaces, in our model we measure robustness in the space in which examples are presented to us (and hence the natural relationship with attribute noise). The robustness is also closely related to the parameter γ used in the definition of the *fat-shattering dimension* [25, 11], and once again coincides (up to a scaling factor) in the case of half-spaces.

3 The main tool: neuron-friendly random projection

In this section we develop "neuronal" versions of random projection, including a discrete version, and provide probabilistic guarantees for them, all with transparent proofs. Besides being neuron-friendly, these versions of random projection are easier to implement.

To project a given point $u \in \boldsymbol{R}^n$ to a k-dimensional space, we first choose k random vectors $R_1, \ldots, R_k$ (we will shortly discuss suitable probability distributions for these vectors). Then we compute a k-dimensional vector u' whose coordinates are the inner products $u'_1 = R_1^T \cdot u, \ldots, u'_k = R_k^T \cdot u$. If we let R be the $n \times k$ matrix whose columns are the vectors $R_1, \ldots, R_k$, then the projection can be succinctly written as $u' = R^T u$. To project a set of points $u^1, \ldots, u^m$ in $\boldsymbol{R}^n$ to $\boldsymbol{R}^k$, we choose a random matrix R as above, and compute the vectors $R^T u^1, \ldots, R^T u^m$.

Given the matrix R, the above procedure is a simple computational task. It has been shown that if R is a random *orthonormal* matrix, i.e., the columns of R are random unit vectors and they are pairwise orthogonal, then the projection preserves all pairwise distances to within a factor of $(1 + \epsilon)$ for a surprisingly small value of k of about $\log n/\epsilon^2$ [24]. The main observation of this section is to show that this is a rather robust phenomenon, in that the entries of R can be chosen from any distribution with bounded moments. In particular it suffices to use random matrices with *independent* entries chosen from a distribution with bounded support. It is then an easy consequence that the task of random projection can be achieved by a simple 1-layer neural network, viz., k perceptrons (which compute linear combinations of their inputs) each with one output and the same n inputs. The weights of the neural network are assumed to be random and independent. This is illustrated in figure 1. Let $r \in \boldsymbol{R}^n$ be a random vector whose coordinates are independent and identically distributed. We highlight the following two possibilities for the distribution of the coordinates: (a) the standard normal distribution, with mean 0 and variance



Neuronal Random Projection

1, referred to as $N(0,1)$, (b) the discrete distribution defined by $r_i = 1$ with probability $\frac{1}{2}$ and $r_i = -1$ with probability $\frac{1}{2}$, which we will refer to as $U(-1,1)$. Following the conference version of this paper [3], another proof for the case $U(-1,1)$ has also appeared in [1]. The following well-known lemma will be useful. We provide a proof for convenience.

Lemma 1 *Let X be drawn from $N(0,\sigma)$, the normal distribution with mean zero and standard deviation σ. Then for any $\alpha < \frac{1}{2\sigma^2}$,*

$$\mathsf{E}(e^{\alpha X^2}) = \frac{1}{\sqrt{1-2\alpha\sigma^2}}.$$

Proof. We recall the density function of $N(0,\sigma)$, the normal distribution with mean 0 and standard deviation σ, to be

$$\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}.$$

Using this,

$$\begin{aligned}
\mathsf{E}(e^{\alpha X^2}) &= \int_{-\infty}^{\infty} e^{\alpha x^2} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}\, dx \\
&= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}(1-2\alpha\sigma^2)}\, dx \\
&= \frac{1}{\sqrt{1-2\alpha\sigma^2}} \int_{-\infty}^{\infty} \frac{\sqrt{1-2\alpha\sigma^2}}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}(1-2\alpha\sigma^2)}\, dx \\
&= \frac{1}{\sqrt{1-2\alpha\sigma^2}}.
\end{aligned}$$

Here we have used the observation that the integrand is the normal density with standard deviation $\sigma/\sqrt{1-2\alpha\sigma^2}$. □

We begin with the case when each entry of the projection matrix is chosen independently from the standard Normal distribution.

Lemma 2 *Let $R = (r_{ij})$ be a random $n \times k$ matrix, such that each entry r_{ij} is chosen independently according to $N(0,1)$. For any vector fixed $u \in \boldsymbol{R}^n$, and any $\epsilon > 0$, let $u' = \frac{1}{\sqrt{k}}(R^T u)$. Then, $\mathsf{E}(||u'||^2) = ||u||^2$ and*

$$Pr\left[||u'||^2 > (1+\epsilon)||u||^2\right] \le ((1+\epsilon)e^{-\epsilon})^k \le e^{-(\epsilon^2-\epsilon^3)\frac{k}{4}}$$

$$Pr\left[||u'||^2 < (1-\epsilon)||u||^2\right] \le ((1-\epsilon)e^{\epsilon})^k \le e^{-(\epsilon^2-\epsilon^3)\frac{k}{4}}.$$

Proof. The expectation follows from a simple calculation. To obtain the bound on the concentration near the mean, let $X_j = (R_j^T \cdot u)/||u||$ and observe that

$$X = \sum_{j=1}^{k} X_j^2 = \sum_{j=1}^{k} \frac{(R_j^T \cdot u)^2}{||u||^2}$$

where R_j denotes the jth column of R. Each X_j has the standard normal distribution (since each component of R_j does). Also note that

$$||u'||^2 = \frac{||u||^2}{k} X.$$

Using Markov's inequality, we can then estimate the desired probability as

$$\begin{aligned}
\mathsf{P}(||u'||^2 \geq (1+\epsilon)||u||^2) = Pr(X \geq (1+\epsilon)k) &= Pr(e^{\alpha X} \geq e^{(1+\epsilon)k\alpha}) \\
&\leq \frac{\mathsf{E}(e^{\alpha X})}{e^{(1+\epsilon)k\alpha}} \\
&= \frac{\Pi_{j=1}^{k} \mathsf{E}(e^{\alpha X_j^2})}{e^{(1+\epsilon)k\alpha}} = \left(\frac{\mathsf{E}(e^{\alpha X_1^2})}{e^{(1+\epsilon)\alpha}} \right)^k .
\end{aligned}$$

In the last line above, we have used the independence of the X_j's. Similarly,

$$\mathsf{P}(||u'||^2 \leq (1-\epsilon)||u||^2) \leq \left(\frac{\mathsf{E}(e^{-\alpha X_1^2})}{e^{-(1-\epsilon)\alpha}} \right)^k .$$

¿From Lemma 1,

$$\mathsf{E}(e^{\alpha X_1^2}) = \frac{1}{\sqrt{1-2\alpha}}$$

for any $\alpha < \frac{1}{2}$. Thus we get,

$$Pr(X \geq (1+\epsilon)k) \leq \left(\frac{e^{-2(1+\epsilon)\alpha}}{(1-2\alpha)} \right)^{\frac{k}{2}} .$$

The optimal choice of α is $\epsilon/2(1+\epsilon)$. With this,

$$Pr(X \geq (1+\epsilon)k) \leq \left((1+\epsilon)e^{-\epsilon}\right)^{\frac{k}{2}} \leq e^{-(\epsilon^2-\epsilon^3)\frac{k}{4}} .$$

Similarly,

$$Pr(X \leq (1-\epsilon)k) \leq \left(\frac{e^{2(1-\epsilon)\alpha}}{(1+2\alpha)} \right)^{\frac{k}{2}} \leq ((1-\epsilon)e^{\epsilon})^{\frac{k}{2}} \leq e^{-(\epsilon^2-\epsilon^3)\frac{k}{4}} .$$

$\square$

The main theorem of this section shows that this phenomenon is not specific to the Normal distribution. In the statement below, the condition that $\mathsf{E}(r^2) = 1$ is for convenience. Instead one could have an arbitrary finite value σ^2 for this expectation, and scale the projection by σ.

Theorem 1 *Let R be a random $n \times k$ matrix, with each entry r chosen independently from a distribution $\mathcal{D}$ that is symmetric about the origin with $\mathsf{E}(r^2) = 1$. For any fixed vector $u \in \boldsymbol{R}^n$, let $u' = \frac{1}{\sqrt{k}} R^T u$.*

1. *Suppose* $B = \mathsf{E}(r^4) < \infty$. *Then for any* $\epsilon > 0$,

$$\mathsf{P}\left([||u'||^2 \le (1-\epsilon)||u||^2)\right] \le e^{-\frac{(\epsilon^2-\epsilon^3)k}{2(B+1)}}.$$

2. *Suppose* $\exists L > 0$ *such that for any integer* $m > 0$, $\mathsf{E}(r^{2m}) \le \frac{(2m)!}{2^m m!} L^{2m}$. *Then for any* $\epsilon > 0$,

$$\mathsf{P}\left(||u'||^2 \ge (1+\epsilon)L^2||u||^2\right) \le \left((1+\epsilon)e^{-\epsilon}\right)^{k/2} \le e^{-(\epsilon^2-\epsilon^3)\frac{k}{4}}.$$

Proof. Without loss of generality, assume that $||u||^2 = 1$. Let

$$X_i = R_i^T u \text{ for } i = 1, \ldots, k.$$

We have

$$\mathsf{E}(X_i^2) = \mathsf{E}((R_i^T u)^2) = \mathsf{E}((\sum_{j=1}^n R_{ij}u_j)^2) = \sum_{j=1}^n \mathsf{E}(R_{ij}^2)u_j^2 = 1.$$

Then, if we define Y as follows

$$Y := \sum_{i=1}^k X_i^2 = k||u'||^2, \qquad \mathsf{E}(Y) = \sum_{i=1}^k \mathsf{E}(X_i^2) = k.$$

The deviation below the mean is relatively easy to bound, using the independence of the X_i's and Markov's inequality.

$$\begin{aligned}
\mathsf{P}\left(||u'||^2 < (1-\epsilon)||u||^2\right) &= \mathsf{P}(Y < (1-\epsilon)k) \\
&= \mathsf{P}\left(e^{-\alpha Y} > e^{-\alpha(1-\epsilon)k}\right) \\
&\le \frac{\mathsf{E}(e^{-\alpha Y})}{e^{-\alpha(1-\epsilon)k}} \\
&= \left(\mathsf{E}(e^{-\alpha X_1^2})e^{\alpha(1-\epsilon)}\right)^k
\end{aligned}$$

and, using that $e^{-\alpha X_1^2} \le 1 - \alpha X_1^2 + \alpha^2 X_1^4/2$, we get

$$\mathsf{P}\left(||u'||^2 < (1-\epsilon)||u||^2\right) \le \left((1 - \alpha\mathsf{E}(X_1^2) + \frac{\alpha^2}{2}\mathsf{E}(X_1^4))e^{\alpha(1-\epsilon)}\right)^k.$$

We can evaluate the moments easily: $\mathsf{E}(X_1^2) = 1$ and, if we observe that the expectation of odd powers of r is zero because of symmetry, we have (using the fact that $B \ge 1$),

$$\begin{aligned}
\mathsf{E}(X_1^4) &= \mathsf{E}((\sum_{j=1}^n R_{1j}u_j)^4) \\
&= \sum_{j_1,j_2,j_3,j_4=1}^n \mathsf{E}(R_{1j_1}R_{1j_2}R_{1j_3}R_{1j_4})u_{j_1}u_{j_2}u_{j_3}u_{j_4} \\
&= \sum_{j=1}^n \mathsf{E}(R_{1j}^4)u_j^4 + 3\sum_{j_1 \ne j_2, j_1, j_2=1}^n \mathsf{E}(R_{1j_1}^2 R_{1j_2}^2)u_{j_1}^2 u_{j_2}^2 \\
&\le B\sum_{j=1}^n u_j^4 + 3\sum_{j_1 \ne j_2, j_1, j_2=1}^n u_{j_1}^2 u_{j_2}^2 \\
&\le (B+2)(\sum_j u_j^2)^2 \\
&= B+2.
\end{aligned}$$

Therefore, using the Taylor expansion of e^x,

$$
\begin{aligned}
\mathsf{P}\left(||u'||^2 < (1-\epsilon)||u||^2\right) &\leq \left(\left(1-\alpha+\frac{\alpha^2}{2}(B+2)\right)e^{\alpha(1-\epsilon)}\right)^k \\
&\leq \left(e^{-\alpha+\frac{\alpha^2(B+2)}{2}-\frac{1}{2}(\alpha-\frac{\alpha^2(B+2)}{2})^2}e^{\alpha(1-\epsilon)}\right)^k \\
&\leq e^{-\frac{(\epsilon^2-\epsilon^3)k}{2(B+1)}}.
\end{aligned}
$$

The last line above is obtained by setting $\alpha = \epsilon/(B+1)$ and noting that $B \geq 1$.

Similarly, for the deviation above the mean,

$$
\mathsf{P}\left(||u'||^2 > (1+\epsilon)L^2||u||^2\right) \leq \left(\frac{\mathsf{E}(e^{\alpha X_1^2})}{e^{\alpha L^2(1+\epsilon)}}\right)^k.
$$

The main task is bounding $\mathsf{E}(e^{\alpha X_1^2})$ from above using the assumptions of the theorem. This expectation is hard to evaluate directly since we don't know the distribution explicitly. However we have bounds on all the moments of X_1^2. Therefore, if we define a random variable Z whose moments are all at least the moments of X_1^2, then $\mathsf{E}(e^{\alpha Z})$ will be an upper bound on the required expectation. The following claim will be useful.

Claim 1 *Let f, g be distributions on $\boldsymbol{R}$ that are symmetric about the origin with the property that for any nonnegative integer m, $\mathsf{E}(Y^{2m}) \leq \mathsf{E}(Z^{2m})$ where Y, Z are drawn from f, g respectively. Let $Y_1, \ldots, Y_n$ be i.i.d. from f, $Z_1, \ldots, Z_n$ be i.i.d from g. Then for any $u \in \boldsymbol{R}^n$, the random variables $\hat{Y} = \sum_{j=1}^n u_j Y_j$ and $\hat{Z} = \sum_{j=1}^n u_j Z_j$ satisfy $\mathsf{E}((\hat{Y})^{2m}) \leq \mathsf{E}((\hat{Z})^{2m})$ for every nonnegative integer m.*

The claim is easy to prove. Compare the expectations of individual terms of $(\hat{Y})^{2m}$ and $(\hat{Z})^{2m}$. Since Y_i, Z_i are symmetric about the origin, all terms in which they appear with an odd power have an expectation of zero. For any term in which all powers are even, by the assumption, the term from $\mathsf{E}((\hat{Z})^{2m})$ dominates.

To apply this to our setting, we know that

$$
X_1 = \sum_{j=1}^n u_j r_j
$$

where each r_j is drawn from the given distribution $\mathcal{D}$. Define

$$
Y_1 = \sum_{j=1}^n u_j r'_j
$$

where each r'_j is drawn from $N(0, L)$. Then for all j, and any integer $m > 0$,

$$
\mathsf{E}(r_j^{2m}) \leq \frac{(2m)!}{2^m m!}L^{2m} = \mathsf{E}\left((r'_j)^{2m}\right)
$$

using the well-known formula for the moments of $N(0, L)$. So, $\mathsf{E}(X_1^{2m}) \leq \mathsf{E}(Y_1^{2m})$. Moreover, the distribution of Y_1 is $N(0, L)$. Therefore,

$$
\mathsf{E}(e^{\alpha X_1^2}) \leq \mathsf{E}(e^{\alpha Y_1^2}) = \frac{1}{\sqrt{1-2\alpha L^2}}.
$$

Using this,

$$\mathsf{P}\left(||u'||^2 > (1+\epsilon)L^2||u||^2\right) \quad \leq \quad \left(\frac{e^{-2\alpha L^2(1+\epsilon)}}{1-2\alpha L^2}\right)^{\frac{k}{2}}.$$

The optimal choice of α is $\epsilon/2L^2(1+\epsilon)$, and we get that for any $\epsilon > 0$,

$$\mathsf{P}\left(||u'||^2 > (1+\epsilon)L^2||u||^2\right) \leq \left((1+\epsilon)e^{-\epsilon}\right)^{\frac{k}{2}} \leq e^{-(\epsilon^2-\epsilon^3)\frac{k}{4}}.$$

The last inequality was obtained by using the inequality $\ln(1+\epsilon) \leq \epsilon - \epsilon^2/2 + \epsilon^3/2$.

□

Corollary 1 *If every entry of an $n \times k$ matrix R is chosen according to $U(-1,1)$, then for any fixed vector $u \in \boldsymbol{R}^n$ and any $\epsilon > 0$, the vector $u' = \frac{1}{\sqrt{k}}R^T u$ satisfies*

$$\mathsf{P}\left(||u'||^2 \geq (1+\epsilon)||u||^2\right) \leq e^{-(\epsilon^2-\epsilon^3)\frac{k}{4}} \text{ and } \mathsf{P}\left(||u'||^2 \leq (1-\epsilon)||u||^2\right) \leq e^{-(\epsilon^2-\epsilon^3)\frac{k}{4}}.$$

Proof. For r drawn from $U(-1,1)$, $\mathsf{E}(r^{2m}) = 1$ for any integer $m > 0$. Therefore, we can apply Theorem 1 with $L = B = 1$ to get the conclusion of the corollary. □

Let R be an $n \times k$ matrix whose entries are chosen independently from either $N(0,1)$ or $U(-1,1)$, independently. The following theorem summarizes the results of this section. Alternative proofs for the case of $N(0,1)$ appeared in [23, 14].

Theorem 2 (Neuronal RP) *Let $u, v \in \boldsymbol{R}^n$. Let u' and v' be the projections of u and v to $\boldsymbol{R}^k$ via a random matrix R whose entries are chosen independently from either $N(0,1)$ or $U(-1,1)$. Then,*

$$\mathsf{P}\left[(1-\epsilon)||u-v||^2 \leq ||u'-v'||^2 \leq (1+\epsilon)||u-v||^2\right] \geq 1 - 2e^{-(\epsilon^2-\epsilon^3)\frac{k}{4}}.$$

Proof. Apply Theorem 1 to the vector $u - v$. □

We conclude this section with a useful corollary. A similar proof can be found in [6].

Corollary 2 *Let u, v be vectors in $\boldsymbol{R}^n$ s.t. $||u||, ||v|| \leq 1$. Let R be a random matrix whose entries are chosen independently from either $N(0,1)$ or $U(-1,1)$. Define $u' = \frac{1}{\sqrt{k}}R^T u$ and $v' = \frac{1}{\sqrt{k}}R^T v$. Then for any $\epsilon > 0$,*

$$\mathsf{P}\left(u \cdot v - c \leq u' \cdot v' \leq u \cdot v + c\right) \geq 1 - 4e^{-(\epsilon^2-\epsilon^3)\frac{k}{4}}.$$

Proof. Applying Theorem 2 to the vectors u, v and $u - v$, we have that with probability at least $1 - 4e^{-(c^2-c^3)\frac{k}{4}}$,

$$\begin{aligned} (1-c)||u-v||^2 \leq \quad & ||u'-v'||^2 \quad \leq (1+c)||u-v||^2 \\ \text{and } (1-c)||u+v||^2 \leq \quad & ||u'+v'||^2 \quad \leq (1+c)||u+v||^2. \end{aligned}$$

Then,

$$\begin{aligned} 4u' \cdot v' &= ||u'+v'||^2 - ||u'-v'||^2 \\ &\geq (1-c)||u+v||^2 - (1+c)||u-v||^2 \\ &= 4u \cdot v - 2c(||u||^2 + ||v||^2) \\ &\geq 4u \cdot v - 4c. \end{aligned}$$

Thus $u' \cdot v' \geq u \cdot v - c$. The other inequality is similar. □

In what follows, we will apply random projection by picking entries of the projection matrix independently from $N(0,1)$ or $U(-1,1)$. We remark that one could use other distributions via Theorem 1.

4 Learning efficiently by reducing dimensionality

In this section, we describe learning algorithms for robust concepts and derive bounds on the number of examples required and the running times. Our bounds will be functions of the robustness parameter l, and the ϵ, δ learning parameters, but will be independent of the actual number of attributes of the concept class.

We are given labelled examples from an unknown distribution $\mathcal{D}$. The generic algorithm for learning robust concepts is based on two high-level ideas.

1. Since the target concept is robust, *random projection* of the examples to a much lower-dimensional subspace will "preserve" the concept.

2. In the lower-dimensional space, the number of examples and the time required to learn concepts are relatively small.

Before applying this approach to specific concept classes, we recall some fundamental theorems in learning theory. For the concept class $\mathcal{C}$ under consideration, let $C(m, k)$ denote the maximum number of distinct labellings of m points that can be obtained by using concepts from $\mathcal{C}$ in $\boldsymbol{R}^k$. The following well-known theorem [9] gives a bound on the size of the sample so that a hypothesis that has small error with respect to the sample also has, with high probability, small error with respect to the entire distribution. We give a self-contained proof in the appendix for the reader's convenience.

Theorem 3 *For $\epsilon \leq 1/4$, let w be a concept from $\mathcal{C}$ in $\boldsymbol{R}^k$ that correctly classifies at least a $(1 - \epsilon/8)$ fraction of a sample of m points drawn from $\mathcal{D}$ such that*

$$m \geq \frac{32}{\epsilon}\left(\log C(2m, k) + \log \frac{2}{\delta}\right).$$

Then with probability at least $1 - \delta$, w correctly classifies at least a $1 - \epsilon$ fraction of $\mathcal{D}$.

When the algorithm is able to find a consistent hypothesis (rather than nearly consistent as in the previous theorem), then the sample complexity can be improved by a constant factor (see [9] or [26]).

Theorem 4 *Let $\mathcal{C}$ be any concept class in $\boldsymbol{R}^k$. Let w be a concept from $\mathcal{C}$ that is consistent with m labelled examples of some concept in C. Then with probability at least $1 - \delta$, w correctly classifies at least $(1 - \epsilon)$ fraction of $\mathcal{D}$ provided*

$$m > \frac{4}{\epsilon}\log C(2m, k) + \frac{4}{\epsilon}\log\frac{2}{\delta}.$$

The notion of *VC-dimension* [43] is closely connected to the number of distinct labelings.

Theorem 5 *Let C be a concept class of VC-dimension d. Then, the number of distinct labelings of m points by concepts in C is at most*

$$C[m] \leq \sum_{i=0}^{d} \binom{m}{i}.$$

4.1 Half-spaces

We begin with the problem of learning a half-space in $\boldsymbol{R}^n$ (a linear threshold function). This is one of the oldest problems studied in learning theory. The problem can be solved in polynomial-time by using an algorithm for linear programming on a sample of $O(n)$ examples (note that this is not a *strongly* polynomial algorithm — its complexity depends on polynomially on the number of bits in the input). Typically, however, it is solved by using simple greedy methods. A commonly-used greedy algorithm is the *Perceptron Algorithm* [38, 2], which has the following guarantee: Given a collection of data points in $\boldsymbol{R}^n$, each labeled as *positive* or *negative*, the algorithm will find a vector w such that $w \cdot x > 0$ for all positive points x and $w \cdot x < 0$ for all negative points x, if such a vector exists[1]. The running time of the algorithm depends on a separation parameter (described below). However, in order for the hypothesis to be reliable, we need to use a sample of $\Omega(n)$ points, since the VC-dimension of half-spaces in $\boldsymbol{R}^n$ is $n+1$.

Let $\mathcal{H}_n$ be the class of homogenous half-spaces in $\boldsymbol{R}^n$. Let $(h, \mathcal{D})$ be a concept-distribution pair such that the half-space $h \in \mathcal{H}_n$ is ℓ-robust with respect to the distribution $\mathcal{D}$ over $\boldsymbol{R}^n$. We restrict $\mathcal{D}$ to be over the unit sphere (i.e. all the examples are at unit distance from the origin). The latter condition is not really a restriction since examples can be scaled to have unit length without changing their labels. The parameters k and m in the algorithm below will be specified later.

Half-space Algorithm:

1. Choose an $n \times k$ random matrix R by picking each entry independently from $N(0,1)$ or $U(-1,1)$.

2. Obtain m examples from $\mathcal{D}$ and project them to $\boldsymbol{R}^k$ using R.

3. Run the following Perceptron Algorithm in $\boldsymbol{R}^k$: Let $w = 0$. Perform the following operation until all examples are correctly classified:
   Pick an arbitrary misclassified example x and let $w \leftarrow w + \mathit{label}(x)x$.

4. Output R and w.

A future example x is labelled positive if $w \cdot (R^T x) \geq 0$ and negative otherwise. This is of course the same as checking if $(wR^T) \cdot x > 0$, i.e. a half-space in the original n-dimensional space.

We can assume that h, the normal vector to the concept half-space, is of unit length. The idea behind the algorithm is that when k is large enough, in the k-dimensional subspace obtained by projection, the half-space through the origin defined by $R^T h$, i.e., $(R^T h) \cdot y \geq 0$, classifies most of the projected distribution correctly. We will show that in fact this half-space remains robust with respect to a projected sample of sufficiently large size. To find a consistent half-space, we use the classical perceptron algorithm. It is well-known (see [33]) that the convergence of this algorithm depends on the margin, i.e., in our terminology, the robustness of the target half-space.

Theorem 6 *[33] Suppose the data set S can be correctly classified by some unit vector w. Then, the Perceptron Algorithm converges in at most $1/\sigma^2$ iterations, where*

$$\sigma = min_{x \in S} \frac{|w \cdot x|}{||x||}.$$

For an ℓ-robust half-space, we have $\sigma \geq \ell$. The theorem says that the perceptron algorithm will find a consistent half-space in at most $1/\ell^2$ iterations. We can now state and prove the main result of this section.

[1] A zero threshold can be achieved by adding an extra dimension to the space.

Theorem 7 *An ℓ-robust half-space in $\boldsymbol{R}^n$ can be (ϵ, δ)-learned by projecting a set of m examples to $\boldsymbol{R}^k$ where*

$$k = \frac{100}{\ell^2} \ln \frac{100}{\epsilon \ell \delta}, \quad m = \frac{8k}{\epsilon} \log \frac{48}{\epsilon} + \frac{4}{\epsilon} \log \frac{4}{\delta} = O\left(\frac{1}{\ell^2} \cdot \frac{1}{\epsilon} \cdot \ln \frac{1}{\epsilon} \ln \frac{1}{\epsilon \ell \delta} \right)$$

in time $n \cdot poly(\frac{1}{\ell}, \frac{1}{\epsilon}, \log \frac{1}{\delta})$ time.

Proof. For an example point x, we let x' denote its projection. We let h' denote the projection of h, the normal to the target half-space. We would like the following events to occur (by the choice of the projection matrix R):

1. For each example x, its projection x' has length at most $1 + \frac{\ell}{2}$. Similarly, $||h'|| \leq 1 + \frac{\ell}{2}$.
2. For each example x, if $h \cdot x \geq \ell$, then $h' \cdot x \geq \frac{\ell}{2}$; If $h \cdot x \leq -\ell$, then $h' \cdot x' \leq -\frac{\ell}{2}$.

We now bound the probability that one of these events does not occur. For any single example x, applying Theorem 2 with $\epsilon = \ell/2$ and our choice of k, the probability that $||x'|| > 1 + \frac{\ell}{2}$ is at most

$$e^{-(\frac{\ell^2}{4} - \frac{\ell^3}{8})\frac{k}{4}} \leq e^{-\frac{\ell^2 k}{32}} \leq \left(\frac{\epsilon \ell \delta}{100} \right)^{\frac{100}{32}} < \frac{\delta}{4(m+1)}.$$

Adding this up over all the m examples and the vector h, we get a failure probability of at most $\delta/4$.

Next, by Corollary 2, with $u = h$ and $v = x$, the probability that the second event does not occur for any particular example x is at most $\delta/4m$. Again this contributes a total failure probability of at most $\delta/4$. Thus, both events occur with probability at least $1 - \delta/2$.

These events imply that the half-space in $\boldsymbol{R}^k$ defined by h' correctly classifies all the m examples after projection (with probability at least $1-\delta/2$). Moreover, after scaling the examples to have length at most 1, the margin is at least

$$\sigma \geq \frac{\ell/2}{1 + \frac{\ell}{2}} \geq \frac{\ell}{3}.$$

Now, by Theorem 6, the perceptron algorithm will find a consistent half-space in $9/\ell^2$ iterations.

Finally, we need to show that m is large enough that hypothesis found generalizes well. We will apply Theorem 4 to half-spaces in $\boldsymbol{R}^k$. The VC-dimension of the latter concept class is d and so, by Theorem 5, we get the following well-known bound on the number of distinct half-spaces (see e.g. [26]):

$$C(2m, k) \leq \sum_{i=0}^{k} \binom{2m}{i} \leq \left(\frac{2em}{k} \right)^k. \tag{1}$$

Our choice of m satisfies

$$m = \frac{8k}{\epsilon} \log \frac{48}{\epsilon} + \frac{4}{\epsilon} \log \frac{4}{\delta} > \frac{4}{\epsilon} \log C(2m, k) + \frac{4}{\epsilon} \log \frac{4}{\delta}.$$

Therefore, applying Theorem 4 with $\delta/2$ in place of δ, the half-space found by the algorithm correctly classifies at least $1 - \epsilon$ of the original distribution with probability at least $1 - \delta/2$. This gives an overall success probability of at least $1 - \delta$. $\square$

The perceptron algorithm and its variants are known to be resistant to various types of random classification noise [10, 7]. It is a straightforward consequence that these properties continue to hold for the algorithm described here. In the concluding section, we discuss straightforward bounds for agnostic learning.

4.2 Intersections of half-spaces

The next problem we consider is learning an intersection of t half-spaces in $\boldsymbol{R}^n$, i.e., the positive examples all lie in the intersection of t half-spaces and the negative examples lie outside that region. It is not known how to solve the problem for an arbitrary distribution. However efficient algorithms have been developed for reasonably general distributions assuming that the number of half-spaces is relatively small [8, 44]. Here, we derive efficient learning algorithms for robust concepts in this class.

We assume that all the half-spaces are homogenous. Let the concept class of intersections of half-spaces be denoted by $\mathcal{H}(t,n)$. A single concept in this class is specified by a set of t half-spaces $P = \{h_1, \ldots, h_t\}$, and the positive examples are precisely those that satisfy $h_i \cdot x \geq 0$ for $i = 1 \ldots t$. Let $(P, \mathcal{D})$ be a concept-distribution pair such that P is ℓ-robust with respect to the distribution $\mathcal{D}$. We assume that the support $\mathcal{D}$ is a subset of the unit sphere (and remind the reader that this as well as homogeneity are not really restrictive, as they can be achieved by scaling and adding an extra dimension, respectively; see e.g. [44]).

Let denote $C(m,t,k)$ denote the maximum number of distinct labellings of m examples in R^k using concepts from $\mathcal{H}(t,k)$. Then,

$$C(2m,t,k) \leq \left(\sum_{i=0}^{k-1} \binom{2m}{i} \right)^t \leq \left(\frac{2em}{k} \right)^{tk}. \tag{2}$$

This can be seen as follows: For $t = 1$, this is just (1), the number of ways to assign $+$ or -1 to $2m$ points using a half-space. If we give each point t labels, one for each of t half-spaces, then the total number of possible labellings is the middle term in (2). We consider two labellings distinct iff the subset of points that are labelled $+$ by all t half-spaces is different. Thus the total number of distinct labellings by t half-spaces can only be smaller than this bound.

Given m examples, we can always find a consistent hypothesis (if one exists) using a brute-force algorithm that enumerates all the combinatorially distinct half-spaces and pick t of them (with replacement). We apply this to learning a robust intersection of t-half-spaces *after* projecting a sufficiently large sample to a lower-dimensional subspace. The parameters k and m below will be specified shortly.

t-Half-spaces Algorithm:

1. Choose an $n \times k$ random matrix R for projection by choosing each entry independently from $N(0,1)$ or $U(-1,1)$.
2. Obtain m examples from $\mathcal{D}$ and project them to $\boldsymbol{R}^k$ using R.
3. Find a hypothesis $Q = \{w_1, \ldots, w_t\}$ where each $w_i \in \boldsymbol{R}^k$ such that the intersection of the half-spaces $w_i \cdot x \geq 0$ for $i = 1, \ldots, t$ is consistent with the labels of the projected examples.
4. Output R and Q.

A future example x is projected down as $R^T x$ and labelled according to Q, i.e., it is positive if $w_i \cdot (R^T x) \geq 0$ for all $i = 1, \ldots, t$.

Theorem 8 *An ℓ-robust intersection of t half-spaces in $\boldsymbol{R}^n$ can be (ϵ, δ)-learned by projecting m examples to $\boldsymbol{R}^k$ where*

$$k = \frac{100}{\ell^2} \ln \frac{100t}{\epsilon \ell \delta} \quad \textit{and} \quad m = \frac{8kt}{\epsilon} \log \frac{48t}{\epsilon} + \frac{4}{\epsilon} \log \frac{4}{\delta} = O\left(\frac{t}{\epsilon \ell^2} \log \frac{t}{\epsilon} \log \frac{t}{\ell \epsilon \delta} \right)$$

in time $O(nmk) + \left(\frac{48t}{\epsilon} \log \frac{4t}{\epsilon\delta} \right)^{kt}$.

Proof. The proof is similar to that of Theorem 7 and we only sketch it.

Let the original set of half-spaces be $h_1 \cdot x \geq 0, \ldots, h_t \cdot x \geq 0$, where each h_i is a unit vector in $\boldsymbol{R}^n$. We consider the projections of these, $h_i' = \frac{1}{\sqrt{k}} R^T h_i$, and the following events: For each example x and normal vector h_i, if $h_i \cdot x \geq \ell$, then $h_i' \cdot x' > 0$; If $h_i \cdot x \leq -\ell$, then $h_i' \cdot x' < 0$.

For our choice of k and m, it follows from Corollary 2 that these events all happen with probability at least $1-\delta/2$. Therefore, after projection, with this probability, there is a hypothesis from $\mathcal{H}(t,k)$ that is consistent with all m examples. Using Theorem 4 along with (2), it follows that any hypothesis consistent with a sample of size

$$m = \frac{8kt}{\epsilon} \log \frac{2t}{\epsilon} + \frac{4}{\epsilon} \log \frac{4}{\delta}$$

will correctly classify $(1-\epsilon)$ of the distribution with probability at least $1-\delta/2$. This gives an overall success probability of at least $1-\delta$. The running time of the enumerative algorithm is $O((2em/k)^{kt})$. □

If $t, \ell, \epsilon, delta$ are all constant, then the algorithm runs in linear time. If only ℓ, ϵ, δ are constant, then the algorithm has running time $O(nt \log^3 t) + (t \log t)^{O(t \log t)}$. This is significantly faster than the best-known algorithms for the general case (see Section 1.1 for recent improvements). Both results do not need any further assumptions on the distribution $\mathcal{D}$ besides robustness. Previous algorithms for the problem assumed that $\mathcal{D}$ was either symmetric [5], uniform [8] or non-concentrated [44]. Recently, an improved time complexity for learning robust intersections of half-spaces was obtained in [27] using an algorithm for learning polynomial threshold functions in the projected space in place of the enumerative algorithm used here. The improvement in the time complexity comes along with a substantial increase in the sample complexity.

4.3 Balls

Finally, we briefly discuss the concept class of balls in $\boldsymbol{R}^n$, illustrating how robustness plays a role in learning nonlinear concepts.

A Ball $B(x_0, r)$ in $\boldsymbol{R}^n$ is defined as

$$B(x_0, r) = \{x \in \boldsymbol{R}^n \; : \; ||x - x_0|| \leq r\}$$

where x_0 (the center) is a fixed point in $\Re^n$ and r (the radius) is a fixed real value. The set of points in $B(x_0, r)$ are labelled positive and those outside are labelled negative.

It is well-known that the VC-dimension of balls in $\boldsymbol{R}^n$ is $n+1$ and so the number of examples required to (ϵ, δ)-learn a ball is $O(\frac{n}{\epsilon} \log \frac{1}{\epsilon} + \frac{1}{\epsilon} \log \frac{1}{\delta})$. How many examples do we need to learn an ℓ-robust ball? The following theorem follows easily from the neuronal projection theorem.

Theorem 9 *An ℓ-robust ball of radius in $\boldsymbol{R}^n$ of radius at most 1 can be (eps, δ)-learned by projecting m examples to $\boldsymbol{R}^k$ where*

$$k = \frac{100}{\ell^2} \ln \frac{100}{\epsilon \ell \delta} \quad \textit{and} \quad m = \frac{8k}{\epsilon} \log \frac{48}{\epsilon} + \frac{4}{\epsilon} \log \frac{4}{\delta}$$

and then finding a ball in $\boldsymbol{R}^k$ consistent with the projected examples.

Proof. With probability 1, any positive example x drawn from the distribution $\mathcal{D}$ will satisfy

$$||x - x_0|| \leq r - l$$

while any negative example x will satisfy

$$||x - x_0|| \geq r + l.$$

Using Theorem 2 with our choice of k and $\epsilon = \ell/2$, for any one x, its projection x' satisfies

$$(1 - \frac{\ell}{2})||x - x_0|| \leq ||x' - x'_0|| \leq (1 + \frac{\ell}{2})||x - x_0||$$

with probability at least $1 - \frac{\delta}{2m}$. So, with probability $1 - \delta/2$, all the projected examples satisfy the above inequality. Further, since the radius of the concept ball is at most 1,

$$||x - x_0|| + \frac{\ell}{2} \leq ||x' - x'_0|| \leq ||x - x_0|| + \frac{\ell}{2}.$$

Thus, the ball $B(x'_0, r)$ in $\boldsymbol{R}^k$ is consistent with the projected examples and the theorem follows. Finally, we can use Theorem 4 to verify that m is large enough for this to be an (ϵ, δ)-learning algorithm. $\square$

5 Discussion

5.1 Robust concepts are easier

The model studied in this paper could also be viewed as a rudimentary model of human learning. In this model, at the outer level of processing, there is a layer of neurons that produces a random summary of stimuli presented. It is this summary that is then processed by learning algorithms. The outer level plays the role of random projection. The main insight of the model is that even a random summary that is *independent* of the specific concept being learned can preserve the essential elements of the classification and thus is sufficient for learning. The ease of learning and the extent to which the summary preserves the concepts depends on their robustness — the more robust a concept, the shorter a summary needs to be and the easier it is to learn. In this section, we draw from work in psychology to see how the predictions of this model hold up.

An interesting prediction of the theorems of the previous section is that learning concepts that are more robust requires fewer examples. This prediction is supported by many psychological studies [20, 29, 34, 35, 36, 37].

Psychological literature reports that *natural categories* (concept classes) as formed by humans are hierarchical [34], with three clear levels called the *Superordinate, Basic level* and *Subordinate*. For example, for the Superordinate category of *Mammals*, some Basic level categories are Elephant, Dog, Human, and the Subordinate categories for Elephant would be African Elephant and Indian Elephant. Similarly, the Superordinate category of *Musical Instruments* has below it the Basic level categories of Guitar, Piano, Drum, etc. and Guitar, has below it Subordinate categories such as Folk Guitar and Classical Guitar. The Basic level categories are considered the most important, and are the most clearly demarcated from each other. In our terminology they are the most *robust*, and we expect them to be easier to learn. This is indeed the case as noted by Rosch et al. [37], "...basic level categories are the most differentiated from one another and they are therefore the first categories we learn and the most important in language."

A leading theory of how humans form categories is based on the notion of *Prototypes* [29, 20]. Prototypes represent the most typical members of a category. The theory says that we abstract a prototype for a category by forming some weighted average of (a subset of) the defining features of examples from the category. A subsequent instance is compared to the prototype, and if it has a sufficient degree of similarity to the prototype then it is judged to be a member of the category. This explains the results of studies where it is found that when asked to list examples of a category, subjects consistently list members that are closer to the prototype both earlier and more often (e.g. for the category Bird, the examples Sparrow and Robin are produced more often than Ostrich) [36]. Further when asked to classify instances, it is found that examples closer to the prototype are classified more quickly. Similar results were found in studies with artificially generated categories [35].

For a prototype P, we could define a family of nested concepts within the category of P according to the distance from P. Then the members of the innermost concept are very similar to P, the members of the next concept are a bit more varied, and the variation increases as the maximum distance of the concepts from P grows. In other words the innermost concept is the most robust in terms of demarcating the category from objects that are not members, and the robustness decreases as we move outward. The arguments in this paper imply that the inner concepts would be easier to learn and label than the outer ones. This is exactly what was observed in the aforementioned studies.

5.2 Noise tolerance

It is worth mentioning that the algorithms can be adapted to be resistant to malicious classification noise (agnostic learning). Let γ be the noise rate ($0 \leq \gamma < 1/2$), i.e. if we choose a sample of s examples, γs of their labels are corrupted arbitrarily. Fix a hypothesis class H and let $f(\ell)$ be the bound on the number of examples required to learn concepts with robustness ℓ. Then to deal with noise rate γ, we use a sample of size $f(\ell)/(1-\gamma)$, and then for every subset of size $f(\ell)$ of the sample, we run the learning algorithm for the hypothesis class and output a hypothesis that correctly classifies the subset. The total number of runs of the algorithms is at most $2^{f(\ell)}$. So, for example, half-spaces can be learned in $poly(n)$ time for robustness as low as $\sqrt{\frac{\log\log n}{\log n}}$. Another way to interpret this is that we can find hypothesis that minimize the number of mistakes. This was observed by Avrim Blum.

5.3 Open problems

An important open problem in computational learning theory is that of learning DNF formulae from the uniform distribution without membership queries. This concept class can be viewed as intersections of half-spaces with robustness at least $1/\sqrt{n}$. If the algorithm can be made to run for boolean formulae in time polynomial in n and $1/\ell$, this would solve the problem of learning DNF formulae without membership queries.

We have seen that robust linear thresholds are easy to learn. What is the complexity of robust polynomial threshold functions? What are concept classes for which robustness does not reduce learning complexity?

References

[1] D. Achlioptas: Database friendly random projections. In Proc. Principles of Database Systems (PODS), pages 274–281, 2001.

[2] S. Agmon: The relaxation method for linear inequalities. Canadian Journal of Mathematics, 6(3):382–392, 1954.

[3] R. I. Arriaga and S. Vempala: An algorithmic theory of Learning: robust concepts and random projection. Proc. of the 39th IEEE Foundations of Computer Science, 1999.

[4] N. Balcan, A. Blum and S.Vempala: On Kernels, Margins and Low-dimensional mappings. To appear in Proc. of ALT 2004.

[5] E. B. Baum: On learning a union of half spaces, Journal of Complexity **6**(1), 67–101, 1990.

[6] S. Ben-David, N. Eiron and H. Simon: Limitations of Learning Via Embeddings in Euclidean Half Spaces. Journal of Machine Learning Research **3**, 441–461, 2002.

[7] A. Blum, A. Frieze, R. Kannan, and S. Vempala: A Polynomial-time Algorithm for Learning Noisy Linear Threshold Functions. Proc. of the 37th IEEE Foundations of Computer Science, 1996.

[8] A. Blum and R. Kannan: Learning an intersection of k halfspaces over a uniform distribution. Proc. of the 34th IEEE Symposium on the Foundations of Computer Science, 1993.

[9] A. Blumer, A. Ehrenfeucht, D. Haussler and M. K. Warmuth: Learnability and the Vapnik-Chervonenkis dimension. J. ACM **36**(4), 929–965, 1989.

[10] T. Bylander: Learning linear threshold functions in the presence of classification noise. Proc. 7th Workshop on Computational Learning Theory, 1994.

[11] P. Bartlett and J. Shawe-Taylor: Generalization Performance of Support Vector Machines and Other Pattern Classifiers. In *Advances in Kernel Methods - Support Vector Learning* (Ed.s B. Schvlkopf, C. Burges, and A. J. Smola) MIT Press, 1998.

[12] E. Cohen: Learning noisy perceptrons by a perceptron in polynomial time. Proc. of the 38th IEEE Foundations of Computer Science, 1997.

[13] C. Cortes and V. Vapnik: Support-Vector Networks. Machine Learning 20, 273-297, 1995.

[14] S. Dasgupta and A. Gupta: An elementary proof of the Johnson-Lindenstrauss Lemma. Tech Rep. U.C. Berkeley, 1999.

[15] W. Feller, *An Introduction to Probability Theory and Its Applications*, John Wiley and Sons, Inc., 1957.

[16] P. Frankl and H. Maehara: The Johnson-Lindenstrauss Lemma and the Sphericity of some graphs, J. Comb. Theory B **44**, 355–362, 1988.

[17] Y. Freund and R. E. Schapire, "Large margin classification using the perceptron algorithm. Machine Learning, 37(3):277-296, 1999.

[18] A. Garg, S. Har-Peled and D. Roth: On generalization bounds, projection profile, and margin distribution. ICML,171-178, 2002.

[19] A. Garg and D. Roth: Margin Distribution and Learning. ICML, 210–217, 2003.

[20] A. L. Glass, K. J. Holyoak, J. L. Santa: The Structure of Categories. In *Cognition*, Addison-Wesley, 1979.

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

[22] W. Hoeffding: Probability inequalities for sums of bounded random variables. Journal of the American Statistical Association, 58, 13-30, 1963.

[23] P. Indyk and R. Motwani: Approximate nearest neighbors: Towards removing the curse of dimensionality. Proc. of ACM STOC 1998.

[24] W. B. Johnson and J. Lindenstrauss: Extensions of Lipshitz mapping into Hilbert space. Contemporary Mathematics, 26, 189-206, 1984.

[25] M. J. Kearns and R. E. Schapire: Efficient distribution-free learning of probabilistic concepts. Journal of Computer and System Sciences, 48(3):464-497, 1994.

[26] M. J. Kearns and U. Vazirani: Introduction to Computational Learning Theory, *MIT Press*, 1994.

[27] A. Klivans and R. Servedio: Learning intersections of halfspaces with a margin, Proc. 17th Workshop on Computational Learning Theory, 2004.

[28] J. Kleinberg: Two algorithms for nearest-neighbor search in high dimensions. Proc. 29th ACM Symposium on Theory of Computing, 1997.

[29] L. K. Komatsu: Recent views on conceptual structure. Psychological Bulletin, 112(3), 1992.

[30] N. Linial, E. London and Y. Rabinovich: The geometry of graphs and some of its algorithmic applications. Proc. of 35th IEEE Foundations of Computer Science, 1994.

[31] N. Littlestone: Learning Quickly When Irrelevant Attributes Abound: A New Linear-threshold Algorithm. Machine Learning, 2:285-318, 1987.

[32] N. Littlestone: Redundant noisy attributes, attribute errors, and linear threshold learning using winnow. Proc. 4th Workshop on Computational Learning Theory, 1991.

[33] M. Minsky and S. Papert. *Perceptrons: An Introduction to Computational Geometry.* The MIT Press, 1969.

[34] S. K. Reed: Categorization, in *Cognition: Theory and Applications*, Brooks/Cole, 1982.

[35] S. K. Reed and M. P. Friedman: Perceptual vs. Conceptual Categorization. Memory and Cognition, 1, 1973.

[36] E. Rosch: Principles of Categorization" in *Cognition and Categorization* (Ed.s E. Rosch and B. B. Lloyd), Hillsdale, 1978.

[37] E. H. Rosch, C. B. Mervis, W. D. Gray, D. M. Johnson and P.Boyes-Braem: Basic ojects in natural categories. Cognitive Psychology, 8, 1976.

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

[39] R. E. Schapire, Y. Freund, P. L. Bartlett, W. S. Lee, Boosting the margin: a new explanation for the effectiveness of voting methods, Annals of Statistics **26**(5), 1651–1686, 1998.

[40] L. G. Valiant: A Theory of the Learnable. Communications of the ACM, 27(11), 1134-1142.

[41] L. G. Valiant: A Neuroidal Architecture for Cognitive Computation. Proc. of ICALP 98.

[42] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, 1995.

[43] V. N. Vapnik and A. Ya. Chervonenkis: On the uniform convergence of relative frequencies of events to their probabilities. Theory of Probability and its applications, XVI(2):264–280, 1971.

[44] S. Vempala: A Random Sampling based Algorithm for learning the Intersection of Half-spaces. Proc. of the 38th IEEE Foundations of Computer Science, 1997.

[45] S. Vempala: Random Projection: A New Approach to VLSI Layout. Proc. of the 39th IEEE Foundations of Computer Science, 1998.

6 Appendix

Proof. (of Theorem 3.)

We basically mimic the proof of the fundamental VC theorem. The only difference is that in that theorem, it is assumed that there is a hypothesis consistent with the *entire* sample. Here we can only assume that there is a hypothesis that correctly classifies $1 - \epsilon/4$ fraction of the sample.

Let us call a hypothesis a bad hypothesis if it has error more than ϵ on the distribution. Let A be the event that there exists a bad consistent hypothesis, i.e., a hypothesis that has error less than $\epsilon/8$ on the sample and error greater than ϵ on the distribution. We would like to show that the probability of event A is at most δ. To do this, we define B to be the event that for a sequence of $2m$ examples, there is a concept that has error less than $\epsilon/8$ on the first m and greater than $\epsilon/2$ on the remaining m.

Next we observe that $Pr(A) \leq 2 \cdot Pr(B)$. This is because

$$Pr(B) \geq Pr(A \text{ and } B) = Pr(A) \cdot Pr(B/A)$$

The probability of B given A, $Pr(B/A)$ is the probability that a hypothesis that has error ϵ on the distribution has error at least $\epsilon/2$ on a set of m examples. Using Chebychev's inequality, this latter probability is at least $1/2$.

To complete the proof we will bound the probability of B. Fix any set of $2m$ examples and consider a random partition of them into two equal-sized sets S_1 and S_2. Let $\hat{h}$ be a hypothesis which disagrees with the target hypothesis on at least $\epsilon m/2$ of the $2m$ examples. This is a candidate for causing event B.

Let X_i, for $i = 1, \ldots, m$ denote the event that $\hat{h}$ makes an error on the i'th example in S_1. Then $E(X_i) = \epsilon/4$. Define

$$X = \sum_{i=1}^{m} X_i.$$

Then $E(X) = \epsilon m/4$. By Chernoff's inequality,

$$Pr(X \leq \frac{\epsilon}{4}(1-c)) \leq e^{-\frac{\epsilon m c^2}{8}}$$

That is,

$$Pr(X \leq \frac{\epsilon}{8}) \leq e^{-\epsilon m/32}.$$

The total number of distinct hypothesis for the set of $2m$ examples is at most $C(2m, k)$. In other words, this is the number of distinct ways to partition $2m$ points using concepts from $\mathcal{C}$ in $\boldsymbol{R}^k$. Adding up over all the hypotheses, we get that

$$Pr(B) \leq C(2m, k)e^{-\epsilon m/32}.$$

For the value of m considered in the theorem, we have $Pr(B) < \delta/2$ and hence $Pr(A) < \delta$ as required. $\square$