# On Exact Learning Halfspaces with Random Consistent Hypothesis Oracle

Ehab Wattad

Technion - Computer Science Department - M.Sc. Thesis MSC-2006-30 - 2006

# On Exact Learning Halfspaces with Random Consistent Hypothesis Oracle

Research Thesis

Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Computer Science

**Ehab Wattad**

Submitted to the Senate of
the Technion — Israel Institute of Technology

Ayar 5766 Haifa March 2006

The research was done under the supervision of Prof. Nader Bshouty in the Department of Computer Science.

The generous financial help of the Technion Graduate School is gratefully acknowledged.

I would like to thank Prof. Nader, without his great ideas and guidance this thesis whouldn't have been possible.

I also want to thank my parents, Abed and Rashida, and family for always believing and pushing.

Special thanks to my lovely wife Islam for her support and encouragement.

# Contents

# List of Figures

# Abstract

We investigate several learning strategies for exact learning halfspaces over the domain $\{0, 1, \ldots, n-1\}^d$ and study the query complexity and the time complexity of exact learning using those strategies. Our strategies are based on the RCH-oracle that returns a random consistent hypothesis with the counterexamples received from the equivalence query oracle.

We first give a new polynomial time learning algorithm that uses the RCH-oracle for learning halfspaces from majority of halfspaces. We show that the query complexity of this algorithm is less (by some constant factor) than the best known algorithm that learns halfspaces from halfspaces.

We then study the query complexity of exact learning when limited number of calls to the RCH-oracle is allowed in each trial, i.e., before each equivalence query. We first show that an $\tilde{O}(d)$ calls to the RCH-oracle in each trial is sufficient for learning in polynomial number of queries. We then show that any "reasonable" strategy must use the RCH-oracle at least $\Omega(\sqrt{d})$ times in each trial.

Then we show that if only one call to RCH-oracle is allowed in each trial then the query complexity of the learning algorithm is $2^{\Theta(d)} \log n$. We then give a tight lower bound $2^{\Omega(d)} + \Omega(d^2 \log n)$. This proves that this learning algorithm does not run in polynomial time for $d = \omega(\log \log n)$.

# List of Symbols & Abbreviations

| | |
|---|---|
| $O$ | Asymptotic upper bound. |
| $o$ | Upper bound that is not asymptotically tight. |
| $\tilde{O}$ | $O$ with logarithmic factors ignored. |
| $\Omega$ | Asymptotic lower bound. |
| $\omega$ | Lower bound that is not asymptotically tight. |
| $\Theta$ | Asymptotic tight bound. |
| $\mathcal{U}$ | The uniform distribution. |
| $x \in_D X$ | $x$ is chosen from $X$ according to the distribution $D$. |
| $E_{x \in_D X}$ | The expectation of $x$ over the distribution $D$. |
| $E_X$ | Same as $E_{x \in_D X}$. |
| $E_{x \in_{\mathcal{U}} S}$ | The expectation of $x$ over the uniform distribution. |
| $E_S$ | Same as $E_{x \in_{\mathcal{U}} S}$. |
| $C$ | Concept class. |
| $C^{\perp}$ | The dual concept class. |
| $F_{\mid_A}$ | The *projection of $F$ on $A$*. |
| VCdim | The *Vapnik-Chervonenkis dimension*. |
| $\text{VCdim}^{\perp}$ | The *dual VC-dimension*. |
| $\text{HS}_X^d$ | The class of all halfspace functions over $X$. |
| $\text{HS}_n^d$ | The class of all halfspace functions over $[n]^d = \{0, 1, \ldots, n-1\}^d$. |
| $\text{HS}^d$ | The class of all halfspace functions over $\Re^d$. |
| $H^d$ | The set of all hyperplanes in $\Re^d$. |
| EQ | Equivalence query. |
| $V_d(R)$ | The volume of a $d$-ball of *radius* $R$. |
| $S_d(R)$ | The surface area of a $d$-sphere of *radius* $R$. |

# Chapter 1

# Introduction

In this thesis we consider learning the class $\mathrm{HS}_X^d$ for $X = \{0, 1, \ldots, n-1\}^d$.
$\mathrm{HS}_X^d$ is the class of all halfspaces $f_{w,t}$ where $w = (w_1, \ldots, w_d) \in \Re^d$ and $t \in \Re$,

$$f_{w,t}(x) = \begin{cases} 1 & w^T x \geq t \\ 0 & \text{otherwise} \end{cases}$$

for every $x \in X$.
When $X = \{0, 1, \ldots, n-1\}^d$ we denote $\mathrm{HS}_X^d$ by $HS_n^d$ and when $X = \Re^d$ we denote it by $HS^d$.
We consider learning via a standard learning model called *exact* learning with *equivalence queries*. In this model we assume some unknown target halfspace $f$ is chosen by a teacher. The learning algorithm may propose as a hypothesis any formula $g$ by making an *Equivalence Query* ( $\mathrm{EQ}(g)$ ) to the teacher. If $g$ is logically equivalent to $f$ then the teacher returns "yes" otherwise it returns a counterexample, i.e, a point $a$ s.t. $f(a) \neq g(a)$. The goal of the algorithm is to identify $f$ with minimal number of equivalence queries.

We investigate several learning strategies for exact learning the class $HS_n^d$ and study the query complexity and the time complexity of exact learning using those strategies. Our strategies are based on two basic oracles. An $\mathrm{RCH}_{\mathcal{C}}$-oracle that chooses a uniform random consistent halfspace (with the counterexamples) from $\mathrm{HS}_n^d$ and an RCH-oracle that chooses a random consistent halfspace from $HS^d$. The advantage of the RCH-oracle over the $\mathrm{RCH}_{\mathcal{C}}$-oracle is that it can be simulated in polynomial time [L98].

We first give a new polynomial time learning algorithm that uses the RCH-oracle for learning halfspaces. We show that the query complexity of this algorithm is less (by some constant factor) than the best known algorithm that learns halfspaces.

We then study the query complexity of exact learning when limited number of calls to the RCH-oracle is allowed in each trial, i.e., before each equivalence query. We first show that an $\tilde{O}(d)$ calls to the RCH-oracle in each trial is sufficient for learning in polynomial number of queries. We then show that any "reasonable" strategy must use the RCH-oracle at least $\Omega(\sqrt{d})$ times in each trial.

Then we show that if only one call to RCH-oracle is allowed in each trial then the query complexity of the learning algorithm is $2^{\Theta(d)} \log n$. Since the RCH-oracle can be simulated in polynomial time this learning algorithm runs in polynomial time for $d = O(\log \log n)$. We then give a tight lower bound $2^{\Omega(d)} + \Omega(d^2 \log n)$. This proves that this learning algorithm does not run in polynomial time for $d = \omega(\log \log n)$.

The thesis is organized as follows: In this chapter we give some preliminary results in probability theory and some properties of Halfspaces and define the learning model. In chapter 2 we give old and new results. In chapter 3 we give the randomized halving algorithm that uses the $\mathrm{RCH}_{\mathcal{C}}$-oracle and investigate the number of calls to $\mathrm{RCH}_{\mathcal{C}}$-oracle in each trial. In chapter 4 we give our algorithm that learns in polynomial time Halfspaces and analyze its complexity. All the other results of this thesis are in chapter 5. We conclude with some open problems in chapter 6.

## 1.1 Preliminaries

In this section we give some preliminaries and introduce some terms and concepts that will be used throughout the thesis

### 1.1.1 Probability

Let $F$ be a boolean functions $F : X \to \{0, 1\}$ and $D$ a distribution on $X$. Let $\mathcal{U}$ be the uniform distribution over $X$. We will write $x \in_D X$ when we want to indicate that $x$ is chosen from $X$ according to the distribution $D$. Suppose we randomly and independently choose $S = \{x_1, \ldots, x_m\}$ from $X$, each $x_i$ according to the distribution $D$. We will write $E_X$ for $E_{x \in_D X}$. So for finite $X$ we have

$$E_X[F(x)] = \sum_{x \in X} D(x)F(x).$$

We use $E_S$ for $E_{x \in_{\mathcal{U}} S}$. So for a finite sample $S \subset X$ we have

$$E_S[F(x)] = \sum_{x \in S} \frac{F(x)}{|S|}.$$

We say that $\mathcal{S} = (X, C)$ is a *range space* if $C$ is a set of boolean functions $X \to \{0, 1\}$. Each function in $C$ can be also regarded as a subset of $X$. We will also call $C$ *concept class*. For a boolean function $F \in C$ and a subset $A \subseteq X$ the *projection of $F$ on $A$* is the boolean function $F_{|_A} : A \to \{0, 1\}$, such that, for every $x \in A$ we have $F_{|_A}(x) = F(x)$. For a subset $A \subseteq X$ we define the *projection of $C$ on $A$* to be the set

$$P_C(A) = \{F_{|_A} \mid F \in C\}.$$

If $P_C(A)$ contains all the functions, $2^A$, then we say that $A$ is *shattered*. The *Vapnik-Chervonenkis dimension* (or VC-dimension) of $\mathcal{S}$, denoted by $\text{VCdim}(\mathcal{S})$, is the maximum cardinality of a shattered subset of $X$.

Let $(X, C)$ be a range space and $D$ be a distribution on $X$. We say that a set of points $S \subseteq X$ is an *$\epsilon$-net* if any $F \in C$ satisfies $E_X[F(x)] > \epsilon$ contains at least one positive point, i.e., a point $y$ in $S$ such that $F(y) = 1$. Notice that $E_S[F(x)] = 0$ if and only if $S$ contains no positive point for $F$. Therefore, $S$ is not an $\epsilon$-net if and only if

$$(\exists F \in C)\ E_X[F(x)] > \epsilon \text{ and } E_S[F(x)] = 0.$$

We say that $S$ is $\epsilon$-sample if

$$(\forall F \in C)\ |E_X[F(x)] - E_S[F(x)]| \leq \epsilon.$$

Notice that an $\epsilon$-sample is an $\epsilon$-net.

We now list few results from the literature

**Lemma 1.** *Let $F : X \to \{0, 1\}$ be a boolean function. Suppose we randomly and independently choose $S = \{x_1, \ldots, x_m\}$ from $X$ according to the distribution $D$.*
**Bernoulli** *For*

$$m = \frac{1}{\epsilon} \ln \frac{1}{\delta}$$

*we have*

$$\Pr[E_X[F(x)] > \epsilon \text{ and } E_S[F(x)] = 0] \leq \delta.$$

**Chernoff** *(Additive form) For*

$$m = \frac{1}{2\epsilon^2} \ln \frac{2}{\delta}$$

*we have*

$$\Pr\left[|E_X[F(x)] - E_S[F(x)]| > \epsilon\right] \leq 2e^{-2\epsilon^2 m} = \delta.$$

It follows from Lemma 1

**Lemma 2.** *Let $C$ be a concept class of boolean functions $F : X \to \{0, 1\}$. Suppose we randomly and independently choose $S = \{x_1, \ldots, x_m\}$ from $X$ according to the distribution $D$.*
**Bernoulli** *For any finite concept class $C$ and*

$$m = \frac{1}{\epsilon}\left(\ln |C| + \ln \frac{1}{\delta}\right)$$

*we have*

$$\Pr\left[(\exists F \in C)\ E_X[F(x)] > \epsilon \text{ and } E_S[F(x)] = 0\right] \leq \delta.$$

*That is, with probability at least $1-\delta$, the set $S$ is $\epsilon$-net.*

**Chernoff** *(Additive form) For any finite concept class $C$ and*

$$m = \frac{1}{2\epsilon^2}\left(\ln|C| + \ln\frac{2}{\delta}\right)$$

*we have*

$$\Pr\left[(\exists F \in C)\,|E_X[F(x)] - E_S[F(x)]| > \epsilon\right] \leq \delta.$$

*That is, with probability at least $1-\delta$, the set $S$ is an $\epsilon$-sample.*

The following uses the VCdim and for many classes $C$ gives a better bound

**Lemma 3.** *Let $C$ be a concept class of boolean functions $F : X \to \{0,1\}$. Suppose we randomly and independently choose $S = \{x_1, \ldots, x_m\}$ from $X$ according to the distribution $D$.*

**$\epsilon$-Net** *([HW87], [BEHW89]) There is a constant $c_{Net}$ such that for any concept class $C$ and*

$$m = \frac{c_{Net}}{\epsilon}\left(VCdim(C)\log\frac{1}{\epsilon} + \log\frac{1}{\delta}\right)$$

*we have*

$$\Pr\left[(\exists F \in C)\ E_X[F(x)] > \epsilon \text{ and } E_S[F(x)] = 0\right] \leq \delta.$$

*That is, with probability at least $1-\delta$, the set $S$ is $\epsilon$-net.*

**$\epsilon$-Sample** *([VC71]) There is a constant $c_{VC}$ such that for any concept class $C$ and*

$$m = \frac{c_{VC}}{\epsilon^2}\left(VCdim(C)\log\frac{VCdim(C)}{\epsilon} + \log\frac{1}{\delta}\right)$$

*we have*

$$\Pr\left[(\exists F \in C)\,|E_X[F(x)] - E_S[F(x)]| > \epsilon\right] \leq \delta.$$

*That is, with probability at least $1-\delta$, the set $S$ is an $\epsilon$-sample.*

### 1.1.2 Halfspaces and Hyperplane

A Halfspace is a simple model of neuron activity. A simplified account of how neuron works can be found in [P94] Chapter 3.

Let $w = (w_1, \ldots, w_d) \in \Re^d$ and $t \in \Re$. We define the *halfspace* $f_{w,t}$ over $X \subseteq \Re^d$ (also called *linear threshold function* [P94] and *Perceptron* [MP43]) as follows:

$$f_{w,t}(x) = \begin{cases} 1 & w^T x \geq t \\ 0 & \text{otherwise} \end{cases}$$

for every $x \in X$. We will also use the notation $f_{w,t} = [w^T x \geq t]$. The constants $w = (w_1, \ldots, w_d)$ are called the *weights* and $t$ is called the *threshold* value. The class $\mathrm{HS}_X^d$

is the class of all halfspace functions over $X$. For the sake of notational convenience when $X = [n]^d = \{0, 1, \ldots, n-1\}^d$ we denote $\mathrm{HS}_X^d$ by $\mathrm{HS}_n^d$ and when $X = \Re^d$ we denote it by $\mathrm{HS}^d$. When $n = 2$ we call $f_{w,t} \in \mathrm{HS}_2^d$ a *threshold* boolean function. When the threshold value $t = 0$ we write $f_w$ for $f_{w,t}$ and call it *zero halfspace* (also called *zero threshold* [P94]).

The boundary of $f_w$,

$$[w^T x = 0] = \{x \mid w^T x = 0\}$$

is called a *hyperplane* (or *separating hyperplane* [P94]). We denote by $H^d$ the set of all hyperplanes in $\Re^d$.

We will now give some results for Halfspaces that will be used in the sequel. We start with the following

**Lemma 4.** *([P94]) We have*

$$n^{d(d+1)} > |HS_n^d| > n^{d(d-1)/2}.$$

The proof in ([P94]) is for $\mathrm{HS}_2^d$. See Appendix A for a proof for $\mathrm{HS}_n^d$.

Another fact that will be used in this thesis is that the VC-dimension of $\mathrm{HS}_X^d$ is at most $d + 1$ and of $\mathrm{HS}_2^d$ is exactly $d + 1$, [WD81].

## 1.2 Learning Models and Definitions

In the *Online learning model* [L88] the learning task is to identify an unknown *target* Halfspace $f$ that is chosen by a *teacher* from $\mathrm{HS}_X^d$. At each *trial*, the teacher sends a point $x \in X$ to the *learner* and the learner has to predict $f(x)$. The learner returns to the teacher the prediction $y$. If $f(x) \neq y$ then the teacher returns "mistake" to the learner. The goal of the learner is to minimize the number of prediction mistakes.

In the online learning model we say that algorithm $\mathcal{A}$ of the learner *Online learns* the class $\mathrm{HS}_X^d$ if for any $f \in \mathrm{HS}_X^d$ and for any $\delta$, algorithm $\mathcal{A}(\delta)$ with probability at least $1 - \delta$ makes a bounded number of mistakes. We say that $\mathrm{HS}_X^d$ is *Online learnable* with $t$ mistakes if the number of mistakes is bounded by $t$. We say that $\mathrm{HS}_X^d$ is *efficiently Online learnable* with $t$ mistakes if the number of mistakes is bounded by $t$ and the running time of the learner for each prediction is $poly(1/\delta, d, \log|X|)$. The bound of the number of mistakes $t$ of an algorithm is also called *the mistake bound* of the algorithm.

In the *Exact learning model* [A88] the learning task is to identify an unknown *target* Halfspace $f$, that is chosen by a *teacher* from $\mathrm{HS}_X^d$, from *queries*. The learner at each trial sends the teacher a hypothesis $h$ from some class of hypothesis $H$ and asks the teacher whether this hypothesis is equivalent to the target function (this is called the *equivalence*

*query*). The teacher either sends back a "YES" indicating that $h$ is equivalent to the target function $f$ or, otherwise, it sends a counterexample $a$. That is, an instance $a \in X$ such that $h(a) \neq f(a)$.

In the exact learning model we say that algorithm $\mathcal{A}$ of the learner *Exactly learns* the class $\mathrm{HS}_X^d$ from $H$ if for any $f \in \mathrm{HS}_X^d$ and for any $\delta$, algorithm $\mathcal{A}(\delta)$ with probability at least $1-\delta$ makes a bounded number of equivalence queries and finds a hypothesis in $H$ that is equivalent to the target function $f$. We say that $\mathrm{HS}_X^d$ is *Exactly learnable* from $H$ with $t$ equivalence queries if the number of equivalence queries is bounded by $t$. We say that $\mathrm{HS}_X^d$ is *efficiently exactly learnable* from $H$ with $t$ equivalence queries if the number of equivalence queries is bounded by $t$ and the running time of the learner is $poly(1/\delta, d, \log|X|)$.

It is known [A88] that if $\mathrm{HS}_X^d$ is exactly learnable from $H$ with $t$ equivalence queries then $\mathrm{HS}_X^d$ is online learnable with $t$ mistakes. If $\mathrm{HS}_X^d$ is efficiently exactly learnable from $H$ with $t$ equivalence queries and elements of $H$ are efficiently computable (for each $h \in H$ and $x \in X$ we can compute $h(x)$ in polynomial time) then $\mathrm{HS}_X^d$ is efficiently Online learnable with $t-1$ mistakes.

# Chapter 2

# Old and New Results

In this thesis we consider different learning strategies for exact learning halfspaces and study the query complexity and time complexity of learning with those strategies. Our strategies are based on two basic oracles:

1. An RCH-oracle that chooses a uniform random Halfspace that is consistent with the counterexamples seen so far.

2. An $\mathrm{RCH}_C$-oracle that chooses a uniform random hypothesis from the class $C$ being learned, that is consistent with the counterexamples seen so far.

We will study the query complexity as well as the number of calls to the RCH-oracles and $\mathrm{RCH}_C$-oracle.
Unlike the *equivalence query* (the teacher), these oracles are part of the learning algorithm, and their running time should be considered when calculating the time complexity of the learning algorithm. The RCH-oracle can be simulated in polynomial time, [L98], and therefore all the algorithms in this thesis that uses this oracle runs in polynomial time. On the other hand, it is not known how to simulate the $\mathrm{RCH}_C$-oracle in polynomial time.

The first algorithm considered in this thesis is the Halving algorithm [A88, L88]. In the Halving algorithm the learner chooses at each trial the majority of all the Halfspaces that are consistent with the examples seen so far. Then it asks equivalence query with this hypothesis. Each counterexample for this hypothesis eliminates at least half of the consistent halfspaces. Therefore, by Lemma 4 the query complexity of the Halving algorithm is at most

$$\log |\mathrm{HS}_n^d| \leq d(d+1)\log n.$$

The randomized Halving algorithm [BC+96] uses the $\mathrm{RCH}_C$-oracle and asks on average

$$(1+c)d(d+1)\log n$$

equivalence queries for any constant $c > 0$. For each query it takes the majority of $t = O(d \log n)$ uniform random consistent halfspaces from $C$. In the next chapter we will show that

$$t = O(d \min(\log d, \log n))$$

calls to the $\mathrm{RCH}_C$-oracle suffices. Notice that, for large $n$ the number of calls is $O(d \log d)$. This significantly improves the number of calls to the $\mathrm{RCH}_C$-oracle. In particular, for constant dimension, the number of calls to the oracle is $O(1)$. Unfortunately, we do not know if the $\mathrm{RCH}_C$-oracle can be simulated in polynomial time and therefore this algorithm will not give a polynomial time learning algorithm.

The first (exponential time) learning algorithm for Halfspaces was the Perceptron learning algorithm PLA [R62] in the Exact (and the Online) learning model. The algorithm asks equivalence query with (initially any) hypothesis $h_u(x) = [u^T x \geq 0]$. For a positive counterexample $(a, 1)$ it updates the hypothesis to $h_{u+a}$ and for a negative counterexample it updates the hypothesis to $h_{u-a}$.

The equivalence query complexity of this algorithm is known to be

$$\frac{\|w\|^2 \delta_{max}}{\delta_{min}^2}$$

where

$$\delta_{min} = \min_{x \in X} |w^T x| \quad \text{and} \quad \delta_{max} = \max_{x \in X} \|x\|^2.$$

and for $\mathrm{HS}_2^d$ the above query complexity is less than (see [M94])

$$d^{2+d/2}.$$

Therefore the running time of PLA is exponential.

Littlestone [L88] gave another algorithm (Winnow 2) for learning Halfspaces. It is known from [S02] that Winnow 2 learning algorithm runs in exponential time.

The first polynomial time learning algorithm is given by Maass and Turan [MT94]. They show that there is an Exact learning algorithm for $\mathrm{HS}_n^d$ that runs in polynomial time and asks

$$O(d^2(\log d + \log n))$$

equivalence queries with hypotheses that are halfspaces. Using recent results in linear programming we show that this algorithm uses

$$1.512 \cdot d^2 \left( \log n + \frac{\log d}{2} \right)$$

equivalence queries using $O(d)$ calls to the RCH-oracle in each trial.

In this thesis we use a different approach and achieve a learning algorithm that uses, for any constant $c > 0$,

$$(1+c) \cdot d^2 \left( \log n + \frac{\log d}{2} \right) \tag{2.1}$$

equivalence queries with hypotheses that are majority of halfspaces. Our algorithm uses $O(d \log d)$ calls to the RCH-oracle in each trial. Since the RCH-oracle can be simulated in polynomial time, our algorithm also runs in polynomial time.

In [MT94] Maass and Turan also gave a lower bound of

$$\binom{d}{2} \log n \leq \frac{1}{2} d^2 \log n. \tag{2.2}$$

on the number of equivalence queries needed to learn $\mathrm{HS}_n^d$ with any learning algorithm that has unlimited computational power and that can ask equivalence query with any hypothesis.

Then we study the query complexity of learning using the RCH -oracle when limited number of calls to this oracle is allowed in each trial. We first study the complexity of the learning algorithm when only one call to the RCH-oracle is allowed at each trial. We show that the query complexity of this algorithm is $2^{\Theta(d)} \log n$. This is polynomial when the dimension $d = O(\log \log n)$. We also give a lower bound $2^{\Omega(d)} + \Omega(d^2 \log n)$ for the query complexity which, in particular, shows that this algorithm does not run in polynomial time for $d = \omega(\log \log n)$.

We then show that any learning algorithm with a strategy that uses $o(\sqrt{d})$ calls to the RCH-oracles and build any convex combination of the halfspaces received by the oracle for the equivalence query will have an exponential (in $d$) query complexity. This with (2.2) shows that any learning algorithm for learning Halfspaces with the RCH-oracle has to use the oracle at least $d^{2.5} \log n$ times.

# Chapter 3

# Learning With Halving Algorithm

In this chapter we will study the query complexity of the standard randomized Halving algorithm. First we present the standard deterministic Halving algorithm for learning any class $C$. We will need the following:

**Definition 1.** *For a set $F$ of boolean functions $f : X \to \{0,1\}$ we define the function*

$$Maj(F)(x) = \begin{cases} 1 & \Pr_{f \in_U F}[f(x) = 1] \geq \frac{1}{2} \\ 0 & otherwise \end{cases}$$

We say that a boolean function $f : X \to \{0,1\}$ is *consistent* with a set of examples $S \subset X \times \{0,1\}$ if $f(x) = b$ for every $(x,b) \in S$.

Now we present the algorithm:

Algorithm **Halv**

1. $\ell \leftarrow 1$, $S \leftarrow \emptyset$.
2. Let $F_\ell$ be the set of all functions in $C$ consistent with $S$.
3. Ask EQ$(Maj(F_\ell)) \to b_\ell$.
4. If $b_\ell$="Yes" then output$(Maj(F_\ell))$
5. else $S \leftarrow S \cup \{(b_\ell, \overline{Maj(F_\ell)(b_\ell)})\}$
6. $\ell \leftarrow \ell + 1$
7. Goto 2

Figure 3.1: The Deterministic Halving Algorithm.

To analyse the number of equivalence queries made by the algorithm observe that $|F_\ell| \geq 1$ since at least the target function $f$ is in $F_\ell$. Note that any counterexample $b_\ell$ recieved will eliminate at least half of the functions in $F_\ell$. That is $|F_{\ell+1}| \leq \frac{|F_\ell|}{2}$. This implies that the algorithm will find the target function after at most $\log |C|$ equivalence queries. The problem with this algorithm is that the hypothesis $Maj(F_\ell)$ proposed may be a majority of exponential number of functions.

The randomized Halving algorithm [BC+96] solves this problem by asking equivalence query with the majority of polynomial number of randomly uniformly chosen functions from $F_l$. See algorithm in Figure 3.2.

Algorithm **RanHalv**

1. $S \leftarrow \emptyset$.
2. Choose $m = \Theta(d \log n)$ uniform random functions $F = \{f_{w_1,t_1}, \ldots, f_{w_m,t_m}\}$ consistent with $S$, using the $\mathrm{RCH}_C$-oracle.
3. Ask $\mathrm{EQ}(Maj(F)) \rightarrow b$.
4. If $b$="Yes" then output$(Maj(F))$
5. else $S \leftarrow S \cup \{(b, \overline{Maj(F)(b)})\}$
6. Goto 2

Figure 3.2: Randomized Halving using the $\mathrm{RCH}_C$-oracle.

The randomized Halving algorithm asks on average $(1+c)d(d+1)\log n$ equivalence queries for any constant $c > 0$. For each query it takes the majority of $O(d \log n)$ uniform random consistent halfspaces from $C$.
In section 3.2 we will show that $O(d \min(\log d, \log n))$ calls to the $\mathrm{RCH}_C$-oracle suffices. Notice that, for large $n$ the number of calls is $O(d \log d)$. This significantly improves the number of calls to the $\mathrm{RCH}_C$-oracle. In particular, for constant dimension, the number of calls to the oracle is $O(1)$.

## 3.1 The Dual Concept Class

Let $C$ be a concept class of functions $f : X \rightarrow \{0, 1\}$. We define the *dual class* $C^\perp$ of $C$, the set of all boolean functions $F_x : C \rightarrow \{0, 1\}$ where, for $x \in X$, we have $F_x(f) = f(x)$

for every $f \in C$. The *dual VC-dimension* of $C$, $\mathrm{VCdim}^{\perp}(C)$ is defined to be the VC-dimension of the dual class of $C$, i.e., $\mathrm{VCdim}^{\perp}(C) = \mathrm{VCdim}(C^{\perp})$. It is convenient to think of a class $C$ as a matrix $M$. Each row of $M$ corresponds to a function $f \in C$ and each of its columns corresponds to a function $F_x \in C^{\perp}$. The class $C^{\perp}$ is the class represented by the transposed matrix $M^T$.
Notice that in this sense $\mathrm{VCdim}(C)$ is the maximal number $d$ of columns in which all the $2^d$ combinations $\{0,1\}^d$ appear.

The connection between the VC-dimension of $C$ and its dual is given in the following

**Lemma 5.** *[BBK97] We have*

$$\lfloor \log VCdim(C) \rfloor \leq VCdim^{\perp}(C) \leq 2^{VCdim(C)+1}.$$

**Proof.** Denote $d = \mathrm{VCdim}(C)$. That is, the matrix $M$ corresponding to $C$ has a set $B$ of $d$ columns in which all the $2^d$ combinations $\{0,1\}^d$ appear. Assume for convenience that $d = 2^k$. We now show that $M$ contains $\log d = k$ rows in which all $2^{\log d} = d$ combinations $\{0,1\}^{\log d}$ appear. This implies the claim.

Enumerate the $d$ columns in $B$ as $0, 1, \ldots, d-1$, and define $k$ vectors $v_i$ $(i = 1, \ldots, k)$ as follows: $v_{ij}$ is the $i$-th bit in the binary representation of the number $j-1$. Since the $d$ columns contain all $2^d$ combinations then in particular, there are $k$ rows which contain (in the entries corresponding to $B$) the $k$ vectors $v_1, \ldots, v_k$. This implies that in this $k$ rows all the $2^k = d$ combinations $\{0,1\}^d$ appear, as needed.□

Since the dual of $\mathrm{HS}_n^d$ is subset of $\mathrm{HS}^{d+1}$, we have

$$\mathrm{VCdim}^{\perp}(\mathrm{HS}_n^d) \leq d + 2.$$

We write $g =_{\eta} Maj(C)$ if $g(x) = Maj(C)(x)$ for all points $x$ that satisfies

$$\Delta(x) \stackrel{def}{=} |\Pr_f[f(x) = 1] - \Pr_f[f(x) = 0]| \geq \eta.$$

The latter is equivalent to

$$\Pr_f[f(x) = 1] \geq \frac{1}{2} + \frac{\eta}{2} \text{ or } \Pr_f[f(x) = 1] \leq \frac{1}{2} - \frac{\eta}{2}.$$

We now show

**Lemma 6.** *Let $f_1, \ldots, f_m$ be $m$ independently uniform random functions from $C$ where*

$$m = \frac{2}{\eta^2}\left(\ln|X| + \ln\frac{2}{\delta}\right).$$

*Then with probability at least $1-\delta$ we have $Maj(f_1, \ldots, f_m) =_{\eta} Maj(C)$.*

**Proof.** Consider the set

$$W = \{x \mid \Delta(x) \geq \eta\}.$$

Let the domain be $\mathcal{X} = C$, the concept class be $\mathcal{C} = \{F_x | x \in W\}$.

Consider the sample $S = \{f_1, \ldots, f_m\}$.

$$\begin{aligned}
\Pr[Maj(S) \neq_\eta Maj(C)] &= \Pr[(\exists x \in W)\ Maj(f_1(x), \ldots, f_m(x)) \neq Maj(C)(x)] \\
&\leq \Pr[(\exists x \in W)\ |E_{f\in\mathcal{X}}[f(x)] - E_{f\in S}[f(x)]| \geq \eta/2] \\
&= \Pr[(\exists F_x \in \mathcal{C})\ |E_{f\in\mathcal{X}}[F_x(f)] - E_{f\in S}[F_x(f)]| \geq \eta/2] \\
\textit{by Lemma 2} &\leq \delta. \square
\end{aligned}$$

Notice that $m$ is infinite when $X$ is infinite. In the next lemma we show that the sample is finite when the dual dimension is finite.

**Lemma 7.** *Let $f_1, \ldots, f_m$ be $m$ independently uniform random functions from $C$ where*

$$m = \frac{c_{VC}}{\eta^2}\left(VCdim^{\perp}(C) \log \frac{VCdim^{\perp}(C)}{\eta} + \ln \frac{1}{\delta}\right).$$

*Then with probability at least $1 - \delta$ we have $Maj(f_1, \ldots, f_m) =_\eta Maj(C)$.*

**Proof.** We use the same proof as in Lemma 6 with Lemma 3.$\square$

## 3.2 The Randomized Halving Algorithm

In this section we prove that it is sufficient for the randomized halving algorithm to take the majority of $O(d \min(\log d, \log n))$ uniform random consistent halfspaces for each *equivalence query*.

First we prove a more general result.

**Theorem 8.** *Let $C$ be a concept class. The randomized Halving algorithm for $C$ asks on average*

$$(1 + c) \log |C|$$

*equivalence queries for any constant $c$, where for each query it takes the majority of*

$$m = O(\min(\log |X|,\ VCdim^{\perp}(C) \log VCdim^{\perp}(C)))$$

*uniform random consistent functions from $C$.*

**Proof of Theorem 8** We will show that if we choose

$$m = \min\left(\frac{2}{\eta^2}\left(\ln |X| + \ln \frac{2}{\delta}\right), \frac{c_{VC}}{\eta^2}\left(\mathrm{VCdim}^{\perp}(C) \log \frac{\mathrm{VCdim}^{\perp}(C)}{\eta} + \ln \frac{1}{\delta}\right)\right)$$

uniform random functions $f_1, \ldots, f_m$ from $C$ then with probability at least $1-\delta$ the equivalence query with $Maj(f_1, \ldots, f_m)$ returns a counterexample that eliminates at least $1/2 - \eta/2$ of the elements of $C$. This implies that on average the number of equivalence queries is

$$\frac{\log |C|}{(1-\delta)(1-\log(1+\eta))} \leq (1+c)\log |C|$$

for some constants $\eta$ and $\delta$. Then for constant $\eta$ and $\delta$ we have

$$m = O(\min(\log |X|, \mathrm{VCdim}^{\perp}(C) \log \mathrm{VCdim}^{\perp}(C)).$$

To prove the above, let $(x_0, y_0)$ be a counterexample received by the equivalence query oracle. We have two cases: If $\Delta(x_0) > \eta$ then by Lemma 6, 7 with probability at least $1-\delta$, $Maj(f_1(x_0), \ldots, f_m(x_0)) = Maj(C)(x_0)$ and then (as in the Halving algorithm) this counterexample eliminates at least half of the functions in $C$. If $\Delta(x_0) < \eta$ then by the definition of $\Delta$ at least $1/2 - \eta/2$ of the functions in $C$ are not equal to $y_0$ on $x_0$ and therefore this counterexample eliminates at least $1/2 - \eta/2$ of the elements of $C$.□

Now we prove our main theorem:

**Theorem 9.** *The randomized halving algorithm learns the class $HS_n^d$ with on average,*

$$(1+c)d(d+1)\log n$$

*equivalence queries for any constant $c > 0$ where for each query it takes the majority of $t = O(d \min(\log d, \log n))$ uniform random consistent halfspaces (by calling $RCH_C$).*

**Proof of Theorem 9** Applying theorem 8: The number of equivalence queries for learning $C = HS_n^d$ is on average

$$\begin{aligned}
(1+c)\log |C| &= (1+c)\log |HS_n^d| \\
\textit{by lemma 4} \quad &\leq (1+c)\log(n^{d(d+1)}) \\
&= (1+c)d(d+1)\log n
\end{aligned}$$

And each query takes the majority of $m$ uniform random consistent functions from $HS_n^d$, for

$$\begin{aligned}
m &= O(\min(\log |X|, \mathrm{VCdim}^{\perp}(C) \log \mathrm{VCdim}^{\perp}(C))) \\
&= O(\min(\log |n^d|, \mathrm{VCdim}^{\perp}(HS_n^d) \log \mathrm{VCdim}^{\perp}(HS_n^d))) \\
&= O(\min(d \log n, d \log d))
\end{aligned}$$

since $\mathrm{VCdim}^{\perp}(HS_n^d) \leq d+2$. □

# Chapter 4

# Polynomial time Learning Halfspaces

In this chapter we introduce Mass and Turan algorithm using the new linear programming results from the literature. We give the analysis of the complexity of the algorithm, then give our new algorithm and analyse its complexity.

## 4.1 Preliminaries

To analyse Mass and Turan algorithm and prove our result, we will need the following definitions and lemmas:

### 4.1.1 Bounding the weights of halfspaces

**Lemma 10.** *([P94]) For every $f \in HS_2^d$ there is $w \in \mathbb{Z}^d$ and $t \in \mathbb{Z}$ such that*

$$|w_i| \leq \frac{(d+1)^{(d+1)/2}}{2^d} = d^{\frac{d}{2} - \frac{d}{\log d} + o(1)}$$

*and $f_{w,t} = f$.*

Hastad [H94] showed that this bound is tight for $d$ that is a power of 2. That is, for any integer $d = 2^k$ for some integer $k$ there is a Threshold with

$$|w_i| \geq d^{\frac{d}{2} - \frac{d}{\log d}}.$$

For $\mathrm{HS}_n^d$ and $d$ that is power of 2, Hastad achieves the bound

$$|w_i| \geq (n-1)^d d^{\frac{d}{2} - \frac{d}{\log d}}.$$

On the other hand, we have

**Lemma 11.** *([MT94]) For every $f \in HS_n^d$ there is $w \in \mathbb{Z}^d$ and $t \in \mathbb{Z}$ such that $|w_i| \leq 3n^{2(d+1)}(2d+2)^{d+2}$ and $f = f_{w,t}$.*

The bound achieved in [MT94] is $2(2d+2)!n^{2d+2}(2d+3)$. The bound in the lemma can be achieved by replacing the bound $(2d+2)!n^{2d+2}$ for the determinant of $(2d+2)\times(2d+2)$ matrices with entries from $[-n,n]=-n,-n+1,\ldots,n-1,n$ with the bound $(2d+2)^{d+1}n^{2d+2}$ that can be obtained from the following:

**Lemma 12. Hadamard inequality for matrices.** *For any matrix $A=(a_{i,j})\in\Re^{r\times r}$ we have*

$$|det(A)|\leq\prod_{i=1}^{r}\sqrt{\sum_{j=1}^{r}a_{i,j}^2}.$$

Using a similar technique as in [P94] we prove the following

**Lemma 13.** *For every $f\in HS_n^d$ there is $w\in\mathbb{Z}^d$ and $t\in\mathbb{Z}$ such that*

$$|w_i|\leq\frac{(n-1)^{d-1}(d+1)^{(d+1)/2}}{2^d},$$

$$|t|<(n-1)\sum_i|w_i|=\frac{(n-1)^d d(d+1)^{(d+1)/2}}{2^d}$$

*and $f=f_{w,t}$.*

**Proof.** Let $f=[\hat{w}^Tx\geq\hat{t}]\in\mathrm{HS}_n^d$. We will assume that $f$ depends on all the variables. Otherwise, we restrict $f$ to the relevant variables and apply the same proof on smaller dimension $d$.

Let

$$\mu=\min_{f(x)=0}(\hat{t}-\hat{w}^Tx).$$

Then $\mu>0$ and for $\tilde{t}=\hat{t}/\mu$ and $\tilde{w}=\hat{w}/\mu$ we have: $f(x)=1$ if and only if $\tilde{w}^Tx\geq\tilde{t}$ and $f(x)=0$ if and only if $\tilde{w}^Tx\leq\tilde{t}-1$.

Consider all the $n^d$ inequalities on $(w_1,\ldots,w_d,t)$, (one inequality for each $x\in[n]^d$),

$$\begin{cases} w^Tx-t\geq 0 & \text{for } f(x)=1\\ w^Tx-t\leq -1 & \text{for } f(x)=0\end{cases}\tag{4.1}$$

This defines a convex polytope $P$ in $\Re^{d+1}$. All the points of this polytope including the boundary corresponds to halfspaces that equivalent to $f$. This polytope is nonempty since $(\tilde{w},\tilde{t})$ satisfies all inequalities and therefore it is a point in $P$. Consider a point $(w,t)$ on the boundary $\partial P$ of this polytope. This point must satisfy at least one equation in (4.1), $w^Tx^{(1)}-t=\xi_1$ for some $x^{(1)}\in[n]^d$ and $\xi_1\in\{0,-1\}$. We choose $(w',t')$ in $\partial P$ that satisfies maximal number of equations. Suppose

$$w^T x^{(i)} - t = \xi_i, \quad i = 1, \ldots, m, \tag{4.2}$$

are all the equations that $(w', t')$ satisfies.

Let $H = Span\{(x^{(i)}, -1) | i = 1, \ldots, m\}$. We now show that $\dim H = d + 1$. Consider the set $S$ of all solutions of the linear equations in (4.2). We have two cases

**Case I.** $S \subset P$. If $S \subset P$ then any solution of (4.2) represents $f$. If $\dim H < d$ then there is a solution of the equations in (4.2) with one of the weights $w_j$ is equal to 0. This implies that $f$ is independent of $x_j$, and we get a contradiction . If $\dim H = d$ then either we can find a solution with $w_j = 0$ for some $j$ and then as before we get a contradiction, or, there is one solution for $w$ with nonzero entries and then $t$ is arbitrary. This case cannot happen since different values of $t$ can give different functions. Obviously, $\dim H < d + 2$ since the elements of $H$ are of dimension $d + 1$. Therefore, $\dim H = d + 1$.

**Case II.** $S \nsubseteq P$. In this case we choose a point $z \in S \backslash P$. Consider the line that passes through $z$ and $(w', t')$. Since $(w', t') \in S$ and $z \in S$ every point on this line is in $S$. Since $(w', t') \in P$ and $z \notin P$ this line intersect the boundary of $P$ in some pont $z'$. This point satisfies all the equations in (4.2) and at least one more. This is a contradiction because $m$ is maximal. Therefore this case cannot happen.

Therefore, $\dim H = d + 1$ and there are $d + 1$ equations

$$w^T x^{(i_j)} - t = \xi_{i_j}, \quad j = 1, \ldots, d + 1,$$

where $\{(x^{(i_j)}, -1) | j = 1, \ldots, d + 1\}$ are linearly independent. These are of the form

$$\begin{cases} w_1 x_{1,1} + w_2 x_{1,2} + \cdots + w_d x_{1,d} - t & = \xi'_1 \\ w_1 x_{2,1} + w_2 x_{2,2} + \cdots + w_d x_{2,d} - t & = \xi'_2 \\ \quad \vdots & \\ w_1 x_{d+1,1} + w_2 x_{d+1,2} + \cdots + w_d x_{d+1,d} - t & = \xi'_d \end{cases}$$

where $x^{(i_j)} = (x_{j,1}, \ldots, x_{j,d})$ and $\xi'_j = \xi_{i_j}$. By Cramer's rule, the solution is $w_i = \Delta_i / det(x_{i,j})$ and $t = \Delta_{d+1} / det(x_{i,j})$ where for $i < d + 1$,

$$\Delta_i = \begin{vmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,i-1} & \xi'_1 & x_{1,i+1} & \cdots & x_{1,d} & -1 \\ x_{2,1} & x_{2,2} & \cdots & x_{2,i-1} & \xi'_2 & x_{2,i+1} & \cdots & x_{2,d} & -1 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots \\ x_{d+1,1} & x_{d+1,2} & \cdots & x_{d+1,i-1} & \xi'_{d+1} & x_{d+1,i+1} & \cdots & x_{d+1,d} & -1 \end{vmatrix}.$$

Notice here that each $x_{i,j} \in \{0, \ldots, n-1\}$. We divide each $x_{i,j}$ by $n-1$ and then multiply each column $j \neq i$ by 2 and add to it the last column. Then we multiply column $i$ by $-2$ and add to it the last column. This gives a matrix $\Delta'_i$ with entries in $[-1, 1]$ where

$$|\Delta_i| = \frac{(n-1)^{d-1}}{2^d} |\Delta'_i|.$$

Now since by Hadamard inequality $|\Delta'_i| \leq (d+1)^{(d+1)/2}$ the result follows.□

### 4.1.2 Dual Domain

Define a map $\phi : \Re^{d+1} \rightarrow \mathrm{HS}_X^d$ where $\phi(w,t) = f_{w,t}$. Notice that two different points $(w_1,t_1)$ and $(w_2,t_2)$ may map into the same halfspace, i.e., $\phi(w_1,t_1) = \phi(w_2,t_2)$. We call the domain $\Re^{d+1}$ in $\phi$ the *coefficients domain* or the *dual domain*. For $f \in \mathrm{HS}_X^d$, we call $\phi^{-1}(f) \subseteq \Re^{d+1}$ the polytope that corresponds to $f$ in the dual space. For a $d$-hypercube $U_R = [-R,R]^{d+1}$ we say that $U_R$ *covers* $\mathrm{HS}_X^d$ if for every $f_{w,t} \in \mathrm{HS}_X^d$ there is $(u,t') \in U_R$ such that $\phi(u) = f_{u,t'} \equiv f_{w,t}$. For a hypercube $U_R$ that covers $\mathrm{HS}_X^d$ we define

$$V_{min}(R) = \min_{f \in \mathrm{HS}_X^d} \mathrm{Vol}(U_R \cap \phi^{-1}(f)),$$

where Vol() is the volume in the $(d+1)$-dimensional space. This is the minimal volume of polytope in $U_R$ that corresponds to $f \in \mathrm{HS}_X^d$.

By Lemma 13 we can choose for $\mathrm{HS}_n^d$,

$$R = \frac{(n-1)^d d(d+1)^{(d+1)/2}}{2^d}. \tag{4.3}$$

**Lemma 14.** *For $R$ in (4.3) we have*

$$V_{min}(R) \geq \frac{1}{2^{d+1}(d(n-1))^d}.$$

**Proof.** In Lemma 13 we showed that for any $f \in HS_n^d$ there is $(w,t) \in U_R$ such that

$$\begin{cases} w^T x - t \geq 0 & \text{for } f(x) = 1 \\ w^T x - t \leq -1 & \text{for } f(x) = 0 \end{cases}. \tag{4.4}$$

Consider $V_i = [w_i - (1/(4d(n-1))), w_i + (1/(4d(n-1)))]$ and $T = (t - 3/4, t - 1/4)$. Let $v \in \Re^d$ where $v_i \in V_i$ and let $t' \in T$. If $w^t x - t \geq 0$ then

$$v^T x - t' > w^T x - \frac{1}{4d(n-1)} \sum_i x_i - t + \frac{1}{4} \geq w^T x - t \geq 0,$$

and if $w^t x - t \leq -1$ then

$$v^T x - t' < w^T x + \frac{1}{4d(n-1)} \sum_i x_i - t + \frac{3}{4} \leq w^T x - t + 1 \leq 0.$$

Therefore, $f = f_{w,t} = f_{v,t'}$. Therefore, $V_1 \times \cdots \times V_d \times T \subseteq U_R \cap \phi^{-1}(f)$ and then

$$\mathrm{Vol}(U_R \cap \phi^{-1}(f_{w,t})) \geq \frac{1}{2^{d+1}(d(n-1))^d}. \square$$

### 4.1.3 Convex Set

A subset $K \subset \Re^d$ is called *convex set* if for every $x, y \in K$ and every $0 \leq \lambda \leq 1$ we have $\lambda x + (1-\lambda)y \in K$, i.e., the line connecting the two points $x$ and $y$ is in $K$.

For a set of points $S = \{x^{(1)}, \ldots, x^{(k)}\}$ the *convex closure* (*convex hull*) of $S$, $Cl(S)$, is the set of all points that can be written as $\lambda_1 x^{(1)} + \cdots + \lambda_k x^{(k)}$, where $\lambda_i \geq 0$ for each $i$, and $\lambda_1 + \cdots + \lambda_k = 1$. It is known that the convex closure of any set is a convex set.

An *affine transformation* is $\phi_{A,B} : x \mapsto Ax + B$ where $A$ and $B$ are $d \times d$ matrices and $A$ is nonsingular ($det(A) \neq 0$). The affine transformation changes the volume of any subset by the same factor $\alpha = |det(A)|$. In particular,

$$\text{Vol}(\phi_{A,B}(K)) = |det(A)|\text{Vol}(K).$$

A *centroid* (*center of gravity*) of a closed and bounded convex set $K$ is defined as

$$\frac{1}{\text{Vol}(K)} \int_{x \in K} x \; dx.$$

The *covariance matrix* of a convex set is a $d \times d$ matrix $M$ where

$$M_{i,j} = \frac{1}{\text{Vol}(K)} \int_{x \in K} x_i x_j \; dx.$$

Notice that for a uniform random point $x$ in $K$ the centroid is $E_K[x]$ and the covariance matrix is $E_K[xx^T]$ . A convex set $K$ in $\Re^d$ is said to be in *isotropic position* if:

1. The centroid of $K$ is the origin,
$$E_K[x] = 0$$
2. The covariance matrix of $K$ is the identity, i.e.,
$$E_K[xx^T] = I.$$

Condition (2) is equivalent to: For any unit vector $u$, ($\|u\| = 1$),

$$E_K[(u^T x)^2] = \frac{1}{\text{Vol}(K)} \int_{x \in K} (u^T x)^2 dx = 1.$$

In other words, the average squared length in any direction is 1. In particular, this implies

$$E_K[\|x\|^2] = \frac{1}{\text{Vol}(K)} \int_{x \in K} \|x\|^2 dx = d. \tag{4.5}$$

For any full-dimensional ($\text{Vol}(K) \neq 0$ in $\Re^d$) convex set, there exists an affine transformation that puts the set in isotropic position.

It is known from [G60]

**Lemma 15.** *For a convex set $K$, any cut through its centroid by a halfspace has at least $1/e$ of the volume on each side.*

In [BV02], Bertsimas and Vempala show

**Lemma 16.** *Let $K$ be a convex set in isotropic position and $z$ be a point at distance $t$ from its centroid. Then any halfspace containing $z$ also contains at least $\frac{1}{e} - t$ of the volume of $K$.*

### 4.1.4 A Ball

An *d-sphere* $B_R$ (also called *d-hypersphere*) is the set of $d$-tuples $(x_1, \ldots, x_d)$ such that $x_1^2 + x_2^2 + \cdots + x_d^2 = R^2$ where $R$ is the *radius* of the $d$-sphere. A *d-ball* is the interior of the $d$-sphere, that is, the set of $d$-tuples $(x_1, \ldots, x_d)$ such that $x_1^2 + x_2^2 + \cdots + x_d^2 \leq R^2$. Denote $S_d(R)$ the surface area of a $d$-sphere and $V_d(R)$ the volume of a $d$-ball. Then

$$V_d(R) = \frac{S_d(1)R^d}{d},$$

and

$$S_d \stackrel{\Delta}{=} S_d(1) = \frac{2\pi^{d/2}}{\Gamma\left(\frac{d}{2}\right)}$$

where $\Gamma$ is the Gamma function. Special forms of $\Gamma(d/2)$ for integer $d$ allow the above expression to be written as

$$S_d = \begin{cases} \frac{2^{(d+1)/2}\pi^{(d-1)/2}}{(d-2)(d-4)\cdots 3\cdot 1} & \text{for } d \text{ odd} \\ \frac{2\pi^{d/2}}{\left(\frac{d}{2}-1\right)!} & \text{for } d \text{ even} \end{cases}. \tag{4.6}$$

We now show

**Lemma 17.** *An isotropic ball $K$ is a ball with radius $R = \sqrt{d+2}$.*

**Proof.** We use (4.5). We have $E_K[\|x\|^2] = d$ and therefore

$$S_d R^d = d\text{Vol}(K) = \int_{x \in K} \|x\|^2 dx = \int_0^R (S_d r^{d-1}) r^2 dr = S_d \frac{R^{d+2}}{d+2}.$$

This implies the result.□

## 4.2 Mass-Turan Algorithm

Now we describe and analyse Mass and Turan algorithm [MT94] for exact learning halfspaces (Figure 4.1).

By Lemma 13 we may assume that the target function $f_{w,t}$ satisfies: $w \in \mathbb{Z}^d$ and $t \in \mathbb{Z}$,

$$|w_i| \leq \frac{(n-1)^{d-1}(d+1)^{(d+1)/2}}{2^d},$$

and

$$|t| < (n-1)\sum_i |w_i| = \frac{(n-1)^d d(d+1)^{(d+1)/2}}{2^d}.$$

Those inequalities define a domain for $(w,t)$ in the dual domain $\Re^{d+1}$. Denote this domain by $W_0$. Also, each counterexample $(x_i, f(x_i))$, $i = 1,\ldots,t$ received by an equivalence query defines a Halfspace in the dual domain $\Re^{d+1}$,

$$\begin{cases} w^T x_i \geq 0 & \text{for } f(x_i) = 1 \\ w^T x_i < 0 & \text{for } f(x_i) = 0 \end{cases}$$

Let $S$ be the set of counterexamples received from the first $\ell$ equivalence queries. Suppose $W_\ell$ is the domain in the dual domain defined by $S = \{(x_i, f(x_i)) \mid i = 1,\ldots,\ell\}$ and $W_0$. Any hypothesis $f_{w',t'}$ that is chosen for the $\ell+1$ equivalence query is a point $(w',t')$ in the dual domain. Any counterexample $p = (x_{\ell+1}, f(x_{\ell+1}))$ for $f_{w',t'}$ defines a new halfspace in the dual domain that does not contain the point $(w',t')$. If the volume of any cut through the point $(w',t')$ has at least $1-\alpha$ of the volume of $W_\ell$ then any counterexample will define a new domain $W_{\ell+1}$ such that $\mathrm{Vol}(W_{\ell+1}) \leq \alpha \mathrm{Vol}(W_\ell)$. By Lemma 14 if the volume $\mathrm{Vol}(W_\ell)$ is less than

$$V_{min} \triangleq \frac{1}{2^{d+1}(d(n-1))^d},$$

then any point $(w',t')$ in the domain gives a unique halfspace. Since

$$Vol(W_0) = \frac{(n-1)^{d^2} d(d+1)^{\frac{(d+1)^2}{2}}}{2^{(d-1)(d+1)}}$$

and

$$Vol(W_{\ell+1}) \leq \alpha Vol(W_\ell),$$

the number of equivalence queries in this algorithm is

$$\begin{aligned} \frac{\log \frac{Vol(W_0)}{V_{min}}}{\log \frac{1}{\alpha}} &= c_\alpha d^2 \left(\log n + \frac{\log d}{2}\right) + O(d(\log n + \log d)) \\ &= O(d^2(\log n + \log d)) \end{aligned} \tag{4.7}$$

where

$$c_\alpha = \frac{1}{\log \frac{1}{\alpha}}.$$

This algorithm is equivalent to the algorithms that finds a point in a convex set using a separation oracle. Today there are many methods for solving this problem [V96, BV02, DV04]. In [BV02] it is shown that there is a polynomial time (in $1/t$) algorithm that with a probability that is exponentially close to 1 finds a point that is at distance $t$ from the centroid. By Lemma 16, we then have $\alpha = 1 - e^{-1}$ and based on this algorithm we get $c_\alpha = 1.512$. Therefore the query complexity is at least

$$1.512 \cdot d^2 \left( \log n + \frac{\log d}{2} \right) + O(d(\log n + \log d)).$$

Following [BV02], the algorithm uniformly randomly chooses $N = O(d)$ points $p'_i = (w'_i, t'_i)$ in the domain $W_\ell$ and takes the average point $p = \sum_i p'_i$ as the next hypothesis in the equivalence query. This is equivalent to using the oracle RCH-oracle, $O(d)$ times.

We use a different approach and achieve a learning algorithm that uses on average

$$(1 + c) \cdot d^2 \left( \log n + \frac{\log d}{2} \right)$$

equivalence queries for any constant $c > 0$ using $O(d \log d)$ calls to the RCH-oracle. Since RCH can be simulated in polynomial time, our algorithm runs in polynomial time.

## 4.3 Our Algorithm

Our algorithm simply uses the Randomized Halving algorithm but with the RCH-oracle instead of the $\mathrm{RCH}_C$-oracle.

We first show the following

**Lemma 18.** *Let $W \subseteq \Re^{d+1}$ be any domain of volume $V$ and consider all the halfspaces over $W$, $HS_W^{d+1}$. Let $S$ be a set of $m$ uniform random points in $W$ where*

$$m = \frac{c_{VC}}{\epsilon^2} \left( (d+2) \log \frac{d+2}{\epsilon} + \log \frac{1}{\delta} \right).$$

*Then with probability at least $1 - \delta$ any cut (by a halfspace) in $W$ that contains at most $m/2$ points is of volume at most $(1/2 + \epsilon)V$.*

**Proof.** This Lemma follows from the $\epsilon$-Sample in Lemma 3. Consider $X = W$ and $C = \mathrm{HS}_W^{d+1}$. Then for a halfspace $F$, $E_X[F(x)]$ is equal to the volume $V_{cut}$ of the cut (of this halfspace) over the volume of $W$ and for the desired cut $E_S[F(x)] < 1/2$. Now since VC-dimension of $\mathrm{HS}_W^{d+1}$ is at most $d + 2$ we have with probability at least $1 - \delta$,

$$|V_{cut}/V - 1/2| < \epsilon.$$

This follows the result.□

The algorithm **RanHalv** in Figure 4.2 is the randomized halving algorithm but instead of using the $\mathrm{RCH}_C$-oracle it uses the RCH-oracle. We prove the following

**Theorem 19.** *Algorithm* **RanHalv** *learns the class* $HS_n^d$ *with, on average,*

$$(1+c)\cdot d^2\left(\log n+\frac{\log d}{2}\right)$$

*equivalence queries for any constant* $c>0$ *using* $m=O(d\log d)$ *calls to the RCH-oracle in each trial. Since the RCH-oracle can be simulated in polynomial time, the algorithm runs in polynomial time.*

**Proof.** Let $f_{w_1,t_1},\ldots,f_{w_m,t_m}$ be

$$m=\frac{c_{VC}}{\epsilon^2}\left((d+2)\log\frac{d+2}{\epsilon}+\log\frac{1}{\delta}\right)$$

uniform random consistent functions in $\mathrm{HS}_n^d$. Let $W_\ell\subset\Re^{d+1}$ be as defined in section 4.2. Then the points $L=\{(w_1,t_1),\ldots,(w_m,t_m)\}$ are random points in $W_\ell$. The counterexample $(b,y)$ for $Maj(f_{w_1,t_1},\ldots,f_{w_m,t_m})$ must be a counterexample for at least $m/2$ function in $F=\{f_{w_1,t_1},\ldots,f_{w_m,t_m}\}$. This means that the cut $W_{\ell+1}$ in $W_\ell$ caused by this counterexample that contains $(w,t)$, contains at most $m/2$ points from $L$. By Lemma 18 with probability at least $1-\delta$ this cut satisfies

$$\mathrm{Vol}(W_{\ell+1})\le\left(\frac{1}{2}+\epsilon\right)\mathrm{Vol}(W_\ell).$$

Therefore, the number of equivalence queries is, on average,

$$\frac{1}{1-\delta}\frac{\log\frac{Vol(W_0)}{Vol(V_{min})}}{\log\frac{2}{1+2\epsilon}}\le(1+c)\cdot d^2\left(\log n+\frac{\log d}{2}\right)$$

for sufficient small constants $\epsilon$ and $\delta$.□

Algorithm **Mass and Turan**

1. $S \leftarrow \emptyset$.
2. $W(S) = W_0 \cap$ The domain in the dual domain defined by $S$.
3. Using one of [V96, BV02, DV04] techniques choose a point $(w, t) \in W(S)$.
4. Ask EQ$(f_{w,t}) \rightarrow b$.
5. If $b$="Yes" then output$(f_{w,t})$
6. else $S \leftarrow S \cup \{(b, \overline{f_{w,t}(b)})\}$
7. Goto 2

Figure 4.1: Mass and Turan algorithm.

Algorithm **RanHalv**

1. $S \leftarrow \emptyset$.
2. $W(S) = W_0 \cap$ The domain in the dual domain defined by $S$.
3. Choose $m = \frac{c_{VC}}{\epsilon^2}\left((d+2)\log\frac{d+2}{\epsilon} + \log\frac{1}{\delta}\right)$ uniform random functions $F = \{f_{w_1,t_1}, \ldots, f_{w_m,t_m}\}$ using the RCH-oracle on the domain $W(S)$.
4. Ask EQ$(Maj(F)) \rightarrow b$.
5. If $b$="Yes" then output$(Maj(F))$
6. else $S \leftarrow S \cup \{(b, \overline{Maj(F)(b)})\}$
7. Goto 2

Figure 4.2: Randomized Halving using the RCH-oracle.

# Chapter 5

# Learning with Other Strategies

In this chapter we study the query complexity of exact learning when limited number of calls to the RCH-oracle is allowed in each trial.

We show that if only one call to RCH-oracle is allowed in each trial then the query complexity of the learning algorithm is $2^{\Theta(d)} \log n$. Since the RCH-oracle can be simulated in polynomial time this learning algorithm runs in polynomial time for $d = O(\log \log n)$. We then give a tight lower bound $2^{\Omega(d)} + \Omega(d^2 \log n)$. This proves that this learning algorithm does not run in polynomial time for $d = \omega(\log \log n)$.

Then we show that any "reasonable" strategy must use the RCH-oracle at least $\Omega(\sqrt{d})$ times in each trial.

## 5.1 Learning with Arbitrary Halfspace

We first show why choosing an arbitrary consistent halfspace for the equivalence query may give an exponential time learning algorithm. This motivates the use of the RCH-oracle and $\text{RCH}_C$-oracle.

We first show the following well known result (folklore):

**Lemma 20.** *Let $X \subset \Re^d$ be a finite or infinite countable set. There is a total order $(X, \preceq)$ such that for every $y \in X$ there is a Halfspace $f_{w,t}$ where $f_{w,t}(z) = 1$ if and only if $y \preceq z$.*

**Proof.** Consider all formal polynomials $p_{z,y}(\lambda) = (z_1 - y_1) + (z_2 - y_2)\lambda + \cdots (z_d - y_d)\lambda^{d-1}$ where $z, y \in X$. Consider the set of all the roots of $P_{z,y}$

$$A = \{\xi \in \Re \mid p_{z,y}(\xi) = 0 \text{ for some } \ z, y \in X\}.$$

Since each $p_{z,y}$ can have at most $d-1$ roots and $X$ is countable, the set $A$ is countable. Therefore there exists a real number $r \in \Re \backslash A$. This $r$ satisfies the property: If $p_{z,y}(r) = 0$

then $z = y$. Define the order $\prec$ on $X$ where $z \preceq y$ if $p_{z,0}(r) \leq p_{y,0}(r)$. We now show that $\preceq$ is a total order.

It is clear that $\preceq$ is reflexive, transitive and satisfies the comparability law. We show now that it is weak antisymmetry. Suppose $y \preceq z$ and $z \preceq y$. Then $p_{z,0}(r) = p_{y,0}(r)$ and $p_{z,y}(r) = p_{z,0}(r) - p_{y,0}(r) = 0$ which implies $y = z$. So $\prec$ is a total order.

Now consider a point $y \in X$ and define the halfspace $f_{w,t} = [p_{x,0}(r) \geq p_{y,0}(r)]$. Here $w = (1, r, r^2, \ldots, r^{d-1})$ and $t = p_{y,0}(r)$. Then $f_{w,t}(z) = 1$ iff $p_{z,0}(r) \geq p_{y,0}(r)$ iff $y \preceq z$.□

**Theorem 21.** *An Exact learner that uses an arbitrary consistent hypothesis for the equivalence query learns $HS_n^d$ with at least $n^d$ equivalence queries.*

**Proof.** Consider the total order $([n]^d, \preceq)$ defined in Lemma 20. Let $x_1 \prec x_2 \prec x_3 \cdots \prec x_{n^d}$ be the elements in $[n]^d$. Let $f_{w,t_i}$ be the halfspace that is 0 for all $x \preceq x_i$ and 1 for the other points. A learner that asks $\mathrm{EQ}(f_{w,t_i})$ at trial $i$ can receive the counterexample $x_{i+1}$. If the target function is the constant function 0 then this learner will ask $n^d$ equivalence queries.□

## 5.2 Learning with One Random Consistent Hypothesis

In this section we study learning $\mathrm{HS}_n^d$ when at each trial the learning algorithm makes one call to the RCH-oracle and uses the outcome for the equivalence query. We prove

**Theorem 22.** *A learning algorithm that uses at each trial a uniform random consistent hypothesis for the equivalence query has expected query and time complexity*

$$4^d poly(d) \log n.$$

*In particular, for constant dimensional space and $d = o(\log \log n)$ this algorithm runs in polynomial time.*

We then prove a lower bound

**Theorem 23.** *A learning algorithm that uses at each trial a uniform random consistent hypothesis for the equivalence query has expected query complexity at least*

$$\Omega(1.192293^d + d^2 \log n),$$

In section 5.2.1 we show that in any convex set the probability that a uniform random point is close to the centroid is $1/d^{O(d)}$. This gives a $O(d^d 4^d (\log n + \log d))$ time complexity algorithm which is sufficient for proving polynomial time complexity for constant dimensional space. Then a more rigorous analysis is given in section 5.2.2 to get

the result. We show that the expectation of the minimal volume of a cut of a uniform random point in a convex set of volume $V$ is at least $V/(e4^d)$. Then in section 5.2.3 we prove that for a ball of volume $V$ the expectation of the minimal volume of a cut of a uniform random point is at most $\alpha^d V$ for some constant $\alpha$. This will lead us to the lower bound.

### 5.2.1 Random Point in a Convex Set

In this subsection we give some properties of uniform random point in a convex set $K$.

We first find the probability that a uniform random point in a convex set $K$ is close to the centroid. We need the following

**Lemma 24.** *Any convex set in an isotropic position has volume at most*

$$S_d(d+2)^{d/2}\left(\frac{1}{2}+\frac{1}{d}\right).$$

*There is a convex set in an isotropic position with volume*

$$S_d(d+2)^{d/2}\frac{1}{d}.$$

**Proof.** Let $K$ be a convex set in an isotropic position. Let

$$K_R = \{x \in K \mid \|x\| \geq R\},$$

and $\overline{K_R} = K \backslash K_R$. Then by (4.5) and since $\overline{K_R} \subseteq B_R$,

$$\begin{aligned} R^2\mathrm{Vol}(K_R) &= \int_{K_R} R^2 dx \leq \int_{K_R} \|x\|^2 dx \leq \int_K \|x\|^2 dx \\ &= d\mathrm{Vol}(K) = d(\mathrm{Vol}(K_R) + \mathrm{Vol}(\overline{K_R})) \\ &\leq d\left(\mathrm{Vol}(K_R) + \frac{S_d R^d}{d}\right) \\ &= d\mathrm{Vol}(K_R) + S_d R^d. \end{aligned}$$

and therefore

$$\mathrm{Vol}(K_R) \leq \frac{S_d R^d}{R^2 - d}.$$

Now for $R = \sqrt{d+2}$, which minimizes the volume,

$$\begin{aligned} \mathrm{Vol}(K) &= \mathrm{Vol}(K_R) + \mathrm{Vol}(\overline{K_R}) \\ &\leq \frac{S_d R^d}{R^2 - d} + \frac{S_d R^d}{d} \\ &= S_d R^d \left(\frac{1}{R^2 - d} + \frac{1}{d}\right) \\ &= S_d(d+2)^{d/2}\left(\frac{1}{2}+\frac{1}{d}\right). \end{aligned}$$

On the other hand, an isotropic ball is a ball of radius $\sqrt{d+2}$. See Lemma 17. Such a ball has the required volume.□

This implies the following

**Lemma 25.** *Let $K$ be a convex set in isotropic position. For any $t < \frac{1}{e}$ the probability that a uniform random point $x$ is at distance at most $t$ from the centroid, that is $\|x\| \leq t$, is at least*

$$\frac{2}{d+2} \cdot \frac{t^d}{(d+2)^{d/2}}.$$

*On the other hand, for the isotropic ball this probability is*

$$\frac{t^d}{(d+2)^{d/2}}.$$

**Proof.** First notice that, by Lemma 16, $B_{1/e} \subset K$. Therefore, by Lemma 24, for a uniform random $x \in K$ and for $t < 1/e$ we have

$$\Pr_K[x \in B_t] = \frac{\mathrm{Vol}(B_t)}{\mathrm{Vol}(K)} \geq \frac{\frac{S_d t^d}{d}}{S_d(d+2)^{d/2}\left(\frac{1}{2}+\frac{1}{d}\right)} = \frac{2t^d}{(d+2)^{d/2+1}}.$$

The other result follows from the fact that the volume of isotropic ball is $S_d(d+2)^{d/2}/d$.□

Now, using (4.7) and Lemma 16, for a constant $t < 1/e$, with probability at least

$$\frac{2t^d}{(d+2)^{d/2+1}},$$

$Vol(W_{\ell+1}) \leq \alpha \cdot Vol(W_\ell)$ for some constant $\alpha$ less than one. This proves that the expected number of equivalence queries is at most

$$d^{O(d)} \log n.$$

In the next subsection we give a better complexity bound.

### 5.2.2 Volume of a Minimal Random Point Cut

We are now interested in bounding the volume of a cut through a uniform random point by a halfspace in any convex set $K$.

We will prove the following

**Theorem 26.** *Let $\alpha < 1$. For any convex set $K$, with probability at least $\alpha^d$ any cut through a uniform random point by a halfspace has at least $(1-\alpha)^d e^{-1}$ of the volume in each side.*

*The expectation of the minimal volume of a cut through a uniform random point by a halfspace in any convex set of volume $V$ is at least*

$$\frac{V}{e4^d} = O(0.25^d V).$$

In the next subsection we prove

**Theorem 27.** *Let $K$ be a ball of volume $V$. The expectation of the minimal volume of a cut through a uniform random point by a halfspace is at most*

$$\Omega(0.75^d V).$$

To prove Theorem 26 we need the following

**Lemma 28.** *Let $K$ be a convex set in isotropic position. Let $l$ be a line that passes through the origin and intersect $\partial K$ in a point $a$. Then for any $\alpha < 1$ the set*

$$\alpha K + (1-\alpha)a = \{\alpha k + (1-\alpha)a \mid k \in K\}$$

*is a subset of $K$ and its boundary intersects the boundary of $K$ in the point $a$.*

**Proof.** Let $c \in \alpha K + (1-\alpha)a$. Then $c = \alpha b + (1-\alpha)a$ for some point $b \in K$. Since $a$ and $b$ are in $K$ and $K$ is convex we have $c \in K$. Therefore, $\alpha K + (1-\alpha)a \subseteq K$.

Since $a$ is a boundary point in $K$ we have $\alpha a + (1-\alpha)a = a$ is a boundary point in $\alpha K + (1-\alpha)a$.□

**Proof of Theorem 26**. Assume, without loss of generality, that the centroid of $K$ is the origin $\mathcal{O}$. Consider the convex set $\alpha K$. Since $\mathrm{Vol}(\alpha K) = \alpha^d \mathrm{Vol}(K)$ and $\alpha K \subset K$, the probability that a uniform random point in $K$ is also in $\alpha K$ is $\alpha^d$.

Next we will show that any halfspace containing a point in $\alpha K$ also contains at least $(1-\alpha)^d e^{-1}$ of the volume of $K$. This will complete the proof.

Let $a$ be any point in $\alpha K$ and $h$ be any hyperplane that passes through $a$. If $h$ passes through the origin then by Lemma 15 the result follows. So we may assume that the origin is not on $h$. The hyperplane $h$ defines two cuts (one on each side). Denote those cuts by $C_1 = K \cap [h \geq 0]$ and $C_2 = K \cap [h \leq 0]$. The line that passes through $\mathcal{O}$ and $a$ intersects the boundary of $K$ in two points $c_1 \in C_1$ and $c_2 \in C_2$. It therefore intersects $\alpha K$ in $\alpha c_1 \in C_1$ and $\alpha c_2 \in C_2$. Consider the halfspace $h_1$ that is parallel to $h$ and passes through $\alpha c_1$. This halfspace splits $C_1$ into two convex sets $C_{1,1} = C_1 \cap [h_1 \leq 0]$ and $C_{1,2} = C_1 \cap [h_1 \geq 0]$. The set $C_{1,1}$ is the points in $C_1$ that are between $h$ and $h_1$ and $C_{1,2} = C_1 \backslash C_{1,1}$ (see figure 5.1).

Consider the convex set $K' = (1-\alpha)K + \alpha c_1$. Since $\mathcal{O}$ is the centroid of $K$, we have $\alpha c_1$ is the centroid of $K'$. Since, by Lemma 28, $K' \subset K$ and since $h_1$ pass through the centroid of $K'$ the cut $K' \cap [h_1 \geq 0] \subseteq C_{1,2}$. Therefore, by Lemma 15 we have

$$\mathrm{Vol}(C_1) \geq \mathrm{Vol}(C_{1,2}) \geq \mathrm{Vol}(K' \cap [h_1 \geq 0]) \geq e^{-1}\mathrm{Vol}(K') = e^{-1}(1-\alpha)^d \mathrm{Vol}(K).$$

This completes the proof of the first part of the Theorem.

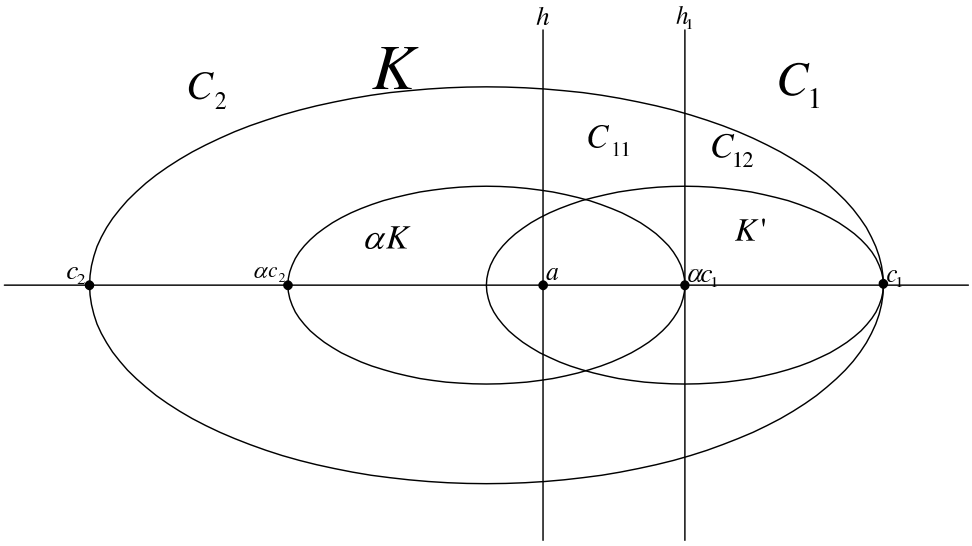

Figure 5.1: Theorem 26 proof

Now we choose $\alpha = 1/2$. With probability at least $2^{-d}$ the minimal volume of a cut is at least $2^{-d}e^{-1}\text{Vol}(K)$. Therefore the expectation is at least $2^{-d}2^{-d}e^{-1}\text{Vol}(K) = \text{Vol}(K)/(e4^d)$. □

Now applying (4.7) on the above Theorem we get Theorem 22.

### 5.2.3 Uniform Random Points in a Ball

In this section we prove the lower bound in Theorem 23

The following Lemma studies the distance of a uniform random point from the origin

**Lemma 29.** *The probability that the distance of a uniform random point in a ball of radius $R$ from the origin is less than $\alpha R$ is $\alpha^d$. In particular, for any $m > 1$, with probability at least $1 - e^{-d/m}$ a uniform random point in a ball of radius $R$ is of distance at least $R - \frac{R}{m}$ from the origin.*

**Proof.** The probability that a uniform random point is of distance at most $\alpha R$ from the origin is the probability that a uniform random point is in a ball of radius $\alpha R$. Therefore, it is equal to $V_d(\alpha R)/V_d(R) = \alpha^d$. When $\alpha = 1 - 1/m$ we have $\alpha^d \leq e^{-d/m}$.□

**Lemma 30.** *The probability that a uniform random point $x = (x_1, \ldots, x_n)$ in a ball of radius $R$ satisfies $|x_1| \geq \eta R$ is at most*

$$2\sqrt{d}e^{-\frac{d-1}{2}\eta^2}.$$

**Proof.** The probability that $|x_1| \geq \eta R$ is equal to two times the volume of the slice of the ball with $x_1 \geq \eta R$ divided by the volume of the ball. Therefore,

$$\Pr[|x_1| > \eta R] \quad = \quad \frac{2\int_{\eta R}^{R} V_{d-1}(\sqrt{R^2 - x^2})dx}{V_d(R)}$$

Substitute $x = \frac{x}{R}, dx = Rdx$ then we get

$$\begin{aligned} \frac{2\int_{\eta R}^{R} V_{d-1}(\sqrt{R^2 - x^2})dx}{V_d(R)} &= 2\int_{\eta}^{1} \frac{d}{d-1}\frac{S_{d-1}(1)}{S_d(1)}(1-x^2)^{\frac{d-1}{2}}dx \\ &\leq 4\frac{S_{d-1}(1)}{S_d(1)}(1-\eta^2)^{\frac{d-1}{2}} \leq 4\frac{S_{d-1}(1)}{S_d(1)}e^{-\frac{d-1}{2}\eta^2}. \end{aligned}$$

Now by (4.6) and [W] we have

$$4\frac{S_{d-1}(1)}{S_d(1)} = \frac{4}{\pi^{1/2}}\frac{\Gamma\left(\frac{d}{2}\right)}{\Gamma\left(\frac{d-1}{2}\right)} \leq \frac{4}{\pi^{1/2}}\sqrt{\frac{d}{2}} \leq 2\sqrt{d}.\square$$

The following Lemma gives the proof for Theorem 27

**Lemma 31.** *For a ball of volume $V$ the expectation of the minimal volume of a cut by a halfspace that passes through a uniform random point is at most*

$$2\alpha_0^d V$$

*where $\alpha_0 < 1$ is the solution of the equation*

$$\alpha_0^d = \sqrt{d}e^{-\frac{d-1}{2}\alpha_0^2}$$

*and is asymptotically ($d \to \infty$) equal to the solution of $\alpha_0 = e^{-\alpha_0^2/2}$ which is approximately equal to $\alpha_0 = 0.753089\cdots$.*

**Proof**. Let $\alpha < 1$. A uniform random point in a ball of radius $R$ can either be inside a ball $B_{\alpha R}$ of radius $\alpha R$ (with the same center) or outside. Each cut in a point inside $B_{\alpha R}$ is bounded by $V$ and the probability that a uniform random point is in $B_{\alpha R}$ is $\alpha^d$. Therefore, the points in $B_{\alpha R}$ contributes at most $V\alpha^d$ to the expectation.

The probability that a uniform random point is in $\overline{B_{\alpha R}}$ is bounded by 1 and, as in the proof of Lemma 30, the minimal volume of any cut by a halfspace that passes through a point in $\overline{B_{\alpha R}}$ is at most

$$\int_{\alpha R}^{R} V_{d-1}(\sqrt{R^2 - x^2})dx \leq \sqrt{d}e^{-(d-1)\alpha^2/2}V. \tag{5.1}$$

Therefore, the points in $\overline{B_{\alpha R}}$ contribute at most $\sqrt{d}e^{-(d-1)\alpha^2/2}V$ to the expectation. Hence, the expectation is at most

$$(\alpha^d + \sqrt{d}e^{-(d-1)\alpha^2/2})V.$$

Now, if $\alpha = \alpha_0$ where $\alpha_0^d = \sqrt{d}e^{-(d-1)\alpha_0^2/2}$, then the expectation is at most $2\alpha_0^d V$.□

### 5.2.4 The Lower Bound

In this section we prove Theorem 23.

We show that: there is some strategy of the teacher for choosing the target function and the counterexamples such that the learner, with probability at least 1/2, asks at least

$$\Omega(\beta^d + d^2 \log n)$$

equivalence queries, for $\beta = 1.192293$.

We will first show the lower bound $\beta^d$ and then $d^2 \log n$.

**The first lower bound**: The teacher will choose the target function to be $f_{\hat{w}} = f_{\hat{w},0}$ where $\hat{w} \in \mathbb{Z}^d$ is in a ball $B_R$ of radius

$$R = \frac{(n-1)^d d(d+1)^{(d+1)/2}}{2^d}.$$

Let $V$ be the volume of this ball.

The strategy of the equivalence oracle (the teacher) for choosing the counterexamples is: For each hypothesis $f_{w_0}$ (a point $w_0 \in B_R$ in the dual space) choose the counterexample

(a Halfspace in the dual space) that its corresponding Hyperplane is perpendicular to the line that passes through the origin $\mathcal{O}$ and $w_0$. This defines a cut with minimal volume.

We now run the algorithm with this strategy $r = \beta^d$ times. Let $C_i$ be the $i$th cut and $V_i$ its volume. Notice that the learner at each stage $k \leq r$ chooses a uniform random hypothesis $f_{w_k}$ in the dual space consistent with the counterexamples. This is equivalent to choosing a uniform random point in the domain $B_R \backslash (\cup_{i<k} C_i)$. This defines a probability space of the points $w_1, w_2, \ldots, w_r$. We can define an equivalent probability space as follows: At stage $k$ run the following

**Choose($k$)**
(1) $\ell \leftarrow 1$.
(2) Choose a uniform random point $w_k^{(\ell)} \in B_R$.
(3) If $w_k^{(\ell)} \in \cup_{i<k} C_i$ then $\ell \leftarrow \ell + 1$ and goto step (2),
(4) else $w_k = w_k^{(\ell)}$; output $w_k$.

We now define the following events:

1. $\mathcal{A}_k$ is the event that in stage $k$, $w_k^{(1)}$ in step (2) of **Choose($k$)** is not in $\cup_{i \leq k} C_i$, that is, $w_k = w_k^{(1)}$.

2. $\mathcal{B}_k$ is the event that $|w_k^{(1)}| > \alpha R$ for $\alpha = 0.83872$.

3. $\mathcal{C}$ is the event $\bigwedge_{i \leq r} (\mathcal{A}_i \wedge \mathcal{B}_i)$

We first show

**Claim 32.** *For large enough constant $d$ we have*

$$\Pr[\mathcal{C}] \geq \frac{1}{2}.$$

**Proof.** By Lemma 29 and (5.1) we have

$$\begin{aligned} \Pr\left[\mathcal{A}_k \wedge \mathcal{B}_k \middle| \bigwedge_{i<k} (\mathcal{A}_i \wedge \mathcal{B}_i)\right] &\geq \frac{V(B_R) - V(B_{\alpha R}) - V\left(\bigcup_{i<k} C_i\right)}{V(B_R)} \\ &\geq 1 - \alpha^d - (k-1)\sqrt{d} e^{-(d-1)\alpha^2/2}. \end{aligned}$$

Therefore, we have

$$\begin{aligned}
\Pr[\mathcal{C}] &\geq \prod_{k=1}^{r}\left(1-\alpha^d-(k-1)\sqrt{d}e^{-(d-1)\alpha^2/2}\right)\\
&\geq \left(1-\alpha^d-r\sqrt{d}e^{-(d-1)\alpha^2/2}\right)^r\\
&\geq 1-r\alpha^d-r^2\sqrt{d}e^{-(d-1)\alpha^2/2}\\
&\geq 1-\left((\alpha\beta)^d+\sqrt{d}e^{\alpha^2/2}\left(e^{-\alpha^2/2}\beta^2\right)^d\right)
\end{aligned}$$

Now since $\alpha\beta<1$ and $e^{-\alpha^2/2}\beta^2<1$, $\alpha$ and $\beta$ were chosen to maximize the value of $\beta$ but still satisfying the previous inequalities, for large enough constant $d$ we have $\Pr[\mathcal{C}]\geq 1/2$.□

Now, if event $\mathcal{C}$ happens then after $\beta^d$ equivalence queries, since each counterexample defines a cut of volume at most $\sqrt{d}e^{-(d-1)\alpha^2/2}V$, the remaining convex set $W_{\beta^d}$ has volume at least

$$V-\beta^d\sqrt{d}e^{-(d-1)\alpha^2/2}V = V-1.42\sqrt{d}(0.78)^dV.$$

That is, after $\beta^d$ equivalence queries, with probability at least $1/2$, only an exponentially small fraction of the volume are eliminated. This proves the first lower bound.

**The second lower bound**: For the second lower bound we use the same strategy of the teacher as in the first lower bound. Consider the uniform random variable $R_i$, the distance of the $i$-th uniform random point $w_i$ from the origin $\mathcal{O}$. Then $R_0=R$. Consider a random variable $X_i$ of the following game: $X_0=R$. At stage $i$ choose a uniform random point in a ball of radius $X_i$ and let $X_{i+1}$ be the distance of this point from the origin. Obviously,

$$E[R_i]\geq E[X_i] \text{ and } \Pr[R_i\geq 1]\geq \Pr[X_i\geq 1].$$

We now prove the following

**Lemma 33.** *With probability at least* $1/2$ *we have*

$$X_{(d-1)\ln(R/2)}\geq 1.$$

This proves that

$$\Pr[R_{(d-1)\ln(R/2)}\geq 1]\geq \frac{1}{2}$$

and therefore the expected number of steps in the algorithm is at least $t=(1/2)(d-1)\ln(R/2)=O(d^2\log n)$.

**Proof.** Consider the random variable $Z_{i+1}=X_i/X_{i+1}$ and

$$Z=\prod_{i=1}^{t}Z_i=\frac{X_0}{X_t}=\frac{R}{X_t}.$$

By Lemma 29 we have

$$\begin{aligned} E[Z_{i+1}] &= E\left[\frac{X_i}{X_{i+1}}\right] = E\left[E\left[\frac{X_i}{X_{i+1}}\middle| X_i\right]\right] \\ &= E\left[X_i E\left[\frac{1}{X_{i+1}}\middle| X_i\right]\right] \\ &= E\left[X_i \int_{\alpha=0}^{1}\left(\frac{1}{\alpha X_i}\right)\, d\alpha^d\right] \\ &= \int_{\alpha=0}^{1}\left(d\cdot\alpha^{d-2}\right)\, d\alpha \\ &= \left(1+\frac{1}{d-1}\right). \end{aligned}$$

Since $Z_i$ are independent random variables we have

$$E\left[\frac{R}{X_t}\right] = E[Z] = E\left[\prod_{i=1}^{t} Z_i\right] = \prod_{i=1}^{t} E[Z_i] = \left(1+\frac{1}{d-1}\right)^t.$$

Now by Markov Inequality we have

$$\Pr\left[X_t \le \frac{R}{2\left(1+\frac{1}{d-1}\right)^t}\right] = \Pr\left[\frac{R}{X_t} \ge 2\left(1+\frac{1}{d-1}\right)^t\right] = \Pr[Z \ge 2E[Z]] \le \frac{1}{2},$$

and for $t = (d-1)\ln(R/2)$ we have

$$\Pr[X_t > 1] \ge \frac{1}{2}.\square$$

## 5.3 Learning with many Random Consistent Hypothesis

In this section we show that any strategy that uses $o(\sqrt{d})$ calls to the RCH-oracle in each trial and builds any hypothesis that corresponds to the points in the convex closure of the Halfspaces received by the oracle to generate a hypothesis for the equivalence query does not run in polynomial time. This shows that any learning algorithm for Halfspaces using the RCH-oracle has to use the RCH-oracle at least $d^{2.5}\log n$ times.

We first prove the following corollaries

**Corollary 34.** *The probability that two independent uniform random points $x$ and $y$ in a ball of radius $R$ satisfies $|x^t y| \ge \eta R^2$ is at most*

$$2\sqrt{d}e^{-\frac{d-1}{2}\eta^2}.$$

**Proof.** Notice that if $y$ is uniform random point in $B_R$ then for any orthogonal matrix $U$ (independent of $y$) we have $Uy$ is uniform random. For any $x$ there is an orthogonal matrix $U$ such that $Ux = (||x||, 0, 0, \ldots, 0)$. Let $y' = Uy$. Then by Lemma 30

$$\begin{aligned}\Pr[|x^t y| \geq \eta R^2] &= \Pr[|(Ux)^t (Uy)| \geq \eta R^2] \\ &= \Pr[\|x\| \cdot |y'_1| \geq \eta R^2] \\ &\leq \Pr[|y'_1| \geq \eta R] \leq 2\sqrt{d} e^{-\frac{d-1}{2}\eta^2}. \square\end{aligned}$$

Now we prove

**Lemma 35.** *For a set $S$ of $k$ independent uniform random points in a ball of radius $R$ with probability at least*

$$1 - \binom{k}{2} \sqrt{d} e^{-\frac{(d-1)}{128k^2}}$$

*we have:*

$$\|Cl(S)\| \stackrel{\Delta}{=} \min_{x \in Cl(S)} \|x\| \geq \frac{R}{\sqrt{2k}}$$

**Proof.** Let $S = \{x^{(1)}, \ldots, x^{(k)}\}$. We have

$$\begin{aligned}\|Cl(S)\|^2 &= \min_{\lambda} \|\lambda_1 x^{(1)} + \cdots + \lambda_k x^{(k)}\|^2 \\ &= \min_{\lambda} \left( \sum_i \lambda_i^2 \|x^{(i)}\|^2 + 2 \sum_{i<j} \lambda_i \lambda_j x^{(i)t} x^{(j)} \right) \\ &\geq \min_{\lambda} \left( \sum_i \lambda_i^2 \|x^{(i)}\|^2 \right) + 2 \min_{\lambda} \left( \sum_{i<j} \lambda_i \lambda_j x^{(i)t} x^{(j)} \right) \\ &\geq \left( \min_i \|x^{(i)}\|^2 \right) \left( \min_{\lambda} \sum_i \lambda_i^2 \right) - 2 \left( \max_{\lambda} \sum_{i<j} \lambda_i \lambda_j \right) \left( \max_{i,j} |x^{(i)t} x^{(j)}| \right) \\ &\geq \frac{\min_i \|x^{(i)}\|^2}{k} - 2 \left( \max_{\lambda} \left( \sum_i \lambda_i \right)^2 \right) \left( \max_{i,j} |x^{(i)t} x^{(j)}| \right) \\ &= \frac{\min_i \|x^{(i)}\|^2}{k} - 2 \max_{i,j} |x^{(i)t} x^{(j)}|\end{aligned}$$

Here the minimum is over all $\sum_i \lambda_i = 1$ and $\lambda_i > 0$ for all $i$. By Lemma 29, with probability at least $1 - ke^{-d/8}$, $\min_i \|x^{(i)}\| > 7R/8$. By Corollary 34, with probability at least

$$1 - \binom{k}{2} \sqrt{d} e^{-\frac{(d-1)}{128k^2}}$$

we have $\max_{i,j} |x^{(i)t}x^{(j)}| \leq R^2/(8k)$ for every $i < j$. Therefore, with probability at least

$$1 - 2\binom{k}{2}\sqrt{d}e^{-\frac{(d-1)}{128k^2}},$$

$$\begin{aligned} \|Cl(S)\|^2 &\geq \frac{\min_i \|x^{(i)}\|^2}{k} - 2\max_{i,j} |x^{(i)t}x^{(j)}| \\ &\geq \frac{49R^2}{64k} - \frac{R^2}{8k} \geq \frac{R^2}{2k} \end{aligned}$$

This implies the result.□

Notice that when $k = o(d^{1/2})$ then with exponentially close to 1 probability the closest point in $Cl(S)$ from the origin is at distance at least $R/d^{1/4}$. By (5.1), for such point, the volume of the minimal cut is $e^{-O(\sqrt{d})}$ which is exponentially small. Now using exactly the same approach as in the previous section the result follows.

# Chapter 6

# Open Problems

We use a new technique and achieve a learning algorithm for halfspaces that uses on average

$$(1+c)\cdot d^2\left(\log n+\frac{\log d}{2}\right)$$

equivalence queries for any constant $c>0$ using $O(d\log d)$ calls to the RCH-oracle.

In [MT94] Maass and Turan show a lower bound of

$$\binom{d}{2}\log n\leq\frac{1}{2}d^2\log n.$$

on the number of equivalence queries needed to learn $\mathrm{HS}_n^d$ with any learning algorithm that has unlimited computational power that can ask equivalence query with any hypothesis. It is now an open problem to

1. Close the gap between this lower bound and the new upper bound.
2. Get rid of the term $(d^2\log d)/2$ in the upper bound.
3. Show that $\mathrm{RCH}_C$ can be simulated in polynomial time. This will give a polynomial time learning algorithm for $\mathrm{HS}_n^d$ with $O(d\log n)$ equivalence queries.

Another interesting question is whether parallel algorithms can speed up learning Halfspaces. From [B97], if there is a parallel algorithm with $e$ processors that asks $t$ parallel equivalence queries then there is a sequential algorithm that asks $t\log e$ equivalence queries. Now since $t\log e>(1/2)d^2\log n$ and $e=poly(d,\log n)$, any parallel algorithm will ask at least

$$\Omega\left(\frac{d^2\log n}{\log d+\log\log n}\right)$$

parallel equivalent queries. Is there such algorithm?

It is also interesting to study learning Halfspaces in other models. See for example [BJT02, BG02].

# Appendix A

In this chapter we prove lemma 4.
The lemma states that

$$n^{d(d+1)} > |\mathrm{HS}_n^d| > n^{d(d-1)/2}.$$

**Proof.** Let $C(d)$ be the number of zero halfspaces $f_w = [w^T x \geq 0]$ over $[n]^d$. Then

$$C(1) = 2.$$

We will prove

$$C(d) \geq (n^{d-1} + 1)C(d-1)$$

Let $f_w \in \mathrm{HS}_n^d$. We will count the number of ways that the weights $w = (w_1, \ldots w_d)$ can be chosen to give different functions $f_w$.
Partition the domain $[n]^d$ into two sets:

$$X_0^d = \{(x_1, \ldots, x_{d-1}, 0) \mid x_i \in [n] \; for \; 1 \leq i < d\}$$

and

$$X_0^d = \{(x_1, \ldots, x_{d-1}, x_d) \mid x_i \in [n] \; for \; 1 \leq i \leq d, \; x_d \neq 0\}$$

Since the number of zero halfspaces $f_w$ over the domain $X_0^d$ is equal to the number of zero halfspaces in $HS_n^{d-1}$, the weights $w_1, \ldots, w_{d-1}$ can be chosen in $C(d-1)$ different ways, each of which makes $f_w$ have a different output for some $x \in X_0^d$. This choice of the first $d-1$ weights fixes the relative order of $w^T x$ for all $X_0^d$, regardless of the choice of $w_d$. Then $w_d$ can be chosen to make $w^T x < 0$ for the first $i$ of the $x \in X_0^d$ in this order, for $0 \leq i \leq n^{d-1}$.
Therefore $C(d) \geq (n^{d-1} + 1)C(d-1)$ as claimed.
Therefore, by induction on $d$,

$$C(d) \geq \prod_{i=0}^{d-1}(n^i + 1) > n^{d(d-1)/2}.$$

For the other direction, observe that by the argument used in proof of lemma 13, each

halfspace corresponds to at least one point at the intersection of $d+1$ hyperplanes in the dual domain. There are $n^d$ such hyperplanes, and hence at most

$$\binom{n^d}{d+1} \leq n^{d(d+1)}.$$

possible points of intersection. Therefore, there are less than $n^{d(d+1)}$ halfspaces over $[n]^d$. □

# Bibliography

[A88] D. Angluin. Queries and concept learning. *Machine Learning*, 2, pp. 319-342, 1987.

[B97] N. H. Bshouty. Exact learning of formulas in parallel. *Machine Learning*, 26, pp. 25-41, 1997.

[BBK97] S. Ben-David, N. H. Bshouty, E. Kushilevitz. A Composition Theorem for Learning Algorithms with Applications to Geometric Concept Classes. STOC 97, pp. 324-333, 1997.

[BC+96] N. H. Bshouty, R. Cleve, R. Gavald, S. Kannan, C. Tamon. Oracles and Queries That Are Sufficient for Exact Learning. *Journal of Computer and System Sciences*, 52(3): pp. 421-433, 1996.

[BEHW89] A. Blumer, A. Ehrenfeucht, D. Haussler, M. K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. J. ACM 36(4): 929-965 (1989)

[BG02] N. H. Bshouty and D. Gavinsky. PAC=PAExact and other equivalent models. FOCS 02. pp. 167-176, 2002.

[BJT02] N. H. Bshouty, J. Jackson and C. Tamon, Exploring learnability between exact and PAC. COLT 02, pp. 244-254, 2002.

[BV02] D. Bertsimas, S. Vempala. Solving convex programs by random walks. STOC 02: pp. 109-115, 2002.

[DV04] J. Dunagan and S. Vempala. A simple polynomial-time rescaling algorithm for solving linear programs. STOC 04, pp. 315-320, 2004.

[G60] B. Grunbaum. Partitions of mass-distributions and convex bodies by hyperplanes. *Pacific J. Math*, 10, pp. 1257-1261, 1960.

[H94] J. Hastad. On the Size of Weights for Threshold Gates. *SIAM Journal on Discrete Mathematics*, (7) 3, pp. 484-492, 1994.

[HW87] D. Haussler and E. Welzl. Epsilon-nets and simplex range queries. *Discrete Comput. Geom.*, 2: pp. 127-151, 1987.

[L88] N. Littlestone. Learning when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2, pp. 285-318, 1988.

[L98] L. Lovász. Hit-and-run mixes fast. *Mathematical Programming*, 86, pp. 443-461, 1998.

[M94] W. Maass. Perspectives of current research about the complexity of learning on neural nets. In V. P. Roychowdhury, K. Y. Siu, and A. Orlitsky, editors, Theoretical Advances in Neural Computation and Learning, pp. 295-336. Kluwer Academic Publishers (Boston), 1994.

[MP43] W. S. McCulloch and W. Pitts. A logical calculus of ideas immanent in nervous activity. Bulletin of mathematical biophysics, 5:115-133, 1943.

[MT94] W. Maass and G. Turan. How fast can a threshold gate learn. In S. J. Hanson, G. A. Drastal, and R. L. Rivest, editors, Computational Learning Theory and Natural Learning System: Constraints and Prospects, MIT Press (Cambridge), pp. 381-414, 1994.

[P94] I. Parberry. Circuit complexity and neural networks. The MIT press (1994).

[R62] F. Rosenblatt, Principles of neurodynamics: Perceptrons and the theory of brain mechanisms, Spartan Books, New York, 1962.

[S02] R. Servedio. Perceptron, Winnow, and PAC Learning. *SIAM Journal on Computing*, 31(5), pp. 1358-1369, 2002.

[V96] P. M. Vaidya. new algorithm for minimizing convex functions over convex sets. *Mathematical Programming*, pp. 291-341, 1996.

[V84] L. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11), pp. 1134-1142, 1984.

[VC71] V. N. Vapnik, A. Y. Chervonenkis, On the uniform convergence of relative frequencies of events to their probabilities. *theory of Probability and its Applications*, 16(2), pp. 264-280, 1971.

[W] E. W. Weisstein. Gamma Function. From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/GammaFunction.html.

[WD81] R. S. Wenocur, R. M. Dudley. Some special Vapnik-Chervonenkis classes, *Discrete Math.*, 33, pp. 313-318, 1981.