Proceedings of Machine Learning Research vol 98:1–14, 2019 30th International Conference on Algorithmic Learning Theory

Hardness of Improper One-Sided Learning of Conjunctions For All Uniformly Falsifiable CSPs

Alexander Durgin A.DURGIN@WUSTL.EDU
Brendan Juba BJUBA@WUSTL.EDU
Washington University in St. Louis, 1 Brookings Dr., St. Louis, MO 63130 USA

Editors: Aurélien Garivier and Satyen Kale

Abstract

We consider several closely related variants of PAC-learning in which false-positive and false-negative errors are treated differently. In these models we seek to guarantee a given, low rate of false-positive errors and as few false-negative errors as possible given that we meet the false-positive constraint. Bshouty and Burroughs first observed that learning conjunctions in such models would enable PAC-learning of DNF in the usual distribution-free model; in turn, results of Daniely and Shalev-Shwartz establish that learning of DNF would imply algorithms for refuting random k-SAT using far fewer constraints than believed possible. Such algorithms would violate a slight strengthening of Feige's R3SAT assumption, and would violate the RCSP hypothesis of Barak et al. We show here that actually, an algorithm for learning conjunctions in this model would have much more far-reaching consequences: it gives refutation algorithms for all predicates that are falsified by one of the uniform constant strings. To our knowledge, this is the first hardness result of improper learning for such a large class of natural average-case problems with natural distributions.

Keywords: Learning Theory, Abductive Learning, CSP, Improper Learning

1. Introduction

We consider the following, closely related learning models:

(i) In Pitt and Valiant's *heuristic learning* model (Pitt and Valiant, 1988), one seeks a "rule of thumb" that commits (almost) no false-positive errors, and matches the best true-positive rate of members of a given class.

(ii) In Kalai, Kanade, and Mansour's *positive-reliable learning* model (Kalai et al., 2012), one seeks a classifier that, again, commits almost no false-positive errors, and almost matches the false-negative rate of the optimal classifier that makes no false-positive errors.

(iii) In Juba's *learning to abduce* model (Juba, 2016), one seeks a "hypothesis" condition with probability as large as possible such that in the corresponding conditional probability distribution over examples, the label is almost always true, thus "empirically entailed."

The only differences are that Kalai et al. formulate their problem as minimizing false-negatives as opposed to maximizing the positive classification rate, and Juba essentially formulates the problem in terms of precision rather than the raw false-positive rate. Thus, in the realizable setting (where a perfect rule exists) all three models are computationally equivalent. (We establish this in the appendix for completeness.) We also remark briefly that these problems also arise as a special case of the *conditional linear regression* problem (Juba, 2017), which is a variant of robust linear regression for small (minority-fraction) subsets, in which we also ask for a formula describing the subset on which the linear predictor is intended to be used.

© 2019 A. Durgin & B. Juba.

Conjunctions are among the simplest and least-expressive nontrivial representations. They present a natural starting point for studying the extent of learnability in any model. Moreover, conjunctions are of particular significance to the learning to abduce model. In the usual formulation of abduction as a reasoning task, conjunctions are widely considered to be the most natural hypothesis formulation. For example a typical application of abduction is in diagnosing faulty circuits, and the hypothesis explaining a given output is usually a conjunction of faults at various points in the circuit. Indeed, many classical formulations of the abduction task in AI only considered conjunctive hypotheses (e.g., ATMS Reiter and de Kleer (1987)). Thus, conjunctions can be considered to be the central representation class for abduction, and therefore the learnability of that class in Juba's model is particularly significant.

It is therefore somewhat surprising and unfortunate that all evidence to date suggests that conjunctions are not learnable in these models. The first results concerned the "proper learning" variant of the task, in which we seek the representation of a specific conjunction solving the task: Pitt and Valiant showed this problem to be NP-hard (Pitt and Valiant, 1988); indeed, Bshouty and Burroughs (2005) noted that it follows from results of Håstad (1996) that even getting a $n^{1-\gamma}$-approximation to the optimal positive classification rate for this problem (for any constant $\gamma > 0$) is NP-hard.

Bshouty and Burroughs (2005) showed furthermore that even for the "improper" variant in which any representation will do, we cannot obtain any polynomial approximation for the positive classification rate, or else we would obtain an algorithm for PAC-learning DNF in the usual, distribution-free model. This was the central problem in computational learning theory raised by Valiant (1984), and the state-of-the-art algorithm for this problem requires $2^{O(n^{1/3})}$ time and examples (Klivans and Servedio, 2004). Until recently, the only evidence for the hardness of learning DNF was its notoriety. But, Daniely and Shalev-Shwartz (2016), building on techniques pioneered by Daniely et al. (2014), show that learning DNF is hard given that it is hard to distinguish random k-CNFs on $n^{f(k)}$ constraints from satisfiable k-CNFs for any $f(k) = \omega(1)$. This assumption is a slight strengthening of Feige's *R3SAT hypothesis,* Feige (2002) which (only) asserts that no polynomial-time algorithm can distinguish random 3-CNFs of size $O(n)$ from satisfiable 3-CNFs. These connections show, in turn that improper learning of conjunctions is intractable under the same assumptions. Thus, furthermore, learning of any representation that can *express* a conjunction is also intractable in these models. Since almost all natural representations can express conjunctions (except disjunctions and parities, which are learnable (Bshouty and Burroughs, 2005; Kanade and Thaler, 2014; Juba, 2016)), this essentially settles the extent of learnability in these models.

The pioneering work of Daniely et al. (2014) had earlier obtained a number of strong hardness of improper learning results, using assumptions about the hardness of random CSPs for unusual predicates, that were unfortunately subsequently falsified by Allen et al. (2015). While the current, random k-SAT variant used by Daniely and Shalev-Shwartz has yet to be falsified (and indeed seems plausible) and PAC-learning of DNF still seems formidable, it is still desirable to have stronger evidence for the hardness of these problems. But, the hardness of learning has almost always been based on the hardness of specific problems, such as the aforementioned hardness of specific random CSP refutation problems, the hardness of specific cryptographic problems, such as integer factoring (Kearns and Valiant, 1994) or shortest vector problems (Klivans and Sherstov, 2009), or the hardness of planted clique (Berthet and Rigollet, 2013). Applebaum et al. (2008) obtained evidence that such a result for improper learning based on NP-hardness is implausible: most simple kinds of reductions would imply the polynomial-time hierarchy collapses, and more complex reductions would yield a generic reduction to construct weak one-way functions from arbitrary problems in NP that are hard on average. Moreover, no natural problem with a natural distribution has yet been shown to be

complete for the average-case analogues of NP[1] so hardness for the average-case analogues of NP seem beyond our current reach, at least.

Our results. We show that the task of improper learning of conjunctions is hard unless most random k-CSPs can be weakly refuted: specifically, for every predicate that is falsified by some uniform assignment (i.e., all-0s or all-1s for Boolean CSPs), if it is intractable to distinguish random systems on $n^{f(k)}$ constraints from systems with a satisfying assignment for $f(k) = \omega(1)$, then improper learning of conjunctions is also intractable. This includes almost all CSPs considered in the literature which tend to be nontrivial, monotone CSPs, as well as more exotic predicates such as not-all-equal-SAT, random predicates (with probability $3/4$), and notably the $\mathrm{XOR} \oplus \mathrm{MAJ}$ predicate introduced by Applebaum and Lovett (2018) as a candidate hard predicate for PRGs in NC^0 achieving high stretch. We stress that to our knowledge, this is the first example of hardness of improper learning for such a large class of problems.

We note that Barak et al. (2013) had explicitly conjectured that a basic semidefinite program should be optimal for weak refutation of *all* predicates with a constant constraint-to-varible ratio. By contrast, we only require that refutation is hard for *some* predicate on $n^{f(k)}$ constraints for arbitrarily slowly growing $f(k)$. In particular, Kothari et al. (2017) show that the usual sum-of-squares formulation cannot efficiently refute such instances whenever there is a $2f(k) + 1$-wise independent distribution on the satisfying assignments of the predicate. (We say that such predicates *"support $2f(k) + 1$-wise independence."*) Therefore, unless we can improve upon the sum-of-squares algorithm for refutation on $n^{f(k)}$ constraints for *all* CSPs that are falsified by some constant assignment and support $2f(k)+1$-wise independence, there is no polynomial-time algorithm for learning conjunctions in these models. Kothari et al. note that for the same family of predicates, it furthermore follows from work by Lee et al. (2015) that no polynomial-size semidefinite programming extended formulation will succeed for the same family and same number of constraints. Therefore, again, a polynomial time algorithm for learning conjunctions in this model will establish that *no* semidefinite programming formulations are optimal for *any* of these predicates (in stark contrast to the conjecture of Barak et al. (2013)).

Technical overview. Our techniques are inspired by the previous works by Daniely et al. (2014) and Daniely and Shalev-Shwartz (2016). For our problems, however, we find a substantially more direct reduction than either of these works. As a consequence of the simplicity of this argument, it applies to the large class of predicates described above. At a high level, we can describe the reduction for the abduction or heuristic learning variants of the problem as follows: we map the constraints of a given input system to examples labeled 0, and add a set of examples encoding random constraints with label 1. Our task is then to find a conjunction that selects a subset of the random constraints and does not pick up any constraints of the input system. We note that for any candidate assignment to the input system, there is a specific set of literals that maps the assignment to the constant string that would falsify the predicate on every subset. If there is a satisfying assignment to the system, then constraints containing these literals for that assignment cannot appear in the system. But, they do appear in a random system with constant probability. So, a conjunction that checks that all variables either do not appear in the predicate, or appear in the literals that map the assignment to the falsifying string will solve the abduction task. By contrast, if the input system was random, then the labels are independent of the examples, and no conjunction can solve the task. Therefore, finding a good solution to the abduction task distinguishes random constraints from constraints of a satisfiable system.

1. In particular, this is in contrast to specific works where either the distribution is natural and the problem is not (Impagliazzo and Levin, 1990, e.g.,) or where the problem is natural and the distribution is not (Livne, 2010).

2. Preliminaries

We now introduce the computational problems and key definitions we consider in this work.

2.1. Abductive Learning Problem

First, we describe the one-sided learning model in which we are studying the learnability of conjunctions, as described by Juba (2016). (We relate this model to positive-reliable learning in the appendix.)

Definition 1 *For a class $\mathcal{H}$ of Boolean formulas over Boolean attributes $x_1, \ldots, x_n$, the* abduction *task is as follows. We are given as input m independent examples $x^{(1)}, \ldots, x^{(m)}$ from an arbitrary distribution D over $\{0,1\}^n$ (assignments to the n attributes), a* query *formula $c(x)$ over $x_1, \ldots, x_n$, and an* alphabet *$A \subseteq \{x_1, \ldots, x_n\}$, for which there exists $h^* \in \mathcal{H}$ only using attributes in A such that $\Pr[c(x) = 1 | h^*(x) = 1] = 1$ and $\Pr[h^*(x) = 1] \geq \mu$. Then, with probability $1 - \delta$, in time polynomial in $n, 1/\mu, 1/\epsilon$, and $1/\delta$, we find an* explanation *$h \in \mathcal{H}$ only using attributes in A such that*

(i) $\Pr[c(x) = 1 | h(x) = 1] \geq 1 - \epsilon$ *and*

(ii) $\Pr[h(x) = 1] \geq 1/p(1/\mu, n, 1/(1-\epsilon))$ *for some positive polynomial p.*

So, in the case of there being a good ("μ-plausible") explanation for the sample data (an $h \in \mathcal{H}$ with no error on its support), an efficient abductive learner in this model will probably output an approximately correct hypothesis (on its support) with the plausibility (size of the support) depending only polynomially on $n, \frac{1}{\mu}, \frac{1}{\epsilon}, \frac{1}{\delta}$.

2.2. Constraint Satisfaction Problems

The hard problem we reduce to abducing conjunctions is that of distinguishing satisfiable instances of constraint satisfaction problems (CSP) from large enough random instances for at least one predicate out of a large class of predicates. We formally describe our problem as follows:

Definition 2 (Constraint Satisfaction Problems) *We call a boolean function $P : \{0,1\}^k \to \{0,1\}$ a* k-predicate *and say that $C : \{0,1\}^n \to \{0,1\}$ is a P-*constraint *if $\exists \ell_1, ..., \ell_k \in \{x_1, \neg x_1, ..., x_n, \neg x_n\}$ such that $C(x) = P(\ell_1(x), ..., \ell_k(x))$ and each variable seen in the choice of ℓ_i is distinct. We denote the set of all P-constraints by CSP_P.*

For k-predicate P, we say that a random algorithm $\mathcal{A}$ solves the problem of distinguishing between satisfiable and random instances of P-constraints of size $m(n)$ *if on input $S = \{C_1, ..., C_{m(n)}\} \subseteq \mathrm{CSP}_P$ the following holds:*

(i) If there exists $x \in \{0,1\}^n$ such that for each $i \in [m(n)]$ we have $C_i(x) = 1$, then $A(S)$ outputs "satisfiable" *with probability at least $\frac{3}{4}$ with respect to the internal randomness of A and*

(ii) If each $C_i \in S$ is drawn uniformly random from all of the P-constraints and independently from the other P-constraints, then for almost-all choices of S, $A(S)$ outputs "random" *with probability at least $\frac{3}{4}$ with respect to the internal randomness of A.*

If there is no such $\mathcal{A}$ that runs in time $\mathrm{poly}(n)$, then we say that this problem is hard. *Otherwise, we say that this problem is* easy.

In particular, the CSP-based hardness assumption we will reduce from is of the following form:

Assumption 3 (Uniformly Falsified CSP Hardness (UFC)) *There exists some k-predicate P such that either $P(0,0,...,0) = 0$ or $P(1,1,...,1) = 0$ and some function $f(k) = \omega(1)$ for which it is hard to distinguish between satisfiable and random instances of P-constraints of size $n^{f(k)}$.*

Definition 4 *We say that a computational problem is* UFC-hard *if it being efficiently solvable contradicts Assumption 3.*

2.3. Scattering and Explainability

Analogous to distinguishing between random and satisfiable instances of CSP problems, our reduction will make use of the following problem of distinguishing between scattered and explainable samples:

Definition 5 ($(\mathcal{H}, \mu)$-explainable and scattered samples) *Let $S = \{(x_1, y_1), ..., (x_{m(n)}, y_{m(n)})\} \subseteq \{0,1\}^n \times \{0,1\}$ be a labeled sample.*

- *We say that S is* $(\mathcal{H}, \mu)$-explainable *for $\mu > 0$ if there exists $h^* \in \mathcal{H}$ such that*

  (i) $\frac{1}{m(n)} \sum_{i=1}^{m(n)} \mathbb{1}_{\{x|h^*(x)=1\}}(x_i) \geq \mu$

  (ii) For each $i \in [m(n)]$, $h^*(x_i) = 1 \implies y_i = 1$

- *We say that a distribution over $(\{0,1\}^n \times \{0,1\})^m$ is* scattered *if for $S \sim D$ the examples (x_i, y_i) are independent and identically distributed, and the y_i in particular are Bernoulli$(\frac{1}{2})$ random variables that are independent of x_i.*

Definition 6 (Distinguishing Explainable From Scattered Samples) *For hypothesis class $\mathcal{H}$, we say that a random algorithm $\mathcal{A}$ solves* the problem of distinguishing between $(\mathcal{H}, \mu)$-explainable samples and scattered samples of size $m(n)$ *if on input $S = \{(x_1, y_1), ..., (x_{m(n)}, y_{m(n)})\} \subseteq \{0,1\}^n \times \{0,1\}$, $\mathcal{A}$ has the following two behaviors:*

(i) if S is $(\mathcal{H}, \mu)$-explainable, then $A(S)$ outputs "explainable" *with probability at least $\frac{3}{4}$ with respect to the internal randomness of A.*

(ii) If S is drawn from a scattered distribution, then with probability $1 - o_n(1)$ with respect to the choice of S, $A(S)$ will output "scattered" *with probability at least $\frac{3}{4}$ with respect to the internal randomness of A.*

If there is no such $\mathcal{A}$ that runs in time $\text{poly}(n)$*, then we say that this problem is* hard*. Otherwise, we say that this problem is* easy*.*

In particular, our reduction will relate the problem of distinguishing $(\mathcal{H}, \mu)$-explainable from scattered examples to the abductive learnability of $\mathcal{H}$. The basic idea behind this relationship is similar to that of Daniely et al., in that since any efficient abductive learner can use at most a polynomial number of bits to describe an output hypothesis, it will almost certainly output a poorly performing explanation in the scattered case. Meanwhile, in the explainable case, it will do well by assumption. We will give a full proof in the next section.

3. Main Result: One-Sided Improper Learning of Conjunctions Refutes All Uniformly Falsifiable Random CSPs

Our main result concludes that under Assumption 3, abductively learning conjunctions is hard, even improperly:

Theorem 7 (Hardness of Improperly Abductively Learning Conjunctions) *It is UFC-hard to improperly abduce conjunctions.*

The method behind our result is based on that of Daniely et al. Namely, we will reduce the (hard) problem of distinguishing between satisfiable and random CSP instances to the problem of distinguishing between conjunctively explainable and scattered samples. The idea will be to label the set of input constraints negatively, then to randomly (with probability $\frac{1}{2}$) replace constraints with (uniformly) random positively labeled constraints. We will then encode the constraints (as a collection of literals) into a larger set of Boolean attributes. In the case that the input system is satisfiable, due to our mild assumption that P is false on either the all 0 or all 1 input, any satisfying assignment induces a mapping $h^* : \{0,1\}^{2n} \to \{0,1\}$ (computable by a conjunction) from the encoded constraints to $\{0,1\}$ such that for P-constraint C and encoding function $\mathrm{enc} : \mathrm{CSP}_P \to \{0,1\}^{2n}$ we have $h^*(\mathrm{enc}(C)) = 1$ only if $C(x^*) = 0$, i.e., there exists a conjunctive explanation for the encoded input system of constraints. In the random case, on the other hand, there is almost always no such explanation h^*. Hence, any efficient abductive learner for conjunctions will be able to solve our original CSP problem.

Note that for brevity the following analysis will be for the case that $P(0,0,\ldots,0) = 0$, but if we do not have falsification on all 0's and instead have falsification of P on all 1's ($P(1,1,\ldots,1) = 0$), that the analysis is much the same. Furthermore, we note that our argument extends directly to all constant size alphabets Γ given a suitable notion of "literals:" we only require that for any pair of symbols $\sigma, \tau \in \Gamma$, there is a literal function that takes σ to τ (the literals are "1-transitive"). In particular, the constant shift literals used by Georgiou et al. (2009) will suffice.

3.1. The Reduction

First, we will describe our encoding of CSP_P over n variables as elements of $\{0,1\}^{2n}$. Let $\mathrm{enc} : \mathrm{CSP}_P \to \{0,1\}^{2n}$, where for $C = P(\ell_{i_1}, \ldots \ell_{i_k})$, $\mathrm{enc}(C) = z$ such that when we identify $\{0,1\}^{2n}$ with $\{0,1\}^{n \times [2]}$, $z_{(i,1)} = 1 \iff$ either x_i does not appear in any literal ℓ_q of C or x_i appears as a literal itself in C, and similarly $z_{(i,2)} = 1 \iff$ either x_i does not appear in any literal ℓ_q of C or $\neg x_i$ appears as a literal itself in C. Hence, every index of $\mathrm{enc}(C)$ is 1 unless the negation of the associated literal appears in the constraint C.

We now describe the algorithm itself. Let $\mathcal{H}_{\mathrm{con}}$ denote the hypothesis class of conjunctions over n variables and let P denote a k-predicate over n variables. Let $\mathcal{A}$ denote an algorithm that efficiently distinguishes between $(\mathcal{H}_{\mathrm{con}}, \mu)$-explainable (with respect to $\mu = \mu(n,k)$) and scattered samples of size n^d. Then $\mathcal{A}'$, given as Algorithm 1 below, is a polynomial time algorithm for the problem of distinguishing between satisfiable and random instances of P-constraints of size n^d, by reducing to the problem of distinguishing between $(\mathcal{H}_{\mathrm{con}}, \mu)$-explainable and scattered samples of size n^d over $\{0,1\}^{2n}$.

3.2. Correctness of The Reduction

We will now establish the correctness of $\mathcal{A}'$, as shown in Algorithm 1.

Algorithm 1: $\mathcal{A}'$, Reduction of the CSP distinguishing problem to the distinguishing problem for scattered versus explainable samples.

Input : $S = \{C_1, \ldots, C_{n^d}\} \subseteq \mathrm{CSP}_P$

Output: Either "satisfiable" or "random"

Let $S' = \{(C_1, 0), \ldots, (C_{n^d}, 0)\}$ by labeling each input constraint 0

for $i \leftarrow 1$ **to** n^d **do**

  With probability $\frac{1}{2}$ replace, in S', labeled example $(C_i, 0)$ with labeled example $(C, 1)$ for C chosen uniformly at random.

end

Let $E = \{(\mathrm{enc}(C_1), b_1), \ldots, (\mathrm{enc}(C_{n^d}), b_{n^d})\}$, where b_i is the label of the ith example after the randomizing loop.

if $\mathcal{A}(E) =$ "explainable" **then**

  return "satisfiable"

else

  return "random"

end

Lemma 8 *On input a satisfiable system of constraints for some uniformly falsifiable k-constraint P, the algorithm $\mathcal{A}'$ will output "satisfiable" with probability at least $\frac{3}{4}$. And if S is random, then $\mathcal{A}'$ will return "random" with probability at least $\frac{3}{4}$ for a $1 - o_n(1)$-fraction of choices of S.*

We consider each of the two cases of the problem separately. In the case that S is structured (satisfiable) we will show that with probability at least $\frac{3}{4}$ we output "satisfiable". And, in the unstructured case (S is random), we will show that over a $1-o_n(1)$ fraction of all choices of S with probability at least $\frac{3}{4}$ $\mathcal{A}'$ outputs "random". And then afterwards, we will prove our main theorem, relating the problem to abduction.

Claim 9 *If $S = \{C_1, ..., C_{n^d}\}$ is satisfiable, then E is $(\mathcal{H}_{\mathrm{con}}, \mu)$-explainable with respect to any $\mu < \frac{1}{2^{k+1}} - \sqrt{\frac{2^{k+1}}{n^d} \ln 4}$ with probability at least $\frac{3}{4}$.*

Proof Let x^* be a satisfying assignment for $\{C_1, ..., C_{n^d}\}$ and consider the following $h^* \in \mathcal{H}_{\mathrm{con}}$: $h^*(z) = \bigwedge_{i=1}^{n} \alpha_i$ where

$$\alpha_i = \begin{cases} z_{(i,1)}, & \text{if } x^*_i = 0 \\ z_{(i,2)}, & \text{if } x^*_i = 1 \end{cases}$$

In words, $h^*(\mathrm{enc}(C)) = 0$ if and only if some literal belonging to C is satisfied by the assignment x^*. Hence, if we have $h^*(\mathrm{enc}(C)) = 1$ on a μ fraction of E, then we are done. Note that this argument only depends on the value of the predicate on the all 0 string. If instead we had considered predicates that are falsified on the all 1 string, then only the construction of h^* changes in this proof (particularly, each $z_{(i,1)}$ appearing in h^* is swapped for a $z_{(i,2)}$ and vice versa).

Let X_i be the indicator random variable that is 1 if and only if $h^*(\mathrm{enc}(C_i)) = 1$. We make two observations. First, we note that if $h^*(\mathrm{enc}(C_i)) = 1$ then the label of the ith example is also 1. Second, we note that $\mathbb{E}[X_i] = (\frac{1}{2})(\frac{1}{2^k})$. And so, by application of the usual multiplicative Chernoff bounds:

$$\Pr\Big[\sum_{i=1}^{n^d} X_i < \mu n^d\Big] < \exp\Big(-\Big(\frac{1}{2}\Big)(\frac{1}{2^{k+1}} - \mu)^2\Big(\frac{1}{2^{k+1}}\Big)n^d\Big)$$

And so, in order to pick μ so that this probability h^* does a poor job at explaining the labeling is less than $\frac{1}{4}$ we need

$$\exp\Big(-\Big(\frac{1}{2}\Big)(\frac{1}{2^{k+1}}-\mu)^2\Big(\frac{1}{2^{k+1}}\Big)n^d\Big) < \frac{1}{4}$$

and continuing with elementary manipulations we find that happens when

$$\mu < \frac{1}{2^{k+1}} - \sqrt{\frac{2^{k+1}}{n^d}\ln 4}$$

and so we are done. ■

Thus, our reduction maps satisfiable instances of P-constraints to to conjunctively explainable instances over the boolean cube.

Next, observe that in the case of S being drawn uniformly at random, that for almost all choices of S, with probability at least $\frac{3}{4}$ with respect to the internal randomness of the algorithm, we will output "random". Indeed, we observe that the distribution induced by the algorithm on examples over $\{0,1\}^{2n} \times \{0,1\}$ is scattered in the random case, and so, by the assumption that $\mathcal{A}$ solves the problem of distinguishing between $(\mathcal{H}_{\text{con}}, \mu)$-explainable and scattered samples we have this guarantee:

Claim 10 *If S is drawn uniformly at random, then for a $1-o_n(1)$ fraction of the possible S, with probability at least $\frac{3}{4}$ $\mathcal{A}'$ will return "random."*

Proof As $\mathcal{A}'$ outputs "random" whenever $\mathcal{A}$ outputs "scattered" on its input E, and we are given that $\mathcal{A}$ outputs "scattered" with probability $\frac{3}{4}$ for a $1-o_n(1)$-fraction of the possible samples produced by a scattered distribution, it suffices to argue that E is indeed scattered.

Indeed, since $\{C_1, \ldots, C_{n^d}\}$ is by assumption a collection of mutually independent and uniformly random constraints, in particular the set of literals appearing in each C_i is an independent and uniformly random set of k literals on distinct variables. Observe that this remains true even if we sample a new, independent constraint for C_i in the first loop of $\mathcal{A}'$. Hence, $\{\text{enc}(C_1), \ldots, \text{enc}(C_{n^d})\}$ is, by construction, a collection of mutually independent and identically distributed random variables. Moreover, our choice of whether or not to resample C_i and replace its label with 1 is an independent Bernoulli trial, and as the new constraint C_i was (once again) independently and uniformly sampled, we find that indeed the labels y_i are independent and unbiased Bernoulli random variables. Thus we see that E is a scattered sample as claimed. ■

Thus, our reduction maps random CSP instances to scattered samples, and (by Claim 9) mapped satisfiable CSP instances to "explainable" samples. Since by hypothesis $\mathcal{A}$ is able to distinguish scattered from "explainable" samples, our reduction is correct.

3.3. Proof of The Main Theorem

Theorem 7 (Hardness of Improperly Abductively Learning Conjunctions) *It is UFC-hard to improperly abduce conjunctions.*

Proof Toward a contradiction let $\mathcal{L}$ be an efficient abductive learner of $\mathcal{H}_{\text{con}}$. We take $g(n, \mu, \epsilon, \delta) \geq \Omega(\Big((1-\epsilon)\frac{\mu}{n}\Big)^d\Big)$ (for some constant d) to be the lower bound of the plausibility of $\mathcal{L}$'s output explanation guaranteed by definition of $\mathcal{L}$ being an efficient abductive learner when there is a μ-plausible explanation.

Hence, for each fixed choice of $\delta, \epsilon \in (0,1)$ there is some $d > 0$ such that (i) $\mathcal{L}$ reads and writes fewer than $(\frac{n}{\mu})^d$ bits over its execution, including those used to read the input examples and the bits required to describe an output explanation. In particular, the number of examples read and used by the algorithm is at most $(\frac{n}{\mu})^d$, and (ii) $(\frac{n}{\mu})^d \geq \frac{1}{g(n,\mu,\epsilon,\delta)}$.

Let $q = d+1$ and let k be large enough such that $f(k) \geq 3q$. Recall that for our reduction to distinguish between satisfiable and random samples our analysis requires that our subroutine $\mathcal{A}$ be able to efficiently distinguish between $(\mathcal{H}_{\text{con}}, \mu)$-explainable and scattered samples for some $\frac{1}{2^{k+2}} \leq \mu < \frac{1}{2^{k+1}}$. Notice that for sufficiently large n, i.e., $n > \frac{1}{\mu^d}$, we have $n^{d+1} > \left(\frac{n}{\mu}\right)^d$, that is, $n^q \geq (\frac{n}{\mu})^d$.

Now consider the algorithm $\mathcal{L}'$, given as Algorithm 2 below, that on input $S \subseteq \{0,1\}^n \times \{0,1\}$ of size n^{3q}, we claim distinguishes between the case that S is $(\mathcal{H}_{\text{con}}, \mu)$-explainable for such a μ and the case that S is scattered.

Algorithm 2: $\mathcal{L}'$, reduction of distinguishing scattered from explainable samples to abductive learning

Input : $S = \{(x_1, y_1), \ldots, (x_{n^{3q}}, y_{n^{3q}})\}$

Output: Either "explainable" or "random"

1 Run $\mathcal{L}$ with the following parameters: Examples drawn uniformly (with replacement) from S as the input example set, μ as above, $\delta = \frac{1}{8}$, and $\epsilon = \frac{1}{4}$ and let h be the output explanation hypothesis.

2 **if** If $\frac{1}{n^{3q}} \sum_{i=1}^{n^{3q}} \mathbb{1}_{\{(x,y) \in S | h(x)=1\}}(x_i) \geq g(n,\mu,\epsilon,\delta)$ and $\text{Err}_{\{(x,y)\in S|h(x)=1\}}(h) < \frac{1}{4}$ **then**

3 | **return** "explainable"

4 **else**

5 | **return** "random"

6 **end**

Where we define

$$\text{Err}_A(h) = \frac{1}{|A|} \sum_{(x,y)\in A} \mathbb{1}_{\{(x,y)\in A | y=0\}}(x,y)$$

i.e., the fraction of false positives h produces over S.

Suppose that S is $(\mathcal{H}_{\text{con}}, \mu)$-explainable, then by assumption of $\mathcal{L}$ being an abductive learner, for large enough n, with probability at least $1 - \delta > \frac{3}{4}$ it will return an h satisfying the condition in line 2 of the algorithm (since $\mathcal{L}$ must work with respect to the uniform distribution over examples we are using) and $\mathcal{L}'$ will return "explainable".

Now suppose that instead S is drawn from a scattered distribution. We bound the probability that h does well (that it satisfies the conditions of line 2). Let us fix an arbitrary h that may be output by $\mathcal{L}$. Note that when h passes the first condition, on the number of positive classifications, since we have chosen q so that $g(n,\mu,\epsilon,\delta) \geq n^{-q}$, h must be positive on at least n^{2q} examples from S. Since S is scattered, for this fixed h, half of the examples from S it classifies positively are false-positives in expectation. Note that since the labels y_i are independent of the examples x_i in a scattered distribution, conditioning on h passing the first check on the number of positive classifications leaves the distribution on labels uniform. Therefore, by a Chernoff bound, the probability that fewer than $1/4$ of the at least n^{2q} examples h classifies positively are false positive errors is at most $e^{-n^{2q}/8}$. And since there are at most 2^{n^q} possible output hypotheses of $\mathcal{L}$, by applying the union bound over these the probability that the output of $\mathcal{L}$ has error lower than $\frac{1}{4}$ is at most $2^{n^q} e^{-n^{2q}/8} = e^{-n^q(n^q/8 - \ln 2)}$. We conclude that with probability $1 - o_n(1)$ over choices of S, $\mathcal{L}'$ will return

"random" with probability $7/8$. Hence, overall, for sufficiently large n $\mathcal{L}'$ returns "random" with probability greater than $3/4$ as needed.

And thus, if there were an efficient abductive learner $\mathcal{L}$ for $\mathcal{H}_{\text{con}}$ we would contradict the UFC assumption as follows. We first take $\mathcal{L}'$ above as the $\mathcal{A}$ subroutine of $\mathcal{A}'$, which efficiently solves the problem of distinguishing between $(\mathcal{H}_{\text{con}}, \mu)$-explainable and scattered samples when we have such an efficient abductive learner $\mathcal{L}$, for all large enough k, and then use our algorithm $\mathcal{A}'$ to solve the problem of distinguishing between satisfiable and random CSP instances for some hard choice of P, k, and $f(k)$. This contradicts the UFC assumption. ■

4. Directions For Future Work

The main intuition behind our reduction is that the literals of a conjunction can encode any fixed assignment as the preimage of $(0, 0, \ldots, 0)$ or $(1, 1, \ldots, 1)$. Thus, we can distinguish satisfiable from random formulas for a large family of predicates. But, it seems plausible that this observation can be pushed further—although our knowledge that the distribution is uniform in one case made the analysis of the reduction easy, this observation about the connection between conjunctions and assignments does not rely on the uniform distribution. We thus ask whether the class of problems against which we show hardness can be broadened further. In particular, somewhat ambitiously, is it possible to show average case NP-hardness?

In another direction, we have positive results for the agnostic variant of this model: we have a $\tilde{O}(\sqrt{n})$-approximation to the optimal disjunction (Zhang et al., 2017) (based on an earlier algorithm by Peleg (2007)) and a simple $\tilde{O}(s \log \log n)$ approximation to the optimal size-s disjunction (Juba et al., 2018). We would like to know how close to optimal either of these algorithms are. We note that Daniely (2016) has succeeded at using similar techniques (but based on strong assumptions about refuting random XOR) to show $\omega(1)$ lower bounds for the blow-up of agnostically learning halfspaces in the standard improper supervised learning model.[2] Noting the relative simplicity of the reductions for the one-sided error models we consider, we are optimistic that it might be possible to extend Daniely's techniques to analyze the blow-up needed for one-sided learning of disjunctions.

Acknowledgements

This work was partially supported by an AFOSR Young Investigator Award and NSF Award CCF-1718380. This work was partially performed while B. Juba was visiting the Simons Institute for the Theory of Computing.

References

Sarah R. Allen, Ryan O'Donnell, and David Witmer. How to refute a random CSP. In *Proc. 56th FOCS*, pages 689–708, 2015.

Benny Applebaum and Shachar Lovett. Algebraic attacks against random local functions and their countermeasures. *SIAM J. Comput.*, 47(1):52–79, 2018.

Benny Applebaum, Boaz Barak, and David Xiao. On basing lower-bounds for learning on worst-case assumptions. In *Proc. 49th FOCS*, pages 211–220, 2008.

2. Daniely also obtains a $2^{\log^{1-\epsilon} n}$ lower bound for very strong assumptions – for polylogarithmic k, Daniely requires refuting random k-XOR to remain hard with up to $n^{O(k)}$ constraints.

Boaz Barak, Guy Kindler, and David Steurer. On the optimality of semidefinite relaxations for average-case and generalized constraint satisfaction. In *Proc. 4th ITCS*, pages 197–214, 2013.

Q. Berthet and P. Rigollet. Complexity theoretic lower bounds for sparse principal component detection. *J. Mach. Learn. Res. (COLT)*, 30, 1046-1066 2013. Extended version, Conference On Learning Theory. arXiv:1304.0828.

Nader H. Bshouty and Lynn Burroughs. Maximizing agreements with one-sided error with applications to heuristic learning. *Machine Learning*, 59(1-2):99–123, 2005.

Amit Daniely. Complexity theoretic limitations on learning halfspaces. In *Proc. 48th STOC*, pages 105–117, 2016.

Amit Daniely and Shai Shalev-Shwartz. Complexity theoretic limtations on learning DNF's. In *Proc. 29th COLT*, volume 49 of *JMLR Workshops and Conference Proceedings*, pages 1–16. 2016.

Amit Daniely, Nati Linial, and Shai Shalev-Shwartz. From average case complexity to improper learning complexity. In *Proc. 46th STOC*, pages 441–448, 2014.

Uriel Feige. Relations between average case complexity and approximation complexity. In *Proc. 34th STOC*, pages 534–543, 2002.

Konstantinos Georgiou, Avner Magen, and Madhur Tulsiani. Optimal Sherali-Adams gaps from pairwise independence. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, volume 5687 of *LNCS*, pages 125–139. Springer, 2009.

Johan Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. In *Proc. 37th FOCS*, pages 627–636, 1996.

Russell Impagliazzo and Leonid A. Levin. No better ways to generate hard NP instances than picking uniformly at random. In *Proc. 31st FOCS*, pages 812–821, 1990.

Brendan Juba. Learning abductive reasoning using random examples. In *Proc. 30th AAAI*, pages 999–1007, 2016.

Brendan Juba. Conditional sparse linear regression. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 67. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017. Proc. 8th ITCS.

Brendan Juba and Hai S. Le. Precision-recall versus accuracy and the role of large data sets. In *Proc. 33rd AAAI*, 2019.

Brendan Juba, Zongyi Li, and Evan Miller. Learning abduction under partial observability. In *Proc. 32nd AAAI*, pages 1888–1896, 2018.

Adam Tauman Kalai, Varun Kanade, and Yishay Mansour. Reliable agnostic learning. *Journal of Computer and System Sciences*, 78(5):1481–1495, 2012.

Varun Kanade and Justin Thaler. Distribution-independent reliable learning. In *Proc. 27th COLT*, volume 35 of *JMLR Workshops and Conference Proceedings*, pages 3–24, 2014.

Michael Kearns and Leslie Valiant. Cryptographic limitations on learning boolean formulae and finite automata. *Journal of the ACM (JACM)*, 41(1):67–95, 1994.

Adam R. Klivans and Rocco A. Servedio. Learning DNF in time $2^{O(n^{1/3})}$. *JCSS*, 68(2):303–318, 2004.

Adam R. Klivans and Alexander A. Sherstov. Cryptographic hardness for learning intersections of halfspaces. *JCSS*, 75(1):2–12, 2009.

Pravesh K. Kothari, Ryuhei Mori, Ryan O'Donnell, and David Witmer. Sum of squares lower bounds for refuting any CSP. In *Proc. 49th STOC*, pages 132–145, 2017.

James R. Lee, Prasad Raghavendra, and David Steurer. Lower bounds on the size of semidefinite programming relaxations. In *Proc. 47th STOC*, pages 567–576, 2015.

Noam Livne. All natural NP-complete problems have average-case complete versions. *Computational Complexity*, 19(4):477–499, 2010.

David Peleg. Approximation algorithms for the label-covermax and red-blue set cover problems. *J. Discrete Algorithms*, 5:55–64, 2007.

Leonard Pitt and Leslie G. Valiant. Computational limitations on learning from examples. *J. ACM*, 35(4): 965–984, 1988.

Raymond Reiter and Johan de Kleer. Foundations for assumption-based truth maintenance systems: Preliminary report. In *Proc. AAAI-87*, pages 183–188, 1987.

Leslie G. Valiant. A theory of the learnable. *Communications of the ACM*, 18(11):1134–1142, 1984.

Mengxue Zhang, Tushar Mathew, and Brendan Juba. An improved algorithm for learning to perform exception-tolerant abduction. In *Proc. 31st AAAI*, pages 1257–1265, 2017.

Appendix A. Computational Equivalence of Abduction From Examples and Reliable Learning

Kalai et al. (2012) proposed models for *"reliable" learning* in which false positives and false negatives are treated as fundamentally different. In particular, in *positive-reliable learning,* one seeks to guarantee that the false positive rate of the classifier is below a given tolerance ϵ. The false negative rate is then minimized, subject to this hard constraint on the false positive rate. Formally:

Definition 11 (Positive-Reliable Learning (Kalai et al., 2012)) *For a* representation class $\mathcal{C}$ *of* $c : \{0,1\}^n \to \{0,1\}$ *the* distribution-independent positive-reliable learning *task is as follows. One is given access to independent examples from an arbitrary distribution D over $\{0,1\}^n \times \{0,1\}$, and input parameters $\epsilon, \delta \in (0,1]$. With probability at least $1-\delta$, the algorithm should produce a hypothesis $h : \{0,1\}^n \to \{0,1\}$ such that*

(i) $\Pr_{(x,b)\in D}[h(x) = 1 \wedge b = 0] \leq \epsilon$ *and*

(ii) $\Pr_{(x,b)\in D}[h(x) = 0 \wedge b = 1] \leq \min_{c\in\mathcal{C}:\Pr[c(x)=1\wedge b=0]=0} \Pr_{(x,b)\in D}[c(x) = 0 \wedge b = 1] + \epsilon$.

When an algorithm solves this task by producing $h \in \mathcal{C}$, we say that the algorithm is proper.

This definition was intended to capture applications such as spam filtering where (i) no simple classifier can achieve 100% accuracy and (ii) we are more willing to tolerate a few spam e-mails than the loss of legitimate e-mails to the filter.

It turns out that this asymmetric treatment of false positives and false negatives is what we require for abduction. We will focus on the case where the "plausibility" ($\Pr[h(x) = 1]$) is near-optimal.

Theorem 12 *If positive-reliable learning can be solved for a concept class $\mathcal{C}$ in time $T(n, 1/\epsilon, 1/\delta)$, abduction for the class $\mathcal{C}$ can be solved with $1/p(1/\mu, n, 1/(1-\epsilon)) = (1-\epsilon)\mu$ in time $T(n, \frac{2}{\mu\epsilon}, \frac{1}{\delta})$. Conversely, If abduction can be solved in time $T(n, 1/\mu, 1/\epsilon, 1/\delta)$, with $1/p(1/\mu, n, 1/(1-\epsilon)) = (1-\epsilon)\mu$, then positive reliable learning can be achieved in time*

$$O\left(T\left(n, \frac{1}{\mu}, \frac{2}{\epsilon}, \frac{1}{\delta}\right) \log \frac{1}{\epsilon} + \frac{1}{\epsilon^3} \log \frac{1}{\epsilon} \log \frac{1}{\delta}\right).$$

The same results hold for both improper and proper versions of the problem as long as the all-false concept is in $\mathcal{C}$.

Proof Abduction From Positive Reliable learning. Given a query formula ψ, we will extend our examples x to labeled examples (x, b) by putting $b = \psi(x)$, and we will delete the attributes that do not lie in the alphabet.

Suppose there exists $c^* \in \mathcal{C}$ such that $\Pr[c^*(x) = 1 \wedge b = 0] = 0$ and $\Pr[c^*(x) = 1] \geq \mu$ for our given μ. We will call the positive reliable learning algorithm on $\epsilon' = \epsilon\mu/2$ and the desired δ. Then with probability $1 - \delta$, the algorithm returns a hypothesis h (on the alphabet attributes) such that $\Pr[h(x) = 1 \wedge b = 0] \leq \epsilon' = \epsilon\mu/2$ and

$$\begin{aligned}
\Pr[h(x) = 0 \wedge b = 1] &\leq \min_{c:\Pr[c(x)=1\wedge b=0]=0} \Pr[c(x) = 0 \wedge b = 1] + \epsilon' \\
&= \Pr[b = 1] - \max_{c:\Pr[c(x)=1\wedge b=0]=0} \Pr[c(x) = 1] + \epsilon' \\
&\leq \Pr[b = 1] - \mu + \epsilon' \\
&= \Pr[b = 1] - (1 - \epsilon/2)\mu
\end{aligned}$$

We therefore find

$$\begin{aligned}
\Pr[h(x) = 1] &= \Pr[h(x) = 1 \wedge b = 0] + \Pr[h(x) = 1 \wedge b = 1] \\
&= \Pr[h(x) = 1 \wedge b = 0] + \Pr[b = 1] - \Pr[h(x) = 0 \wedge b = 1] \\
&\geq 0 + (1 - \epsilon/2)\mu
\end{aligned}$$

so the second condition is satisfied. Returning to the first condition, since $\mu \leq \frac{1}{1-\epsilon/2} \Pr[h(x) = 1]$,

$$\begin{aligned}
\Pr[h(x) = 1 \wedge b = 0] &\leq \frac{\epsilon}{2} \frac{1}{1 - \epsilon/2} \Pr[h(x) = 1] \\
&\leq \epsilon \Pr[h(x) = 1]
\end{aligned}$$

so our first condition is also satisfied. Thus, h is a solution to the abduction task for μ and ϵ; and, if the positive reliable learning algorithm is proper, then $h \in \mathcal{C}$ as well.

Positive Reliable Learning From Abduction. We will extend our examples to contain a new attribute b (*not* in the alphabet) that contains the labels for positive-reliable learning. In general, our query to the algorithm will be of the form "b."

We next note that we can use binary search to optimize the value of μ for which the abduction algorithm succeeds at finding an explanation at a cost of $O(\log \frac{1}{\mu})$ calls to the algorithm. After each call, by the Chernoff bound, we can test if the returned explanation has probability at least $(1 - \epsilon)\mu$ using $O(\frac{1}{\epsilon^2\mu} \log \frac{1}{\delta})$ time and examples. If the largest μ is at most ϵ, then we will argue that the all-false concept is a solution to

the positive-reliable learning problem, and hence this overhead is at most $O(\log \frac{1}{\epsilon})$ calls to the distribution search algorithm and $O(\frac{1}{\epsilon^3} \log \frac{1}{\delta} \log \frac{1}{\epsilon})$ time for testing the hypotheses.

Consider the hypothesis h produced by the abduction algorithm on input parameters $\epsilon/2$ and δ. We will interpret h as a rule for predicting the labels for positive-reliable learning. With probability $1 - \delta$, its false positive rate is

$$\Pr[h(x) = 1 \wedge b = 0] \leq (\epsilon/2) \Pr[h(x) = 1] \leq \epsilon/2.$$

Now, note that

$$\begin{aligned}\Pr[h(x) = 1] &= \Pr[h(x) = 1 \wedge b = 0] + \Pr[h(x) = 1 \wedge b = 1] \\ &\leq \Pr[h(x) = 1 \wedge b = 1] + \epsilon/2.\end{aligned}$$

Since, supposing first that the optimal $\mu \geq \epsilon$, also

$$\begin{aligned}\Pr[h(x) = 1] &\geq (1 - \epsilon/2) \max_{c:\Pr[c(x)=1 \wedge b=0]=0} \Pr[c(x) = 1] \\ &\geq \max_{c:\Pr[c(x)=1 \wedge b=0]=0} \Pr[c(x) = 1 \wedge b = 1] - \epsilon/2\end{aligned}$$

we have

$$\Pr[h(x) = 1 \wedge b = 1] \geq \max_{c:\Pr[c(x)=1 \wedge b=0]=0} \Pr[c(x) = 1 \wedge b = 1] - \epsilon.$$

Therefore, since for any c

$$\begin{aligned}\Pr[h(x) = 0 \wedge b = 1] + \Pr[h(x) = 1 \wedge b = 1] &= \Pr[b = 1] \\ = \Pr[c(x) = 0 \wedge b = 1] + \Pr[c(x) = 1 \wedge b = 1],\end{aligned}$$

subtracting each side from $\Pr[b = 1]$ gives

$$Pr[h(x) = 0 \wedge b = 1] \leq \min_{c:\Pr[c(x)=1 \wedge b=0]=0} \Pr[c(x) = 0 \wedge b = 1] + \epsilon$$

as needed. We now furthermore note that if instead $\max_{c \in \mathcal{C}:\Pr[c(x)=1 \wedge b=0]} \Pr[c(x) = 1] \leq \epsilon$, then the false-positive rate of the all-false concept is zero, and the optimal false-negative rate is at most ϵ, as needed. Therefore, the all-false concept is a solution. Moreover, if the abduction algorithm was proper and the all-false concept is in $\mathcal{C}$, either way we also have $h \in \mathcal{C}$. ■

Discussion. The proof of Theorem 12 highlights the close connection between the two models. It also clarifies the main *distinction* between them, that the all-false concept may make sense for positive-reliable learning, but it is never a legitimate output for abduction: After all, the all-false concept corresponds to an event of probability zero, and our probability space is discrete. Relatedly, there is a need to "terminate the search early" when the probability of positive examples is very small and the all-zero concept is a good approximator. This is because we require at least $1/\mu$ examples to detect an event of probability μ with constant confidence. Indeed, the sample complexity of the abduction problem is essentially that of learning to high precision and recall, which Juba and Le (2019) have shown differs from that of learning to the same accuracy by precisely this $1/\mu$ factor. The running time of the optimization version of abduction (used in the reduction of positive-reliable learning to abduction) therefore inherently depends on $1/\mu$, whereas positive-reliable learning should not depend on such a μ parameter.