Learning Halfspaces;
Literature Review and Some Recent Results

Daniel Khashabi

May 11, 2016

Abstract

This write up addresses the problem of learning half-spaces, a fundamental problem in machine learning. There is a vast literature around this problem, on approximation issues and ardness analysis. I go over the related literature and explain all the necessary background, hopefully, with an easy-to-understand language. Later I go over some important results from the past as well as details of a recent result.

1 Introduction

Half-spaces simply defined as linear weighted sum of some variables:

$$\text{sign}\left(w_1 x_1 + \ldots + w_n x_n - \theta\right)$$

Half-spaces are the cornerstone of the statistical machine learning; vast space of machine learning models are extensions of half-spaces. For example Naive Bayes, Logistic Regression and Conditional Random Fields[1] (normalized exponentiated linear separators), Neural Networks (stack of linear/nonlinear separators), Support Vector Machines[2] (separators with preference for a big margin). Even Boolean functions can implicitly be simulated with linear functions: AND of variables ($\theta = n - \frac{1}{2}$), Or of variables ($\theta = \frac{1}{2}$), and majority-vote ($\theta = \frac{n}{2}$).

2 Preliminaries

We give a brief description of the important terminology, parameters and scenarios. For more more exact definitions we refer the reader to the important surveys in the literature [3, 4].

Basic notation. Define the input space to be $X = \mathbb{R}^d$ and the output space to be $Y = \{\pm 1\}$. Suppose we are given m training instances $S = \{\boldsymbol{x}_i, y_i\}_{i=1}^m$, are drawn from an unknown distribution $\mathcal{D}$, with marginal distributions $\mathcal{D}_X$ and $\mathcal{D}_Y$. Define the hypothesis class $\mathcal{H}$ to be the space of all functions $h_{\boldsymbol{w}}$ that we can use to approximate our problem. Define the expected error (or risk) to be $\text{Err}_{\mathcal{D}}(h) \triangleq \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[h_{\boldsymbol{w}}(\boldsymbol{x}) \neq y\right] = \text{P}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left(h_{\boldsymbol{w}}(\boldsymbol{x}) \neq y\right)$. Define the best hypothesis to be $h^* \triangleq \arg\min_{h\in\mathcal{H}} \text{Err}_{\mathcal{D}}(h)$ and its error $\text{OPT} = \text{Err}_{\mathcal{D}}(h^*)$. The empirical error (risk) is defined according to the training data $\widehat{\text{Err}}_{\mathcal{D}}(h) \triangleq \frac{1}{|S|}\sum_{(\boldsymbol{x},y)\sim S} h_{\boldsymbol{w}}(\boldsymbol{x}) \neq y$. In reality we choose the best hypothesis via training data S and the empirical risk: $\hat{h} \triangleq \arg\min_{h\in\mathcal{H}} \widehat{\text{Err}}_{\mathcal{D}}(h)$. Thus is commonly referred to as *empirical risk minimization* (ERM).

Learning half-spaces. If the hypothesis class $\mathcal{H}$ consists of half-spaces and each member of this space $h_{\boldsymbol{w}} \in \mathcal{H}$ is defined as $h_{\boldsymbol{w}}(\boldsymbol{x}) = \text{sign}\left(\langle \boldsymbol{x}, \boldsymbol{w} \rangle\right)$.

Realizable scenario. When output label does not have any noise and OPT is zero (i.e. there exists a plane which perfectly separates the data in the hypothesis space $\mathcal{H}$). The opposite of the realizable scenario is the *agnostic* scenario for which no assumptions are made. In the literature realizable and agnostic case are also referred to as *noise-free* and *noisy* scenarios, respectively as well.

Proper vs improper learning. If an algorithm is strictly required to output an output from the set $\mathcal{H}$ the algorithm is called *proper learning*. In some scenarios $\mathcal{H}$ is subset of another set $\mathcal{H}'$. If the algorithm is allowed to output hypothesis outside $\mathcal{H}$ inside $\mathcal{H}' \backslash \mathcal{H}$, as long as it satisfies some certain guarantees (possibly in expectation). This is often referred to as *representation-independent learning* or sometimes *improper learning*[*]. In the context of half-spaces, an improper algorithm might sometimes output a classifier that is not a half-space classifier.

The learning problem. Find h such that $\text{Err}_{\mathcal{D}}(h) \leq \text{Err}_{\mathcal{D}}(h^*) + \epsilon$, for some $\epsilon \in [0, 1]$. ϵ is commonly called the *excess* error. This is sometimes referred to as the *exact* learning since the algorithms gets arbitrarily close to the exact objective (in additive sense).

The approximate learning problem. Find h such that $\text{Err}_{\mathcal{D}}(h) \leq \mu \text{Err}_{\mathcal{D}}(h^*) + \epsilon$, for some $\epsilon \in [0, 1]$ and $\mu > 1$. μ is commonly called the *approximation* ratio, which is the relaxed version version of the exact learning (i.e. $\mu = 1$)[†].

For many of the algorithms we introduce here, their performance measures will be a function of the approximation factor μ, and naturally as we decrease μ close to one, the running time should get worse. Hence we one can think of these results as interpolation between approximate learning and exact learning (when setting μ to one).

Another way of converting an approximate problem is to set $\mu = 1 + \epsilon'$:

$$\mu \leq \mu \text{OPT} + \epsilon_{old} = (1 + \epsilon')\text{OPT} + \epsilon = \text{OPT} + \epsilon_{new}, \quad \text{where } \epsilon_{new} = (1 + \text{OPT})\epsilon$$

Hence ϵ can be set small enough so that ϵ_{new} is close to what use desires.

Efficient algorithm/learner. An algorithm is efficient if it runs in poly $\left(\frac{1}{\epsilon}, \log \frac{1}{\delta}, d\right)$, where ϵ is the excess error, d is the input dimension and the guarantee holds with probability at least $1 - \delta$.

Bayes optimal classifier. If the goal is to have minimum misclassification probability, the optimal classifier given the distribution from which the data is generated is given by:

$$h_{\boldsymbol{w}}(\boldsymbol{x}) = \begin{cases} 1 & \eta(\boldsymbol{x}) \geq \frac{1}{2} \\ 0 & o.w., \end{cases}$$

where $\eta(\boldsymbol{x}) \triangleq \mathbb{E}\left[Y = 1 | X = \boldsymbol{x}\right]$.

[*]Although there is nothing "improper" about it!

[†]Statistical estimation problems are mostly evaluated based on additive errors while combinatorial problems approximation problems are evaluated based on ratio of objectives. The one we study here has both factors, but we use the *approximation* term when referring to the multiplicative approximation, following the computer science community.

Random classification noise. Define *noise level* to be $s(\boldsymbol{x}) = \min\left(\mathrm{P}\left(Y = 1|X = \boldsymbol{x}\right), 1 - \mathrm{P}\left(Y = 1|X = \boldsymbol{x}\right)\right)$ which can be simplified to $s(\boldsymbol{x}) = \frac{1-\eta(\boldsymbol{x})}{2}$. Intuitively the smaller the noise level across all the instance is, it is easier to do classification. One way looking at this is that, each label of a noise-free sample is flipped independently with some fixed probability. In this model the noise is independent of the actual example points, which are generated according to a probability distribution. When $s(\boldsymbol{x}) = 0$ almost surely in the deterministic case. Clearly when $s > 0.5$ there is no hope of learning; hence it is almost assume that $s < 0.5$.

In the exact setting ($\mu = 1$) we say an algorithm is learnable in presence of random classification noise, if for any $\epsilon, \delta > 0$ and s it produces an output a classifier with error ϵ with probability at least $1 - \delta$, in time poly $\left(\frac{1}{\epsilon}, \log\frac{1}{\delta}, d, \frac{1}{1-2s}\right)$. Note that in this criterion, as $s \to 0.5$, $\frac{1}{1-2s}$ gets bigger.

This model first introduced by Angluin and Laird[5], and it is sometimes referred to as *benign (nonadversarial)* noise.

Massart noise. Massart noise with parameter β with parameter $\beta > 0$ is a condition that for all $\boldsymbol{x}$:

$$|\mathrm{P}\left(Y = 1|X = \boldsymbol{x}\right) - \mathrm{P}\left(Y = 0|X = \boldsymbol{x}\right)| \geq \beta$$

Equivalently, given a noise-free dataset, an adversary constructs the noisy data by flipping the label with probability at most $\frac{1-\beta}{2}$.

Malicious noise. Introduced by Valiant [6] as an extension of his basic PAC framework. In this model each training instance, with some certain probability, is replaced with an adversarially chosen one (sometimes also referred to as "distribution free" [7]).

3 Known results

In what follows we discuss the existing results for learning half-spaces under certain assumptions. We start off by results of exact learning (i.e. $\mu = 1$) and which will be followed by results of approximate learning (i.e. $\mu > 1$).

Learning in realizable case is *usually* easy; learning half-spaces in realizable case is always easy. In the absence of (label/output) noise (when the data is realizable, i.e. $\mathrm{OPT} = 0$) half-spaces are efficiently learnable.

One simple way of showing it is via Linear Programming. We show that the problem of half-spaces in the realizable case can be modelled as an LP. Given the training data S, we are looking for a realizable answer we need a w such that $y_i \boldsymbol{w}.\boldsymbol{x}_i > 0, \forall i = 1, \ldots, m$. Define $\delta = \min_{i \in [m]} y_i \boldsymbol{w}.\boldsymbol{x}_i$ (because we are in the realizable scenario it exists). The objective can equivalently written as $y_i \boldsymbol{w}.\boldsymbol{x}_i \geq \delta, \forall i \in [m]$. To keep the notation cleaner we rewrite this equation as $y_i \frac{\boldsymbol{w}}{\delta}.\boldsymbol{x}_i \geq 1, \forall i \in [m]$. Note that for any $\boldsymbol{w}$, $\frac{\boldsymbol{w}}{\delta}$ is another solution in the solution space. Hence the objective can be just written as $y_i \boldsymbol{w}.\boldsymbol{x}_i \geq 1, \forall i \in [m]$. Given the definition of the LP the problem is easily solvable. Note that the LP has only inequality constraints (i.e. it is a feasibility LP).

There are other algorithms (with guarantees) for the realizable scenario; for example the Perceptron algorithm of Rosenblatt [8] is generated to find find a correct classifier after at most $(RB)^2$ updates/iterations, where $B = \min\left\{\|\boldsymbol{w}\| : \forall i \in [m], y_i \boldsymbol{w}.\boldsymbol{x}_i \geq 1\right\}$ and $R = \max_i \|\boldsymbol{x}_i\|$. Note how the final bound depends on the realizablity in definition of B.

Another popular algorithm is Winnow, which unlike perceptron uses multiplicative updates. Given that the hypothesis space consists of disjunctions (or conjunctions) and assuming that the target function can be expressed with r coefficients (queries), Winnow algorithms with at most $O\left(r \log n\right)$ mistakes finds the target function [9].

Learning in realizable scenario can sometimes be hard, even in the realizable case. We argued that for half-spaces learning in the realizable case is easy. For completeness we mention that learning some concept classes (beyond half-spaces), even in the realizable case can be hard. A famous example is 3-term DNFs: Unless (randomized poly) RP = NP, *proper* learning of 3-term DNF class is not efficiently PAC-learnable [10]. A common way of proving this is by reduction to the 3-coloring problem [11].

Another interesting point about 3-term DNFs is that, if we remove the *proper* learnabilty condition (i.e. we allow the learning algorithm to sometimes output from a super-set of 3-term DNFs), it is efficiently learnable. Specifically one can efficiently learn 3-term DNFs using 3-CNFs. This is a good example to be aware of, since we will see a similar scenario for learning half-spaces in the *agnostic* case.

Proper learning of half-spaces in agnostic scenario is hard. For many functions ERM in the agnostic case is **NP**-hard; i.e. unless P = NP, there is no polynomial time approximation scheme for finding a member in the class that approximately minimizes the empirical risk on a given training sample. Ben-David *et al.* [12] proves this result for class of monomials (disjunctions), axis-aligned hyper-rectangles, closed balls and monotone monomials.

When restricted to proper algorithms, learning half-spaces is equivalent to *minimizing disagreement* (aka co-agnostic learning) which is a well-studied problem and it's included in Karp's celebrated work [13] as an **NP**-hard problem.

Half-spaces are not efficiently properly agnostically PAC learnable. This has proved in different ways based on several commonly considered hard problems; Kalai *et al.* [14] based on hardness of learning parity functions, Feldman *et al.* [15] based on hardness of shortest vector problem, Daniely and Shalev-Shwartz [16] using hardness of refuting random k-SAT, Klivans and Kothari [17] via hardness of learning sparse parity functions (under uniform distribution), Daniely [18] using hardness of refuting random k-XOR. Feldman[15] has shown that for any constant $\epsilon > 0$ determining whether the best disjunction for a given $\epsilon > 0$ has error $\leq \epsilon$ or error $> \frac{1}{2} - \epsilon$ is **NP**-hard. Guruswami and Raghavendra [19] showed that for any $\epsilon \in (0, 1/2]$, it is **NP**-hard to find a halfspace with error bounded by $1/2 - \epsilon$.

Even improper learning of half-space in the agnostic scenario is *probably* hard. Klivans and Kothari[17] show that any algorithm for improperly agnostically learning half-spaces requires $n^{\Omega(\log(1/\epsilon))}$ time under the assumption that k-sparse parities under uniform distribution requires $n^{\Omega(k)}$ time.

Data-independent and bounded noise. Blum *et al.* [20] gave an efficient algorithm under independent label noise. Awasthi *et al.* [21] achieves similar result by finding efficient algorithm for data corrupted by bounded noise, also called Massart noise [22], as well as providing the lower-bounds under the same noise assumption.

realizable or **a**gnostic	**p**roper or **i**mproper	approximation factor	marginal distribution $\mathcal{D}_X$	margin(γ)?	extra assumptions	run time	label complexity	reference
a	p	$\mu = 1$	uniform	$\times$	-	-	poly $\left(d^{1/\epsilon^4}, \log\frac{1}{\delta}\right)$	Kalai *et al.*, 2008 [14]
a	p	$\mu = 1$	log-concave	$\times$	-	-	poly $\left(d^{f(\epsilon)}, \log\frac{1}{\delta}\right)$	Kalai *et al.*, 2008 [14]
a	p	$\mu = \sqrt{\log\frac{1}{\mathrm{OPT}}}$	uniform	$\times$	-	poly $\left(n, \frac{1}{\epsilon}, \log\frac{1}{\delta}\right)$	poly $\left(\frac{n^2}{\epsilon^2}\log\frac{n}{\delta}\right)$	Kalai *et al.*, 2008 [14]
a	i	$\mu = 1$	-	✓	L-Lipschitz constant	poly $\left(\exp\left(L\log\frac{L}{\epsilon}\right)\right)$	$\left(\frac{2L+3\sqrt{2\ln 8/\delta}}{\epsilon}\right)^2$	Shalev-Shwartz, 2011 *et al.* [28]
a	i	$\mu = \frac{\frac{1}{\gamma}}{\log\frac{1}{\gamma}}$	-	✓	-	-	-	Birnbaum and Shalev-Shwartz [25]
a	i	$\forall\mu > 1$	uniform	-	-	poly $\left(d^{\frac{\log^3\frac{1}{\mu-1}}{(\mu-1)^2}}, \frac{1}{\eta}\right)$	poly $\left(d^{\frac{\log^3\frac{1}{\mu-1}}{(\mu-1)^2}}, \log\frac{1}{\eta}\right)$	Daniely, 2015 [27]
a	p	$\mu = 1$	uniform over unit ball in $\mathbb{R}^d$	$\times$	Massart noise	$O\left(\log\frac{1}{\epsilon}\right)$	$O\left(d(d+\log\frac{k}{\delta})\right)$	Awasthi, 2015 *et al.* [21]
a	p	$\mu = 1$	-	$\times$	-	poly $\left(n^{(1/\epsilon^2)\log^2(1/\epsilon)}, d, \log\frac{1}{\delta}\right)$	-	Zhang, 2015 *et al.* [29]
a	p	$\mu = 1$	-	$\times$	-	$\begin{cases}\text{poly}\left(n, d^{(1/\epsilon^2)\log^2(1/\epsilon)}, \log\frac{1}{\delta}\right) & p=1 \\ \text{poly}\left(n, d, e^{(q/\epsilon^2)\log^2(1/\epsilon)}, \log\frac{1}{\delta}\right) & p>1\end{cases}$	-	Zhang, 2015 *et al.* [29]

Table 1: Summary of upper-bound results for learning half-spaces. The results sorted chronologically.

The fully (distribution-free) agnostic model. The first fully agnostic (distribution free learning) result is for Kearns and Li [23] has an approximation ratio of $\mu = O(d)$.

Half-spaces with margin. One popular scenario is when there is preferred margin between the half-spaces. The best approximation ratio for this setting is $O\left(\frac{1/\gamma}{\log\frac{1}{\gamma}}\right)$ [24, 25].

Approximate learning of half-spaces is efficiently learnable with distributional assupmtions. One can simplify the problem by making distributional assumption for the generative process of data. One common assumption is uniform assumption; under this assumption Kalai *et al.* [14] presented an algorithm with $\mu = O\left(\sqrt{\log\frac{1}{\mathrm{OPT}}}\right)$ approximation ratio. Later Awasthi *et al.* [26] improved the approximation ratio to $\mu = O(1)$ under the same assumption. Daniely [27] extends Awasthi *et al.* algorithm, by providing an efficient algorithm for any fixed μ assuming uniform distribution. The resulting algorithm is exponential the approximation parameter μ, while polynomial in other paramaters (hence the name "PTAS"[‡]) (Section 4).

4 PTAS for agnostically learning half-spaces under uniform dist. [27]

This is a recently result presented by Daniely [27]. The results is referred to as "PTAS" since the final bound has approximation ratio of μ, and the final algorithm has efficient runtime assuming that μ is a constant.

The solution is presented in algorithm 1; it contains different pieces of the previous results. Hence the analysis is relatively simple and it is done by combining results of previosly known results. The final algorithm is *improper* since the may not always output a valid half-space; instead it outputs a polynomial approximation of the sign function, around the decision margin.

A common technique among the previous works is approximating the sign function with a polynomial function (e.g. see [14, 30, 28, 25]). He makes this smart observation that such approximations work best around a strip close to the decision margin (close to the non-linearity of the sign function). Similar ideas are presented by Awasthi *et al.* [26] under the label of *Localization*. The main idea in *localization* [31, 26] is to focus on the solutions "close" to the current best solution in the hypothesis space. The algorithm of [26] iteratively minimize empirical hinge loss on a strip of width γ around the decision margin:$T_{d,\gamma} = \{\boldsymbol{x} | \, |\boldsymbol{x}, \boldsymbol{w}| \leq \gamma\}$.

Next we present the main claim of the paper, as well as the claims used in the analysis borrowed from previous works and sketch of the proof.

[‡]Polynomial-time approximation scheme

Algorithm 1: Improper PTAS learning of half-spaces.

```
Input: Samples from $\mathcal{D}$;
while not at end of this document do
    Find a $\boldsymbol{w}$ such that $\text{Err}_{\mathcal{D}}(h) \leq \alpha_0 s$ [26, Theorem 1.1] ;
    Define a $\gamma$-strip around the decision line: $T_{d,\gamma} = \{\boldsymbol{x} | \, |\boldsymbol{x}, \boldsymbol{w}| \leq \gamma\}$ ;
    Find a $d$-variate degree $r$ polynomial $p$ such that $\text{Err}_{\mathcal{D}|T}(p) \leq \text{Err}_{\mathcal{D}|T}(h^*) + \min_{p'} \|h^* - p'\| + \beta$,
      s.t. $h^*$ is the optimal half-space classifier with respect to $\mathcal{D}$ [14, Theorem 1]. ;
    With probability 0.5 return $h_{\boldsymbol{w}}$ and with probability 0.5 return
        $$h(\boldsymbol{x}) = \begin{cases} h_{\boldsymbol{w}}(\boldsymbol{x}) & |\boldsymbol{w}.\boldsymbol{x}| > \gamma \\ \text{sign}\,(p(\boldsymbol{x})) & |\boldsymbol{w}.\boldsymbol{x}| \leq \gamma \end{cases} \quad (1)$$
```

Theorem 4.1. *The solution presented in algorithm 1, for proper choices of its parameters, for every μ ($\mu > 0$) is efficient algorithm for agnostically learning of half-spaces under uniform distribution with an approximation ratio $1 + \mu$. Specifically the resulting solution tolerates noise rate of $(2 - \mu)s$, i.e. if $Err_{\mathcal{D}}(h_{\boldsymbol{w}^*}) \leq (2 - \mu)s$ then $Err_{\mathcal{D}}(h_{\boldsymbol{w}}) \leq s$, it runs in pseudo-polynomial time poly $\left(d^{\frac{\log^3 \frac{1}{\mu - 1}}{(\mu-1)^2}}, \frac{1}{\eta} \right)$ with pseudo-polynomial label complexity of poly $\left(d^{\frac{\log^3 \frac{1}{\mu - 1}}{(\mu-1)^2}}, \log\frac{1}{\eta} \right)$.*

The runtime and label complexity of the algorithm needs to be calculated by putting together runtime and label complexity of each sub-algorithm. The runtime and label complexity of the first step are poly $\left(d, \frac{1}{s}\right)$ and poly $\left(d, \log\frac{1}{s}\right)$ (see [26, Theorem 1.1]). Based on [14], one can get the calculate the total runtime and label complexity.

The following lemma is a simple useful fact used in many places.

Lemma 4.2.

$$P_{x \sim \mathcal{D}}\left(h_{\boldsymbol{w}_1}(x) \neq h_{\boldsymbol{w}_2}(x)\right) = \frac{\theta(\boldsymbol{w}_1, \boldsymbol{w}_2)}{\pi}$$

As the angle between the weight vectors grow, the upper-bound on their error probability also grows. This is intuitive $h_{\boldsymbol{w}}$ is a linear classifier and only dependents on the direction of $\boldsymbol{w}$.

Lemma 4.3 (Localization lemma)**.** *For any $r > 0$:*

$$P_{x \sim \mathcal{D}}\left(h_{\boldsymbol{w}}(\boldsymbol{x}) \neq h_{\boldsymbol{w}^*}(\boldsymbol{x})\right) \leq \frac{4\pi(\boldsymbol{w}, \boldsymbol{w}^*)}{\pi} \exp\left(-\frac{1}{8} r^2 d\right)$$

Lemma 4.2 is same as previous lemma when $r \to 0$. For bigger values of r, the further a point is from the decision boundary, the lower the error probability is. This lemma can be used to choose a strip-size which has certain error upper-bound. Intuitively this lemma says as the size of the strip grows, the error probability drops exponentially. The proof is relatively simple, based on geometric properties and using concentration inequalities.

For big enough degree, a polynomial can easily approximate a half-space decision.

Lemma 4.4 (Uniform half-spaces [30]). *$\forall \tau > 0$ there exists a polynomial with degree r in d-dimension such that $r = O\left(\frac{\log^2 \frac{1}{\tau}}{\tau^2}\right)$ and $\mathbb{E}_{x \sim \mathcal{D}_X}\left[\|h_{\boldsymbol{w}}(x) - p(\boldsymbol{x})\|\right] \leq \tau$.*

Theorem 4.5 (Kalai *et al.* [14]). *There is an algorithm with run-time poly $\left(d^r, \frac{1}{\epsilon}\right)$ such that for every distribution $\mathcal{D}$ and every h it returns a d-variate degree r polynomial p, with*

$$Err_{\mathcal{D}}(p) \leq Err_{\mathcal{D}}(h^*) + \min_{p'} \left\|h^* - p'\right\| + \beta$$

Next is the sketch of the proof for our Theorem 4.1.

Proof sketch. $\text{Err}_{\mathcal{D}}(h^*) \leq (1-\mu)\eta$ then the error of the returned classifier should be $\text{Err}_{\mathcal{D}}(h) \leq \eta$. Following the algorithm, the expected error is decomposed into two conditionals. Denote the outside $T_{d,\gamma}$ to be $T^c_{d,\gamma}$. Based on equation (1):

$$\text{Err}_{\mathcal{D}}(h) = \text{P}_{(x,y)\sim\mathcal{D}}\left(x \in T_{d,\gamma}\right) \text{Err}_{\mathcal{D}|T\times\{\pm 1\}}(p) + \text{P}_{(x,y)\sim\mathcal{D}}\left(x \notin T_{d,\gamma}\right) \text{Err}_{\mathcal{D}|T^c\times\{\pm 1\}}(h_{\boldsymbol{w}})$$

For the first term, using the bound given in Kalai *et al.*, we choose β and r such that:

$$\text{P}_{(x,y)\sim\mathcal{D}}\left(x \in T_{d,\gamma}\right) \text{Err}_{\mathcal{D}|T\times\{\pm 1\}}(p) \leq \text{P}_{(x,y)\sim\mathcal{D}}\left(x \in T_{d,\gamma}\right) \text{Err}_{\mathcal{D}|T\times\{\pm 1\}}(h_{\boldsymbol{w}^*}) + (\mu-1)s/2$$

For the second term we can set use Lemma 4.3 and set the size of the margin so that $\text{P}_{(x,y)\sim\mathcal{D}}\left(x \notin T_{d,\gamma}\right) \leq (\mu-1)s/2$, and hence:

$$\text{P}_{(x,y)\sim\mathcal{D}}\left(x \notin T_{d,\gamma}\right) \text{Err}_{\mathcal{D}|T^c\times\{\pm 1\}}(h_{\boldsymbol{w}}) \leq \text{P}_{(x,y)\sim\mathcal{D}}\left(x \notin T_{d,\gamma}\right) \text{Err}_{\mathcal{D}|T^c\times\{\pm 1\}}(h_{\boldsymbol{w}^*}) + (\mu-1)s/2$$

Putting these together we would get

$$\text{Err}_{\mathcal{D}}(h) \leq \text{Err}_{\mathcal{D}}(h_{\boldsymbol{w}^*}) + (\mu-1)s = (2-\mu)\eta + (\mu-1)s = s$$

■

5 Open Questions

Before finishing this survey I summarize a set of important open questions.

***Improper* learning half-spaces is *definitely* NP-hard.** There are some evidences for this based upon some cryptographic and average case complexity assumptions, but no strong standalone proof yet.

No known efficient algorithm under uniform noise. Under the uniform distribution there is no known efficient algorithm known, with no approximation (i.e. $\mu = 1$).

***Proper* approximate algorithm.** The result presented by Daniely [27] is an *improper* algorithm. Is it possible to get similar results by restricting the algorithm to be proper?

Approximate solution in distributional assumption. Daniely [27] presented a relatively general (improper) result for $\mu > 1$, under the uniform distribution assumption. It is not clear how these results would extend to other distributional assumptions, e.g. log-concave, permutation-invariant.

References

[1] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[2] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[3] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2012.

[4] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge University Press, 2014.

[5] Dana Angluin and Philip Laird. Learning from noisy examples. *Machine Learning*, 2(4):343–370, 1988.

[6] Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[7] Michael Kearns and Ming Li. Learning in the presence of malicious errors. *SIAM Journal on Computing* (SICOMP), 22(4):807–837, 1993.

[8] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[9] Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine learning*, 2(4):285–318, 1988.

[10] Leonard Pitt and Leslie G Valiant. Computational limitations on learning from examples. *Journal of the ACM (JACM)*, 35(4):965–984, 1988.

[11] Michael J Kearns and Umesh Virkumar Vazirani. *An introduction to computational learning theory*. MIT press, 1994.

[12] Shai Ben-David, Nadav Eiron, and Philip M Long. On the difficulty of approximately maximizing agreements. *Journal of Computer and System Sciences*, 66(3):496–514, 2003.

[13] Richard M Karp. *Reducibility among combinatorial problems*. Springer, 1972.

[14] Adam Tauman Kalai, Adam R Klivans, Yishay Mansour, and Rocco A Servedio. Agnostically learning halfspaces. *SIAM Journal on Computing* (SICOMP), 37(6):1777–1805, 2008.

[15] Vitaly Feldman. Optimal hardness results for maximizing agreements with monomials. In *Computational Complexity, 2006. CCC 2006. Twenty-First Annual IEEE Conference on*, pages 9–pp. IEEE, 2006.

[16] A. Daniely and S. Shalev-Shwatz. Complexity theoretic limitations on learning dnf's. *arXiv preprint arXiv:1404.3378*, 2014.

[17] Adam Klivans and Pravesh Kothari. Embedding Hard Learning Problems Into Gaussian Space. In Klaus Jansen, José D. P. Rolim, Nikhil R. Devanur, and Cristopher Moore, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2014)*, volume 28 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 793–809, Dagstuhl, Germany, 2014. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[18] Amit Daniely. Complexity theoretic limitations on learning halfspaces. *Proc. 38th ACM symposium on Theory of computing* (STOC), 2015.

[19] Venkatesan Guruswami and Prasad Raghavendra. Hardness of learning halfspaces with noise. *SIAM Journal on Computing* (SICOMP), 39(2):742–765, 2009.

[20] Avrim Blum, Alan Frieze, Ravi Kannan, and Santosh Vempala. A polynomial-time algorithm for learning noisy linear threshold functions. *Algorithmica*, 22(1-2):35–52, 1998.

[21] Pranjal Awasthi, Maria-Florina Balcan, Nika Haghtalab, and Ruth Urner. Efficient learning of linear separators under bounded noise. In *Proc. 28th Conf. on Learning Theory* (COLT), pages 167–190, 2015.

[22] Pascal Massart and Élodie Nédélec. Risk bounds for statistical learning. *The Annals of Statistics*, pages 2326–2366, 2006.

[23] Michael J Kearns, Robert E Schapire, and Linda M Sellie. Toward efficient agnostic learning. *Mach. Learn. J.*, 17(2-3):115–141, 1994.

[24] Phil Long and Rocco Servedio. Algorithms and hardness results for parallel large margin learning. In *Advances in Neural Information Processing Systems*, pages 1314–1322, 2011.

[25] Aharon Birnbaum and Shai S Shwartz. Learning halfspaces with the zero-one loss: time-accuracy tradeoffs. In *Advances in Neural Information Processing Systems*, pages 926–934, 2012.

[26] Pranjal Awasthi, Maria Florina Balcan, and Philip M. Long. The power of localization for efficiently learning linear separators with noise. In *Proc. 36th ACM symposium on Theory of computing* (STOC), STOC '14, pages 449–458, New York, NY, USA, 2014. ACM.

[27] Amit Daniely. A ptas for agnostically learning halfspaces. *Proc. 27th Conf. on Learning Theory* (COLT), 2014.

[28] Shai Shalev-Shwartz, Ohad Shamir, and Karthik Sridharan. Learning kernel-based halfspaces with the 0-1 loss. *SIAM Journal on Computing*, 40(6):1623–1646, 2011.

[29] Yuchen Zhang, Jason D Lee, Martin J Wainwright, and Michael I Jordan. Learning halfspaces and neural networks with random initialization. *arXiv preprint arXiv:1511.07948*, 2015.

[30] Ilias Diakonikolas, Daniel M Kane, and Jelani Nelson. Bounded independence fools degree-2 threshold functions. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 11–20. IEEE, 2010.

[31] Peter L Bartlett, Olivier Bousquet, and Shahar Mendelson. Localized rademacher complexities. In *Computational Learning Theory*, pages 44–58. Springer, 2002.