From average case complexity to improper learning complexity*

[Extended Abstract]

Amit Daniely
Dept. of Mathematics, The Hebrew University, Jerusalem, Israel
amit.daniely@mail.huji.ac.il

Nati Linial
School of Computer Science and Engineering, The Hebrew University, Jerusalem, Israel.
nati@cs.huji.ac.il

Shai Shalev-Shwartz
School of Computer Science and Engineering, The Hebrew University, Jerusalem, Israel.
shais@cs.huji.ac.il

ABSTRACT

The basic problem in the PAC model of computational learning theory is to determine which hypothesis classes are efficiently learnable. There is presently a dearth of results showing hardness of learning problems. Moreover, the existing lower bounds fall short of the best known algorithms.

The biggest challenge in proving complexity results is to establish hardness of *improper learning* (a.k.a. representation independent learning). The difficulty in proving lower bounds for improper learning is that the standard reductions from **NP**-hard problems do not seem to apply in this context. There is essentially only one known approach to proving lower bounds on improper learning. It was initiated in [21] and relies on cryptographic assumptions.

We introduce a new technique for proving hardness of improper learning, based on reductions from problems that are hard on average. We put forward a (fairly strong) generalization of Feige's assumption [13] about the complexity of refuting random constraint satisfaction problems. Combining this assumption with our new technique yields far reaching implications. In particular,

- Learning DNF's is hard.
- Agnostically learning halfspaces with a constant approximation ratio is hard.
- Learning an intersection of $\omega(1)$ halfspaces is hard.

Categories and Subject Descriptors

F [**Theory of Computation**]: Computational Learning Theory

*A full version of this paper, containing proof details, can be found no the authors' webpages and on arXiv.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
STOC '14, May 31 - June 03 2014, New York, NY, USA
Copyright 2014 ACM 978-1-4503-2710-7/14/05 ...$15.00
http://dx.doi.org/10.1145/2591796.2591820.

Keywords

Hardness of improper learning, DNFs, Halfspaces, Average Case complexity, CSP problems, Resolution lower bounds

1. INTRODUCTION

Valiant's celebrated *probably approximately correct* (=PAC) model [34] of machine learning led to an extensive research that yielded a whole scientific community devoted to computational learning theory. In the PAC learning model, a learner is given an oracle access to randomly generated samples $(X, Y) \in \mathcal{X} \times \{0,1\}$ where X is sampled from some *unknown* distribution $\mathcal{D}$ on $\mathcal{X}$ and $Y = h^*(X)$ for some *unknown* function $h^* : \mathcal{X} \to \{0,1\}$. Furthermore, it is assumed that h^* comes from a predefined *hypothesis class* $\mathcal{H}$, consisting of $0,1$ valued functions on $\mathcal{X}$. The learning problem defined by $\mathcal{H}$ is to find a function $h : \mathcal{X} \to \{0,1\}$ that minimizes $\mathrm{Err}_{\mathcal{D}}(h) := \Pr_{X\sim\mathcal{D}}(h(X) \neq h^*(X))$. For concreteness' sake we take $\mathcal{X} = \{\pm 1\}^n$, and we consider the learning problem tractable if there is an algorithm that on input ϵ, runs in time $\mathrm{poly}(n, 1/\epsilon)$ and outputs, w.h.p., a hypothesis h with $\mathrm{Err}(h) \leq \epsilon$.

Assuming $\mathbf{P} \neq \mathbf{NP}$, the status of most basic *computational* problems is fairly well understood. In a sharp contrast, 30 years after Valiant's paper, the status of most basic *learning* problems is still wide open – there is a huge gap between the best algorithms' performance and hardness results:

- No known algorithms can learn depth 2 circuits, i.e., DNF formulas. In contrast, we can only rule out learning of circuits of depth d, for some unspecified constant d [22]. This result is based on a relatively strong assumption (a certain subexponential lower bound on factoring Blum integers). Under more standard assumptions (RSA in secure), the best we can do is rule out learning of depth $\log n$ circuits [21].
- It is possible to agnostically learn halfspaces with an approximation ratio of $O\left(\frac{n}{\log n}\right)$. On the other hand, the best known lower bound only rules out exact agnostic learning ([15], based on [27], under the assumption that the $\tilde{O}\left(n^{1.5}\right)$ unique shortest vector problem is hard).
- No known algorithm learns intersections of 2 halfspaces, whereas we can only rule out learning intersections of polynomially many halfspaces ([27], assuming

that $\tilde{O}\left(n^{1.5}\right)$-uSVP is hard).

The crux of the matter, leading to this state of affairs, has to do with the learner's freedom to return *any* hypothesis. A learner who may return hypotheses outside the class $\mathcal{H}$ is called an *improper learner*. This additional freedom makes such algorithms potentially more powerful than proper learners. On the other hand, this added flexibility makes it difficult to apply standard reductions from **NP**-hard problems. Indeed, there was no success so far in proving intractability of a learning problem based on **NP**-hardness. Moreover, as Applebaum, Barak and Xiao [2] showed, many standard ways to do so are doomed to fail, unless the polynomial hierarchy collapses.

The vast majority of existing lower bounds on learning utilize the crypto-based argument, suggested in [21]. Roughly speaking, to prove that a certain learning problem is hard, one starts with a certain collection of functions, that by assumption are one-way trapdoor permutations. This immediately yields some hard (usually artificial) learning problem. The final step is to reduce this artificial problem to some natural learning problem.

Unlike the difficulty in establishing lower bounds for improper learning, the situation in *proper* learning is much better understood. Usually, hardness of proper learning is proved by showing that it is **NP**-hard to distinguish a realizable sample from an unrealizable sample. I.e., it is hard to tell whether there is some hypothesis in $\mathcal{H}$ which has zero error on a given sample. This, however, does not suffice for the purpose of proving lower bounds on improper learning, because it might be the case that the learner finds a hypothesis (not from $\mathcal{H}$) that does not err on the sample even though no $h \in \mathcal{H}$ can accomplish this. In this paper we present a new methodology for proving hardness of improper learning. Loosely speaking, we show that improper learning is impossible provided that it is hard to distinguish a realizable sample from a *randomly generated* unrealizable sample.

Feige [13] conjectured that random 3-SAT formulas are hard to refute. He derived from this assumption certain hardness of approximation results, which are not known to follow from $\mathbf{P} \neq \mathbf{NP}$. We put forward a (fairly strong) assumption, generalizing Feige's assumption to certain predicates other that 3-SAT. Under this assumption, we show:

1. Learning DNF's is hard.
2. Agnostically learning halfspaces with a constant approximation ratio is hard, even over the boolean cube.
3. Learning intersection of $\omega(1)$ halfspaces is hard, even over the boolean cube.
4. Learning finite automata is hard.
5. Learning parity is hard.

We note that result 4 can be established using the cryptographic technique [21]. Result 5 is often taken as a hardness *assumption*. We also conjecture that under our generalization of Feige's assumption it is hard to learn intersections of even constant number of halfspaces. We present a possible approach to the case of four halfspaces. To the best of our knowledge, these results easily imply most existing lower bounds for improper learning.

1.1 Comparison to the cryptographic technique

There is a crucial reversal of order that works in our favour. To lower bound improper learning, we actually need much less than what is needed in cryptography, where a problem and a distribution on instances are appropriate if they fool *every algorithm*. In contrast, here we are presented with *a concrete* learning algorithms and we devise a problem and a distribution on instances that fail it.

Second, cryptographic assumptions are often about the hardness of number theoretic problems. In contrast, the average case assumptions presented here are about CSP problems. The proximity between CSP problems and learning problems is crucial for our purposes: Since distributions are very sensitive to gadgets, reductions between average case problems are much more limited than reductions between worst case problems.

1.2 On the role of average case complexity

A key question underlying the present study and several additional recent papers is what can be deduced from the average case hardness of specific problems. Hardness on average is crucial for cryptography, and the security of almost all modern cryptographic systems hinges on the average hardness of certain problems, often from number theory. As shown by Kearns and Valiant [21], the very same hardness on average assumptions can be used to prove hardness of improper PAC learning of some hypothesis classes.

Beyond these classic results, several recent works, starting from Feige's seminal work [13], show that average case hardness assumptions lead to dramatic consequences in complexity theory. The main idea of [13] is to consider two possible avenues for progress beyond the classic uses of average hardness: (i) Derive hardness in additional domains, (ii) Investigate the implications of hardness-on-average of other problems. For example, what are the implications of average hardness of 3-SAT? What about other CSP problems?

Feige [13] and then [1, 4] show that average case hardness of CSP problems have surprising implications in hardness of approximation, much beyond the consequences of standard complexity assumptions, or even cryptographic assumptions. Recently, [8] and [12] show that hardness on average of planted clique and 3-SAT have implications in learning theory, in the specific context of computational-sample tradeoffs. In particular, they show that in certain learning tasks (sparse PCA and learning halfspaces over sparse vectors) more data can be leveraged to speed up computation. As we show here, average case hardness of CSP problems has implications even on the hardness of very fundamental tasks in learning theory. Namely, determining the tractability of PAC learning problems, most of which are presently otherwise inaccessible.

2. NOTATIONS

For standard definitions and terminology of learning theory and CSP problems, the reader is referred to the full version of this paper (that can be found on the authors' websites). Let $P : \{\pm 1\}^K \to \{0,1\}$ be some boolean predicate. A *P-constraint* with n variables is a function $C : \{\pm 1\}^n \to \{0,1\}$ of the form $C(x) = P(j_1 x_{i_1}, \ldots, j_K x_{i_K})$ for $j_l \in \{\pm 1\}$ and K distinct $i_l \in [n]$. An instance to the problem CSP(P) is a collection $J = \{C_1, \ldots, C_m\}$ of P-constraints and the objective is to find an assignment $x \in \{\pm 1\}^n$ that maximizes the fraction of satisfied constraints (i.e., constraints with $C_i(x) = 1$). The *value* of the instance J, denoted

$\mathrm{VAL}(J)$, is the maximal fraction of constraints that can be simultaneously satisfied. If $\mathrm{VAL}(J) = 1$, we say that J is satisfiable.

For $1 \geq \alpha > \beta > 0$, the problem $\mathrm{CSP}^{\alpha,\beta}(P)$ is the decision promise problem of distinguishing between instances to $\mathrm{CSP}(P)$ with value $\geq \alpha$ and instances with value $\leq \beta$. Denote $\underline{\mathrm{VAL}}(P) = \mathbb{E}_{x \sim \mathrm{Uni}(\{\pm 1\}^K)} P(x)$. We say that P is *approximation resistant* if, for every $\epsilon > 0$, the problem $\mathrm{CSP}^{1-\epsilon,\underline{\mathrm{VAL}}(P)+\epsilon}(P)$ is **NP**-hard. We say that P is *approximation resistant on satisfiable instances* if, for every $\epsilon > 0$, the problem $\mathrm{CSP}^{1,\underline{\mathrm{VAL}}(P)+\epsilon}(P)$ is **NP**-hard. We say that P is *heredity approximation resistant on satisfiable instances* if every predicate that is implied by P (i.e., every predicate $P' : \{\pm 1\}^K \to \{0,1\}$ that satisfies $\forall x,\ P(x) \Rightarrow P'(x)$) is approximation resistant on satisfiable instances. Similarly, we define the notion of *heredity approximation resistance*.

Fix $1 \geq \alpha > \underline{\mathrm{VAL}}(P)$ and a function $m(n)$. We denote by $\mathrm{CSP}^{\alpha,\mathrm{rand}}_{m(n)}(P)$ the problem of distinguishing between instances with value $\geq \alpha$ and instances with $m(n)$ random (uniform and independent) constraints. Throughout, we only consider such problem if m grows fast enough, so that the probability that a random instance with $m(n)$ constraints will have value $\geq \alpha$ is $o_n(1)$.

3. THE METHODOLOGY

We first discuss the methodology in the realm of realizable learning. Treatment of agnostic learning can be found in the full version of this paper. To motivate the approach, recall how one usually proves that a class $\mathcal{H}$ cannot be efficiently *properly* learnable. Let $\Pi(\mathcal{H})$ be the problem of distinguishing between an $\mathcal{H}$-realizable sample S and one with $\mathrm{Err}_S(\mathcal{H}) \geq \frac{1}{4}$. If $\mathcal{H}$ is efficiently *properly* learnable then this problem is in[1] **RP**: To solve $\Pi(\mathcal{H})$, we simply invoke a proper learning algorithm $\mathcal{A}$ that efficiently learns $\mathcal{H}$, with examples drawn uniformly from S. Let h be the output of $\mathcal{A}$. Since $\mathcal{A}$ learns $\mathcal{H}$, if S is a realizable sample, then $\mathrm{Err}_S(h)$ is small. On the other hand, if $\mathrm{Err}_S(\mathcal{H}) \geq \frac{1}{4}$ then, *since* $h \in \mathcal{H}$, $\mathrm{Err}_S(h) \geq \frac{1}{4}$. This gives an efficient way to decide whether S is realizable. We conclude that if $\Pi(\mathcal{H})$ is **NP**-hard, then $\mathcal{H}$ is not efficiently learnable, unless $\mathbf{NP} = \mathbf{RP}$.

However, this argument does not rule out the possibility that $\mathcal{H}$ is still learnable by an *improper* algorithm. Suppose that $\mathcal{A}$ efficiently and improperly learns $\mathcal{H}$. If we try to use the above argument to solve $\Pi(\mathcal{H})$, we get stuck – suppose that S is a sample and we invoke $\mathcal{A}$ on it, to get a hypothesis h. As before, if S is realizable, $\mathrm{Err}_S(h)$ is small. However, if S is not realizable, since h not necessarily belongs to $\mathcal{H}$, it still might be the case that $\mathrm{Err}_S(h)$ is small. Therefore, the argument fails. We emphasize that this is not only a mere weakness of the argument – there are classes for which $\Pi(\mathcal{H})$ is **NP**-hard, but yet, they are learnable by an improper algorithm[2]. More generally, Applebaum et al [2] indicate that it is unlikely that hardness of improper learning can be based on standard reductions from **NP**-hard problems, as the one described here.

We see that it is not clear how to establish hardness of improper learning based on the hardness of distinguishing between a realizable and an unrealizable sample. The core problem is that even if S is not realizable, the algorithm might still return a good hypothesis. The crux of our new technique is the observation that if S is *randomly generated* unrealizable sample then even improper algorithm cannot return a hypothesis with a small empirical error. The point is that the returned hypothesis is determined solely by the examples that $\mathcal{A}$ sees and its random bits. Therefore, if $\mathcal{A}$ is an efficient algorithm, the number of hypotheses it might return cannot be too large. Hence, if S is "random enough", it likely to be far from all these hypotheses, in which case the hypothesis returned by $\mathcal{A}$ would have a large error on S.

We now formalize this idea. Denote $\mathcal{Z} = \mathcal{X}_n \times \{0,1\}$ and let $\mathcal{D} = \{\mathcal{D}_n^{m(n)}\}_n$ be an ensemble of distributions, such that $\mathcal{D}_n^{m(n)}$ is a distribution on $\mathcal{Z}_n^{m(n)}$ and $m(n) \leq \mathrm{poly}(n)$. Think of $\mathcal{D}_n^{m(n)}$ as a distribution that generates samples that are far from being realizable by $\mathcal{H}$. We say that it is hard to distinguish $\mathcal{D}$-random sample from a realizable sample if there is no efficient randomized algorithm $\mathcal{A}$ with the following properties:

- For every realizable sample $S \in \mathcal{Z}_n^{m(n)}$,
$$\Pr_{\text{internal coins of } \mathcal{A}}(\mathcal{A}(S) = \text{“}\textit{realizable}\text{”}) \geq \frac{3}{4}$$
- If $S \sim \mathcal{D}_n^{m(n)}$, then with probability $1 - o_n(1)$ over the choice of S, it holds that
$$\Pr_{\text{internal coins of } \mathcal{A}}(\mathcal{A}(S) = \text{“}\textit{unrelizable}\text{”}) \geq \frac{3}{4}$$

For functions $p, \epsilon : \mathbb{N} \to (0,\infty)$, we say that $\mathcal{D}$ is $(p(n),\epsilon(n))$-*scattered* if, for large enough n, it holds that for every function $f : \mathcal{X}_n \to \{0,1\}$, $\Pr_{S \sim \mathcal{D}_n^{m(n)}}(\mathrm{Err}_S(f) \leq \epsilon(n)) \leq 2^{-p(n)}$.

THEOREM 3.1. *Every hypothesis class that satisfies the following condition is not efficiently learnable. There exists $\beta > 0$ such that for every $c > 0$ there is an (n^c, β)-scattered ensemble $\mathcal{D}$ for which it is hard to distinguish between a $\mathcal{D}$-random sample and a realizable sample.*

REMARK 3.2. *The theorem and the proof below work verbatim if we replace β by $\beta(n)$, provided that $\beta(n) > n^{-a}$ for some $a > 0$.*

PROOF. Let $\mathcal{H}$ be the hypothesis class in question and suppose toward a contradiction that algorithm $\mathcal{L}$ learns $\mathcal{H}$ efficiently. Let $M(n, 1/\epsilon, 1/\delta)$ be the maximal number of random bits used by $\mathcal{L}$ when run on the input n, ϵ, δ. This includes both the bits describing the examples produced by the oracle and "standard" random bits. Since $\mathcal{L}$ is efficient, $M(n, 1/\epsilon, 1/\delta) < \mathrm{poly}(n, 1/\epsilon, 1/\delta)$. Define $q(n) = M(n, 1/\beta, 4) + n$. By assumption, there is a $(q(n), \beta)$-scattered ensemble $\mathcal{D}$ for which it is hard to distinguish a $\mathcal{D}$-random sample from a realizable sample. Consider the algorithm $\mathcal{A}$ defined below. On input $S \in \mathcal{Z}_n^{m(n)}$,

1. Run $\mathcal{L}$ with parameters n, β and $\frac{1}{4}$, such that the examples' oracle generates examples by choosing a random example from S.
2. Let h be the hypothesis that $\mathcal{L}$ returns. If $\mathrm{Err}_S(h) \leq \beta$, output "realizable". Otherwise, output "unrealizable".

[1] The reverse direction is almost true: If the search version of this problem can be solved in polynomial time, then $\mathcal{H}$ is efficiently learnable.

[2] This is true, for example, for the class of DNF formulas with 3 DNF clauses.

Next, we derive a contradiction by showing that $\mathcal{A}$ distinguishes a realizable sample from a $\mathcal{D}$-random sample. Indeed, if the input S is realizable, then $\mathcal{L}$ is guaranteed to return, w.p. $\geq 1 - \frac{1}{4}$, a hypothesis $h : \mathcal{X}_n \to \{0,1\}$ with $\mathrm{Err}_S(h) \leq \beta$. Therefore, w.p. $\geq \frac{3}{4}$ $\mathcal{A}$ will output "realizable".

What if S is drawn from $\mathcal{D}_n^{m(n)}$? Let $\mathcal{G} \subset \{0,1\}^{\mathcal{X}_n}$ be the collection of functions that $\mathcal{L}$ might return when run with parameters $n, \epsilon(n)$ and $\frac{1}{4}$. We note that $|\mathcal{G}| \leq 2^{q(n)-n}$, since each hypothesis in $\mathcal{G}$ can be described by $q(n) - n$ bits. Namely, the random bits that $\mathcal{L}$ uses and the description of the examples sampled by the oracle. Now, since $\mathcal{D}$ is $(q(n), \beta)$-scattered, the probability that $\mathrm{Err}_S(h) \leq \beta$ for some $h \in \mathcal{G}$ is at most $|\mathcal{G}|2^{-q(n)} \leq 2^{-n}$. It follows that the probability that $\mathcal{A}$ responds "realizable" is $\leq 2^{-n}$. This leads to the desired contradiction and concludes our proof. □

4. THE STRONG RANDOM CSP ASSUMPTION

In this section we put forward and discuss a new assumption that we call "the strong random CSP assumption" or SRCSP for short. It generalizes Feige's assumption [13], as well as the assumption of Barak, Kindler and Steurer [4]. This new assumption, together with the methodology described in section 3, are used to establish lower bounds for improper learning. Admittedly, our assumption is strong, and an obvious quest, discussed in the end of this section is to find ways to derive similar conclusions from weaker assumptions.

The SRCSP assumption claims that for certain predicates $P : \{\pm 1\}^K \to \{0,1\}, d > 0$ and $\alpha > 0$, the decision problem $\mathrm{CSP}^{\alpha,\mathrm{rand}}_{n^d}(P)$ is intractable. We first consider the case $\alpha = 1$. To reach a plausible assumption, let us first discuss Feige's assumption, and the existing evidence for it. Denote by $\mathrm{SAT}_3 : \{\pm 1\}^3 \to \{0,1\}$ the 3-SAT predicate $\mathrm{SAT}_3(x_1, x_2, x_3) = x_1 \vee x_2 \vee x_3$.

ASSUMPTION 4.1 (FEIGE). *For every sufficiently large $C > 0$, $\mathrm{CSP}^{1,\mathrm{rand}}_{C \cdot n}(\mathrm{SAT}_3)$ is intractable.*

Let us briefly summarize the evidence for this assumption.

Hardness of approximation. Feige's conjecture can be viewed as a strengthening of Hastad's celebrated result [17] that SAT_3 is approximation resistant on satisfiable instances. Hastad's result implies that under $\mathbf{P} \neq \mathbf{NP}$, it is hard to distinguish satisfiable instances to $\mathrm{CSP}(\mathrm{SAT}_3)$ from instances with value $\leq \frac{7}{8} + \epsilon$. The collection of instances with value $\leq \frac{7}{8} + \epsilon$ includes most random instances with $C \cdot n$ clauses for sufficiently large C. Feige's conjecture says that the problem remains intractable even when restricted to these random instances.

We note that approximation resistance on satisfiable instances is a necessary condition for the validity of Feige's assumption. Indeed, for large enough $C > 0$, with probability $1 - o_n(1)$, the value of a random instance to $\mathrm{CSP}(\mathrm{SAT}_3)$ is $\leq \frac{7}{8} + \epsilon$. Therefore, tractability of $\mathrm{CSP}^{1,\frac{7}{8}+\epsilon}(\mathrm{SAT}_3)$ would lead to tractability of $\mathrm{CSP}^{1,\mathrm{rand}}_{C \cdot n}(\mathrm{SAT}_3)$.

Performance of known algorithms. The problem of refuting random 3-SAT formulas has been extensively studied. The best known algorithms [14] can refute random instances with $\Omega\left(n^{1.5}\right)$ random constraints. Moreover resolution lower bounds [7] show that many algorithms run for exponential time when applied to random instances with $O\left(n^{1.5-\epsilon}\right)$ constraints.

We aim to generalize Feige's assumption in two aspects – (i) To predicates other than SAT_3, and (ii) To problems with super-linearly many constraints. Consider the problem $\mathrm{CSP}^{1,\mathrm{rand}}_{m(n)}(P)$ for some predicate $P : \{\pm 1\}^K \to \{0,1\}$. As above, the intractability of $\mathrm{CSP}^{1,\mathrm{rand}}_{m(n)}(P)$ strengthens the claim that P is approximation resistant on satisfiable instances. Also, for $\mathrm{CSP}^{1,\mathrm{rand}}_{m(n)}(P)$ to be hard, it is necessary that P is approximation resistant on satisfiable instances. In fact, it is easy to see that if $P' : \{\pm 1\}^K \to \{0,1\}$ is implied by P, then the problem $\mathrm{CSP}^{1,\mathrm{rand}}_{m(n)}(P)$ can be easily reduced to $\mathrm{CSP}^{1,\mathrm{rand}}_{m(n)}(P')$. Therefore, to preserve the argument of the first evidence of Feige's conjecture, it is natural to require that P is *heredity* approximation resistant on satisfiable instances.

Next, we discuss what existing algorithms can do. The best known algorithms for the predicate $\mathrm{SAT}_K(x_1, \ldots, x_K) = \vee_{i=1}^K x_i$ only refute random instances with $\Omega\left(n^{\lfloor \frac{K}{2} \rfloor}\right)$ constraints [11]. This gives some evidence that it becomes harder to refute random instances of $\mathrm{CSP}(P)$ as the number of variables grows. Namely, that many random constraints are needed to efficiently refute random instances. Of course, some care is needed with counting the "actual" number of variables. Clearly, only *certain* predicates have been studied so far. Therefore, to reach a plausible assumption, we consider the *resolution refutation complexity* of random instances to $\mathrm{CSP}(P)$. And consequently, also the performance of a large class of algorithms, including Davis-Putnam style (DPLL) algorithms.

DPLL algorithms have been subject to an extensive study, both theoretical and empirical. Due to the central place that they occupy, much work has been done since the late 80's, to prove lower bounds on their performance in refuting random K-SAT formulas. These works relied on the fact that these algorithms implicitly produce a resolution refutation during their execution. Therefore, to derive a lower bound on the run time of these algorithms, exponential lower bounds were established on the resolution complexity of random instances to $\mathrm{CSP}(\mathrm{SAT}_K)$. These lower bounds provide support to the belief that it is hard to refute not-too-dense random K-SAT instances.

We define the 0-*variability,* $\mathrm{VAR}_0(P)$, of a predicate P as the smallest cardinality of a set of P's variables such that there is an assignment to these variables for which $P(x) = 0$, regardless of the values assigned to the other variables. By a simple probabilistic argument, a random $\mathrm{CSP}(P)$ instance with $\Omega\left(n^r\right)$ constraints, where $r = \mathrm{VAR}_0(P)$ is almost surely unsatisfiable with a resolution proof of constant size. Namely, w.p. $1 - o_n(1)$, there are 2^r constraints that are inconsistent, since some set of r variables appears in all 2^r possible ways in the different clauses. On the other hand, we show in the full version of this paper, a random $\mathrm{CSP}(P)$ problem with $O\left(n^{c \cdot r}\right)$ constraints has w.h.p. exponential resolution complexity. Here $c > 0$ is an absolute constant. Namely,

THEOREM 4.2. *There is a constant $C > 0$ such that for every $d > 0$ and every predicate P with $\mathrm{VAR}_0(P) \geq C \cdot d$, the following holds. With probability $1 - o_n(1)$, a random instance of $\mathrm{CSP}(P)$ with n variables and n^d constraints has resolution refutation length $\geq 2^{\Omega\left(\sqrt{n}\right)}$.*

To summarize, we conclude that the parameter $\mathrm{VAR}_0(P)$ controls the resolution complexity of random instances to $\mathrm{CSP}(P)$. In light of the above discussion, we raise the following assumption.

ASSUMPTION 4.3 (SRCSP – PART 1). *There is a function $f : \mathbb{N} \to \mathbb{N}$ such that the following holds. Let P be a predicate that is heredity approximation resistant on satisfiable instances with $\mathrm{VAR}_0(P) \geq f(d)$. Then, it is hard to distinguish between satisfiable instances of $\mathrm{CSP}(P)$ and random instances with n^d constraints.*

Next, we motivate a variant of the above assumption, that accommodates predicates that are not heredity approximation resistant. A celebrated result of Raghavendra [31] shows that under the UGC [23], a certain SDP-relaxation-based algorithm is (worst case) optimal for $\mathrm{CSP}(P)$, for every predicate P. Barak et al. [4] conjectured that this algorithm is optimal even on random instances. They considered the performance of this algorithm on random instances and purposed the following assumption, which they called the "random CSP hypothesis". Define $\overline{\mathrm{VAL}}(P) = \max_{\mathcal{D}} \mathbb{E}_{x\sim\mathcal{D}} P(x)$, where the maximum is taken over all pairwise uniform distributions[3] on $\{\pm 1\}^K$.

ASSUMPTION 4.4 (RSCP). *For every $\epsilon > 0$ and sufficiently large $C > 0$, it is hard to distinguish instances with value $\geq \overline{\mathrm{VAL}}(P) - \epsilon$ from random instances with $C \cdot n$ constraints.*

Here we generalize the RCSP assumption to random instances with much more than $C \cdot n$ constraints. As in assumption 4.3, the 0-variability of P serves to quantify the number of random constraints needed to efficiently show that a random instance has value $< \overline{\mathrm{VAL}}(P) - \epsilon$.

ASSUMPTION 4.5 (SRSCP - PART 2). *There is a function $f : \mathbb{N} \to \mathbb{N}$ such that for every predicate P with $\mathrm{VAR}_0(P) \geq f(d)$ and for every $\epsilon > 0$, it is hard to distinguish between instances with value $\geq \overline{\mathrm{VAL}}(P) - \epsilon$ and random instances with n^d constraints.*

TERMINOLOGY 4.6. *A computational problem is* SRCSP*-hard if its tractability contradicts assumption 4.3 or 4.5.*

4.1 Toward weaker assumptions

The SRCSP assumption is strong. It is highly desirable to arrive at similar conclusions from weaker assumptions. A natural possibility is the SRCSP assumption, restricted to SAT:

ASSUMPTION 4.7. *There is a function $f : \mathbb{N} \to \mathbb{N}$ such that for every $K \geq f(d)$, it is hard to distinguish satisfiable instances of $\mathrm{CSP}(\mathrm{SAT}_K)$ from random instances with n^d constraints.*

We are quite optimistic regarding the success of this direction: Our lower bounds use the SRCSP-assumption only for certain predicates, and do not need the full power of the assumption. Moreover, for the hypothesis classes of DNF's, intersection of halfspaces, and finite automata, these predicates are somewhat arbitrary. In [13], it is shown that for predicates of arity 3, assumption 4.5 is implied by the same assumption restricted to the SAT predicate. This gives a hope to prove, based on assumption 4.7, that the SRCSP-assumption is true for predicates that are adequate to our needs.

[3]A distribution is *pairwise uniform* if, for every pair of coordinates, the distribution induced on these coordinates is uniform.

5. SUMMARY OF RESULTS

5.1 Learning DNF's

A DNF *clause* is a conjunction of literals. A DNF *formula* is a disjunction of DNF clauses. Each DNF formula over n variables induces a function on $\{\pm 1\}^n$. We define the size of a DNF clause as the number of its literals and the size of a DNF formula as the sum of the sizes of its clauses. For a function $q : \mathbb{N} \to \mathbb{N}$, denote by $\mathrm{DNF}_{q(n)}$ the hypothesis class of functions over $\{\pm 1\}^n$ that can be realized by DNF formulas of size at most $q(n)$. Also, let $\mathrm{DNF}^{q(n)}$ be the hypothesis class of functions over $\{\pm 1\}^n$ that that can be realized by DNF formulas with at most $q(n)$ clauses. Since each clause is of size at most n, $\mathrm{DNF}^{q(n)} \subset \mathrm{DNF}_{nq(n)}$.

Already in [34], it is shown that for constant q, learning DNFs with $\leq q$ clauses is tractable. The running time of the algorithm is, however, exponential in q. Also, the algorithm is improper. For general polynomial-size DNF's, the best known result [26] shows learnability in time $\frac{1}{\epsilon} \cdot 2^{\tilde{O}\left(n^{\frac{1}{3}}\right)}$. Better running times (quasi-polynomial) are known under distributional assumptions [28, 29].

As for lower bounds, *properly* learning DNF's is known to be hard [30]. However, proving hardness of improper learning of polynomial DNF's has remained a major open question in computational learning theory. Noting that DNF clauses coincide with depth 2 circuits, a natural generalization of DNF's is circuits of small depth. For such classes, certain lower bounds can be obtained using the cryptographic technique. Kharitonov [22] has shown that a certain subexponential lower bound on factoring Blum integers implies hardness of learning circuits of depth d, for some unspecified constant d. Under more standard assumptions (that the RSA cryptosystem is secure), best lower bounds [21] only rule out learning of circuits of depth $\log(n)$.

As mentioned, for a constant q, the class DNF^q is efficiently learnable. We show that for every super constant $q(n)$, it is SRCSP-hard to learn $\mathrm{DNF}^{q(n)}$:

THEOREM 5.1. *If $\lim_{n\to\infty} q(n) = \infty$ then learning $\mathrm{DNF}^{q(n)}$ is* SRCSP*-hard.*

Since $\mathrm{DNF}^{q(n)} \subset \mathrm{DNF}_{nq(n)}$, we immediately conclude that learning DNF's of size, say, $\leq n\log(n)$, is SRCSP-hard. By a simple scaling argument, we obtain an even stronger result:

COROLLARY 5.2. *For every $\epsilon > 0$, it is* SRCSP*-hard to learn $\mathrm{DNF}_{n^\epsilon}$.*

REMARK 5.3. *Following the Boosting argument of Schapire [33], hardness of improper learning of a class $\mathcal{H}$ immediately implies that for every $\epsilon > 0$, there is no efficient algorithm that when running on a distribution that is realized by $\mathcal{H}$, guaranteed to output a hypothesis with error $\leq \frac{1}{2} - \epsilon$. Therefore, hardness results of improper learning are very strong, in the sense that they imply that the algorithm that just makes a random guess for each example, is essentially optimal.*

5.2 Agnostically learning halfspaces

Let HALFSPACES be the hypothesis class of halfspaces over $\{-1,1\}^n$. Namely, for every $w \in \mathbb{R}^n$ we define $h_w : \{\pm 1\}^n \to \{0,1\}$ by $h_w(x) = \text{sign}(\langle w, x\rangle)$, and let HALFSPACES $= \{h_w \mid w \in \mathbb{R}^n\}$. We note that usually halfspaces are defined over $\mathbb{R}^n$, but since we are interested in lower bounds, looking on this more restricted class just make the lower bounds stronger.

The problem of learning halfspaces is as old as the field of machine learning, starting with the perceptron algorithm [32], through the modern SVM [35]. As opposed to learning DNF's, learning halfspaces in the realizable case is tractable. However, in the agnostic PAC model, the best currently known algorithm for learning halfspaces runs in time exponential in n and the best known approximation ratio of polynomial time algorithms is $O\left(\frac{n}{\log(n)}\right)$. Better running times (usually of the form $n^{\text{poly}(\frac{1}{\epsilon})}$) are known under distributional assumptions (e.g. [20]).

Proper agnostic learning of halfspaces was shown to be hard to approximate within a factor of $2^{\log^{1-\epsilon}(n)}$ [3]. Using the cryptographic technique, improper learning of halfspaces is known to be hard, under a certain cryptographic assumption regarding the shortest vector problem ([15], based on [27]). No hardness results are known for *approximately and improperly* learning halfspaces.

THEOREM 5.4. *For every constant $\alpha \geq 1$, it is* SRCSP*-hard to approximately agnostically learn* HALFSPACES *with an approximation ratio of α.*

5.3 Learning intersection of halfspaces

For a function $q : \mathbb{N} \to \mathbb{N}$, we let $\text{INTER}_{q(n)}$ be the hypothesis class of intersection of $\leq q(n)$ halfspaces. That is, $\text{INTER}_{q(n)}$ consists of all functions $f : \{\pm 1\}^n \to \{0,1\}$ for which there exist $w_1, \ldots w_k \in \mathbb{R}^n$ such that $f(x) = 1$ if and only if $\forall i, \langle w_i, x\rangle > 0$.

Beside being a natural generalization of learning halfspaces, the importance of INTER_q stems from its connection to neural networks [9]. A neural network composed of layers, each of which composed of nodes. The first layer consists of n nodes, containing the input. The nodes in the rest of the layers calculates a value according to a halfspace [4] applied on the nodes in the previous layer. The final layer consists of a single node, holding the output of the whole network. Very simple neural networks can realize INTER_q: those with only an input layer, a single hidden layer consisting of only $q(n)$ nodes, and an output layer. Therefore, lower bounds on improperly learning intersection of halfspaces implies lower bounds on improper learning of neural networks.

Exact algorithms for learning $\text{INTER}_{q(n)}$ run in time exponential in n. Better running times (usually of the form $n^{\text{poly}(\frac{1}{\epsilon})}$) are known under distributional assumptions (e.g. [25]). It is known that properly learning intersection of even 2 halfspaces is hard [24]. For improper learning, Klivans and Sherstov [27] have shown that learning an intersection of polynomially many half spaces is hard, under a certain assumption about the shortest vector problem. Noting that every DNF formula with $q(n)$ clauses is in fact the complement of an intersection of $q(n)$ halfspaces[5], we conclude from theorem 5.1 that intersection of every super constant number of halfsapces is hard.

THEOREM 5.5. *If $\lim_{n\to\infty} q(n) = \infty$ then learning* $\text{INTER}_{q(n)}$ *is* SRCSP*-hard.*

In the full version of this paper we describe a route toward the result that learning INTER_4 is SRCSP-hard.

5.4 Additional results

We show that learning the class of finite automata of polynomial size is SRCSP-hard. Hardness of this class can also be derived using the cryptographic technique, based on the assumption that the RSA cryptosystem is secure [21]. We also show that agnostically learning parity with any constant approximation ratio is SRCSP-hard. Parity is not a very interesting class for practical machine learning. However, this class is related to several other problems in complexity [10]. We note that hardness of agnostically learning parity, even in a more relaxed model than the agnostic PAC model (called the random classification noise model), is a well accepted hardness assumption.

We also prove lower bounds on the size of a resolution refutation for random CSP instances. In addition, we show that unless the polynomial hierarchy collapses, there is no "standard reduction" from an **NP**-hard problem (or a **CoNP**-hard problem) to random CSP problems.

5.5 On the proofs

Next, we outline the proof for DNFs. The proof for halfspaces and parities is similar. Given $c > 0$, we start with a predicate $P : \{\pm 1\}^K \to \{0,1\}$, for which the problem $\text{CSP}^{1,\text{rand}}_{n^c}(P)$ is hard according to the SRCSP-assumption, and reduce it to the problem of distinguishing a $(\Omega(n^c), \frac{1}{5})$-scattered sample from a realizable sample. Since c is arbitrary, the theorem follows from theorem 3.1.

The reduction is performed as follows. Consider the problem $\text{CSP}(P)$. Each assignment naturally defines a function from the collection of P-constraints to $\{0,1\}$. Hence, if we think about the constraints as instances and about the assignments as hypotheses, the problem $\text{CSP}(P)$ turns into some kind of a learning problem. However, in this interpretation, all the instances we see have positive labels (since we seek an assignment that satisfies as many instances as possible). Therefore, the problem $\text{CSP}^{1,\text{rand}}_{n^c}(P)$ results in "samples" which are not scattered at all.

To overcome this, we show that the analogous problem to $\text{CSP}^{1,\text{rand}}_{n^c}(P)$, where $(\neg P)$-constraints are also allowed, is hard as well (using the assumption on the hardness of $\text{CSP}^{1,\text{rand}}_{n^c}(P)$). The hardness of the modified problem can be shown by relying on the special predicate we work with. This predicate was defined in the recent work of Huang [19], and it has the property of being heredity approximation resistant, even though $|P^{-1}(1)| \leq 2^{O(K^{1/3})}$.

At this point, we have an (artificial) hypothesis class which is SRCSP-hard to learn by theorem 3.1. In the next and final step, we show that this class can be efficiently realized by DNFs with $\omega(1)$ clauses. The reduction uses the fact that every boolean function can be expressed by a DNF formula (of possibly exponential size). Therefore, P can be expressed

[4] Or a "soft" halfspace obtained by replacing the sign function with a sigmoidal function.

[5] In the definition of INTER, we considered halfspaces with no threshold, while halfspaces corresponding to DNFs do have a threshold. This can be standardly handled by padding the examples with a single coordinate of value 1.

by a DNF formula with 2^K clauses. Based on this, we show that each hypothesis in our artificial class can be realized by a DNF formula with 2^K clauses, which establishes the proof.

The results about learning automata and intersection of $\omega(1)$ halfspaces follow from the result about DNFs: We show that these classes can efficiently realize the class of DNFs with $\omega(1)$ clauses. In the full version of this paper we suggest a route toward the result that learning intersection of 4 halfspaces is SRCSP-hard: Assuming the UGC, we show that a certain family of predicates are heredity approximation resistant. We show also that for these predicates, the problem $\mathrm{CSP}^{1,\alpha}(P)$ is **NP**-hard for some $1 > \alpha > 0$. This leads to the conjecture that these predicates are in fact heredity approximation resistant. Conditioning on the correctness of this conjecture, we show that it is SRCSP-hard to learn intersection of 4-halfspaces. This is done using the strategy described for DNFs.

The proof of the resolution lower bounds relies on the ideas of [16, 6, 5] and [7]. The proof that it is unlikely that the correctness of the SRCSP-assumption can be based on **NP**-hardness uses the idea introduced in [2]: we show that if an **NP**-hard problem (standardly) reduces to $\mathrm{CSP}^{\alpha,\mathrm{rand}}_{m(n)}(P)$, then the problem has a statistical zero knowledge proof. It follows that **NP** $\subset$ **SZKP**, which collapses the polynomial hierarchy.

6. FUTURE WORK

Weaker assumptions? First and foremost, it is very desirable to draw similar conclusions from assumption substantially weaker than SRCSP (see section 4.1). Even more ambitiously, is it possible to reduce some **NP**-hard problem to some of the problems that are deemed hard by the SRCSP assumption? In the full version of this paper, we show that a pedestrian application of this approach is doomed to fail (unless the polynomial hierarchy collapses). This provides, perhaps, a moral justification for an "assumption based" study of average case complexity.

The SRCSP**-assumption.** We believe that our results, together with [13, 1, 12, 8] and [4], make a compelling case that it is of fundamental importance for complexity theory to understand the hardness of random CSP problems. In this context, the SRCSP assumption is an interesting conjecture. There are many ways to try to refute it. On the other hand, current techniques in complexity theory seem too weak to prove it, or even to derive it from standard assumptions. Yet, there are ways to provide more circumstantial evidence in favor of this assumption:

- As discussed above, try to derive it, even partially, from weaker assumptions.
- Analyse the performance of existing algorithms. In section **??** it is shown that no Davis-Putnam algorithm can refute the SRCSP assumption. Also, Barak et al [4] show that the basic **SDP** algorithm [31] cannot refute assumption 4.5, and also 4.3 for certain predicates (those that contain a pairwise uniform distribution). Such results regarding additional classes of algorithms will lend more support to the assumption's correctness.
- Lower bound the refutation complexity of random CSPs in systems stronger than resolution.

For a further discussion, see [4]. Interest in the SRCSP assumption calls for a better understanding of heredity approximation resistance. For recent work in this direction, see [18, 19].

More applications. We believe that the method presented here and the SRCSP-assumption can yield additional results in learning and approximation. Here are several basic questions in learning theory that we are unable to resolve even under the SRCSP-assumption.

1. No algorithm learns the class of *decision trees*. Is it SRCSP-hard to learn decision trees?
2. What is the real approximation ratio of learning halfspaces under SRCSP? Likewise for learning large margin halfspaces and parity.
3. Is it SRCSP-hard to learn intersections of a constantly many halfspaces? Maybe even 2 halfspaces? in the full version of this paper, we suggest a route to provide a positive answer for 4 halfspaces.

Besides application to learning and approximation, it would be fascinating to see applications of the SRCSP-assumption in other fields of complexity. It will be a poetic justice if we could apply it to cryptography. We refer the reader to [4] for a discussion. Finding implications in fields beyond cryptography, learning and approximation would be even more exciting.

Acknowledgements:.

Amit Daniely is a recipient of the Google Europe Fellowship in Learning Theory, and this research is supported in part by this Google Fellowship. Nati Linial is supported by grants from ISF, BSF and I-Core. Shai Shalev-Shwartz is supported by the Israeli Science Foundation grant number 590-10. We thank Sangxia Huang for his kind help and for valuable discussions about his paper [19]. We thank Guy Kindler for valuable discussions.

7. REFERENCES

[1] M. Alekhnovich. More on average case vs approximation complexity. In *Foundations of Computer Science, 2003. Proceedings. 44th Annual IEEE Symposium on*, pages 298–307. IEEE, 2003.

[2] B. Applebaum, B. Barak, and D. Xiao. On basing lower-bounds for learning on worst-case assumptions. In *Foundations of Computer Science, 2008. FOCS'08. IEEE 49th Annual IEEE Symposium on*, pages 211–220. IEEE, 2008.

[3] S. Arora, L. Babai, J. Stern, and Z. Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. In *Foundations of Computer Science, 1993. Proceedings., 34th Annual Symposium on*, pages 724–733. IEEE, 1993.

[4] B. Barak, G. Kindler, and D. Steurer. On the optimality of semidefinite relaxations for average-case and generalized constraint satisfaction. In *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, pages 197–214. ACM, 2013.

[5] P. Beame, R. Karp, T. Pitassi, and M. Saks. On the complexity of unsatisfiability proofs for random k-cnf formulas. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 561–571. ACM, 1998.

[6] P. Beame and T. Pitassi. Simplified and improved resolution lower bounds. In *Foundations of Computer*

Science, 1996. Proceedings., 37th Annual Symposium on, pages 274–282. IEEE, 1996.

[7] E. Ben-Sasson and A. Wigderson. Short proofs are narrowâĂŤresolution made simple. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 517–526. ACM, 1999.

[8] Q. Berthet and P. Rigollet. Computational lower bounds for sparse pca. In *COLT*, 2013.

[9] C. M. Bishop. *Neural networks for pattern recognition*. Oxford university press, 1995.

[10] A. Blum, A. Kalai, and H. Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *Journal of the ACM (JACM)*, 50(4):506–519, 2003.

[11] A. Coja-Oghlan, C. Cooper, and A. Frieze. An efficient sparse regularity concept. *SIAM Journal on Discrete Mathematics*, 23(4):2000–2034, 2010.

[12] A. Daniely, N. Linial, and S. Shalev-Shwartz. More data speeds up training time in learning halfspaces over sparse vectors. In *NIPS*, 2013.

[13] U. Feige. Relations between average case complexity and approximation complexity. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 534–543. ACM, 2002.

[14] U. Feige and E. Ofek. Easily refutable subformulas of large random 3cnf formulas. In *Automata, languages and programming*, pages 519–530. Springer, 2004.

[15] V. Feldman, P. Gopalan, S. Khot, and A. Ponnuswami. New results for learning noisy parities and halfspaces. In *In Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, 2006.

[16] A. Haken. The intractability of resolution. *Theoretical Computer Science*, 39:297–308, 1985.

[17] J. Håstad. Some optimal inapproximability results. *Journal of the ACM (JACM)*, 48(4):798–859, 2001.

[18] S. Huang. Approximation resistance on satisfiable instances for predicates strictly dominating parity. 2012.

[19] S. Huang. Approximation resistance on satisfiable instances for predicates with few accepting inputs. In *STOC*, 2013.

[20] A. T. Kalai, A. R. Klivans, Y. Mansour, and R. A. Servedio. Agnostically learning halfspaces. *SIAM Journal on Computing*, 37(6):1777–1805, 2008.

[21] M. Kearns and L. G. Valiant. Cryptographic limitations on learning Boolean formulae and finite automata. In *STOC*, pages 433–444, May 1989.

[22] M. Kharitonov. Cryptographic hardness of distribution-specific learning. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, pages 372–381. ACM, 1993.

[23] S. Khot. On the power of unique 2-prover 1-round games. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 767–775. ACM, 2002.

[24] S. Khot and R. Saket. On the hardness of learning intersections of two halfspaces. *Journal of Computer and System Sciences*, 77(1):129–141, 2011.

[25] A. R. Klivans and R. O'Donnell. Learning intersections and thresholds of halfspaces. In *Foundations of Computer Science, 2002. Proceedings. The 43rd Annual IEEE Symposium on*, pages 177–186. IEEE, 2002.

[26] A. R. Klivans and R. Servedio. Learning dnf in time $2^{O(n^{1/3})}$. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 258–265. ACM, 2001.

[27] A. R. Klivans and A. A. Sherstov. Cryptographic hardness for learning intersections of halfspaces. In *FOCS*, 2006.

[28] N. Linial, Y. Mansour, and N. Nisan. Constant depth circuits, Fourier transform, and learnability. In *FOCS*, pages 574–579, Oct. 1989.

[29] Y. Mansour. An $o(n \log \log n)$ learning algorithm for dnf under the uniform distribution. *Journal of Computer and System Sciences*, 50(3):543–550, 1995.

[30] L. Pitt and L. Valiant. Computational limitations on learning from examples. *Journal of the Association for Computing Machinery*, 35(4):965–984, Oct. 1988.

[31] P. Raghavendra. Optimal algorithms and inapproximability results for every csp? In *Proceedings of the 40th annual ACM symposium on Theory of computing*, pages 245–254. ACM, 2008.

[32] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–407, 1958. (Reprinted in *Neurocomputing* (MIT Press, 1988).).

[33] R. Schapire. The strength of weak learnability. In *FOCS*, pages 28–33, Oct. 1989.

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

[35] V. N. Vapnik. *Statistical Learning Theory*. Wiley, 1998.