Project Summary, CAREER: The Computational Complexity of Halfspace-Based Learning

Algorithms for learning to classify data have important applications in almost every area of computer science including data mining, computer vision, compiler design, operating system design, speech recognition, computational biology, computational game theory, computational neuroscience, and traditional algorithm design.

A common, simplifying assumption in learning theory is that labeled data can be classified by a *halfspace* in many dimensions. In fact, several of the most important learning algorithms from the past thirty years, such as Perceptron, Winnow, Boosting, and Support Vector Machines, make critical use of a provably efficient algorithm for learning a single halfspace.

Intellectual Merit: Given the ubiquity of halfspace-based learning methods, it is important to understand the computational complexity of the most fundamental halfspace-based learning tasks. This proposal addresses several basic questions about halfspace-based learning that remain unsolved despite three decades of research:

1. Can we develop algorithms for learning a halfspace in the presence of noise?

Labeled data is often noisy, and it is unrealistic to assume that it can be perfectly classified by a halfspace. Recently, in joint work with Adam Kalai, Yishay Mansour, and Rocco Servedio, the PI gave the first provably efficient algorithm for learning halfspaces in malicious noise models with respect to many natural distributions [KKMS05]. The PI believes this algorithm can be applied to more expressive concept classes and distributions and will be useful for creating learning algorithms that require very few labeled examples.

2. Can we efficiently learn intersections of halfspaces?

Although the concept class of intersections of halfspaces has received much attention, the problem of learning the intersection of even two halfspaces remains open. In joint work with Ryan O'Donnell and Rocco Servedio, the PI has obtained the first set of efficient algorithms for many natural and important restrictions of the problem such as learning in the presence of a large margin or with respect to the uniform distribution [KOS04, KS04]. The PI believes the development of a "Perceptron"-type algorithm for learning the intersection of two halfspaces in the presence of a small margin is within reach.

3. What hardness results can we prove for learning halfspace-based concept classes?

The PI and his graduate student Alexander Sherstov have recently obtained the first hardness results for well-studied, halfspace-related learning tasks, such as a reduction from breaking lattice-based cryptosystems to PAC learning the intersection of polynomially many halfspaces [**?**]. The PI believes this reduction can be improved to give hardness results for PAC learning DNF formulas.

Broader Impact: New learning algorithms can lead to powerful tools for practitioners in Biology, Economics, and Statistics, such as the wildly successful Boosting-based algorithms due to Freund and Schapire [FS97, MO97, DCJ$^+$94, **?**]. Therefore, it is important to find new, provably efficient algorithms for learning interesting concept classes or prove hardness results indicating such algorithms are unlikely to exist.

Educational Plan: The PI has developed an educational plan that bridges the interests of the theory group and the AI group at UTCS. As one example, the PI has created a two-semester graduate sequence in computational learning theory that appeals to both theory students and AI students. The course covers the theoretical foundations of learning theory and allows students to implement and test learning algorithms featured in class. Research-oriented projects are required.

Additionally, the PI will continue to teach a course for the Turing Scholars, a group of honors CS undergraduates at UT, regarding the analysis of programs; projects will include implementations of cutting-edge learning algorithms on real-world data sets.

References

[DCJ+94] H. Drucker, C. Cortes, L. D. Jackel, Y. Lecun, and V. Vapnik. Boosting and other ensemble methods. *Neural Computation*, 6(6):1289–1301, 1994.

[FS97] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[KKMS05] A. Kalai, A. Klivans, Y. Mansour, and R. Servedio. Agnostically learning halfspaces. In *Proceedings of the 46th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 11–20, 2005.

[KOS04] A. Klivans, R. O'Donnell, and R. Servedio. Learning intersections and thresholds of halfspaces. *Journal of Computer & System Sciences*, 68(4):808–840, 2004.

[KS04] A. Klivans and R. Servedio. Learning intersections of halfspaces with a margin. In *Proceedings of the 17th Annual Conference on Learning Theory,*, pages 348–362, 2004.

[MO97] R. Maclin and D. Opitz. An empirical evaluation of bagging and boosting. In *AAAI/IAAI*, pages 546–551, 1997.

CAREER: The Computational Complexity of Halfspace-Based Learning

Project Description

1 Introduction

Algorithms for learning to classify data have important applications in almost every area of computer science including data mining, computer vision, compiler design, operating system design, speech recognition, computational biology, computational game theory, computational neuroscience, and traditional algorithm design.

A standard assumption in machine learning is that a relatively simple rule or function determines the classification of data. In computational biology, for example, practitioners often assume there exists a simple function mapping an amino acid sequence to the value TRUE if it belongs to a particular protein family and FALSE otherwise. A natural question follows: given many amino acid sequences and associated TRUE/FALSE labels, is it possible to learn the underlying simple function? Computational learning theory provides a model for this type of classification task where the primary goal is to develop *provably efficient* algorithms for learning functions with respect to a probability distribution on labeled data points.

Several important learning algorithms from the past thirty years, including Perceptron, Winnow, Boosting, and Support Vector Machines, assume that the underlying function is essentially a *halfspace* or *linear threshold function*, i.e., a function of the form $f = \mathsf{sign}(\sum_i \alpha_i x_i - \theta)$ where the α_i and θ are reals. At the heart of all of the above learning algorithms, as well as many methods for learning neural networks, is a provably efficient algorithm for learning a single halfspace.

Given the ubiquity of halfspace-based learning methods, it is important to understand the computational complexity of the most fundamental halfspace-based learning tasks. This proposal takes aim at several basic algorithmic and complexity-theoretic questions about halfspace-based learning that remain unsolved despite three decades of research:

1. Can we develop algorithms for learning a halfspace in the presence of noise?

Labeled data is often noisy, and it is unrealistic to assume that it can be perfectly classified by a halfspace. Unfortunately, we lack algorithms for learning halfspaces in all but the most benign noise models. Recently, in joint work with Adam Kalai, Yishay Mansour, and Rocco Servedio, the PI gave the first provably efficient algorithm for learning halfspaces in the *agnostic* model of learning (see Section 2.3 for a formal definition) with respect to many natural distributions [KKMS05]. Agnostic learning corresponds to learning with respect to malicious noise, and our algorithm is the *first* positive result for agnostically learning a non-trivial concept class. The PI believes he can remove many of the above distributional assumptions and create learning algorithms that require few labeled examples.

2. Can we efficiently learn intersections of halfspaces?

The most natural generalization of a halfspace is the intersection of two or more halfspaces. Note that this is a powerful concept class: any convex set is equivalent to an intersection of halfspaces. Although the concept class of intersections of halfspaces has received much attention, the problem of learning the intersection of even two halfspaces remains open. In joint work with Ryan O'Donnell and Rocco Servedio, the PI has obtained the first set of efficient algorithms for many natural and important restrictions of the problem such as learning in the presence of a large margin or with respect to the uniform distribution [KOS04, KS04b]. Additionally, the PI has developed the fastest known algorithm for learning polynomial-size DNF formulas [KS04a], the most important subclass of intersections of halfspaces. The PI believes that finding an algorithm for learning the intersection of two halfspaces in the presence of a small margin is within reach.

3. What hardness results can we prove for learning halfspace-based concept classes?

We describe why several well-studied instances of halfspace-related learning problems are unlikely to admit efficient solutions. Our results make use of novel techniques from computational complexity theory. Although the complexity of classification problems has been studied previously, several of our approaches

have little (if any) precedent in the literature. As a result, the PI and his graduate student Alexander Sherstov have recently obtained the first hardness results for well-studied, halfspace-related learning tasks. For example, the PI has recently shown that efficiently learning the intersection of $\log^c n$ (for some sufficiently large $c > 0$) halfspaces in n dimensions is harder than breaking cryptosystems based on the worst case complexity of well-known lattice problems [KS06a]. The PI feels that he can use these techniques to prove a hardness result for the DNF learning problem, arguably the most famous open problem in learning theory.

Broader Impact: New learning algorithms can lead to powerful tools for practitioners in Biology, Economics, and Statistics, such as the wildly successful Boosting-based algorithms due to Freund and Schapire [FS97, MO97, DCJ+94, SSL+03]. Therefore, we believe it is important to find new, provably efficient algorithms for learning interesting concept classes or prove hardness results indicating such algorithms are unlikely to exist.

Although this proposal aims to solve notoriously difficult problems in learning theory, we do not feel that we are being overly ambitious. In the last three years, the PI and his colleagues have solved several important open problems in learning theory, such as finding the first efficient algorithm for learning a non-trivial concept class in the agnostic model (halfspaces) [KKMS05], giving the first representation independent hardness results for learning intersections of halfspaces [KS06a], and proving that polynomial-size DNF formulas cannot be properly learned unless $\mathsf{RP} = \mathsf{NP}$ [ABF+04]. These solutions required the development of a set of new techniques in learning theory, and the PI is in a strong position to leverage these techniques to solve future problems in the field.

1.1 Educational Plan

The PI has developed an educational plan for students at all levels via teaching and mentoring. The PI's proposal involves providing students with a comprehensive study of learning theory through new courses and projects that are rooted in mathematics but have many practical learning applications. The PI will encourage collaboration among faculty, graduate students, and undergraduates of differing backgrounds (both AI and Theory) by organizing group meetings and seminars. More specific highlights of this plan include

- Creating a two semester course in machine learning for both AI and Theory students. The first semester will consist of the "fundamentals of learning theory," and the second semester will focus on advanced algorithms and complexity-theoretic techniques for learning. Both semesters will require a project that may involve implementations of learning algorithms.
- Teaching the "Turing Scholars," a UTCS group of honors computer science students, regarding the analysis of programs. The PI will give them a head start in theoretical computer science with both theoretical and applied concepts from machine learning.
- Mentoring undergraduate honors students' research theses.
- Conducting research meetings with both theory students and students from AI who have interests in machine learning.
- Running the theory group's seminar. Last year the PI organized a broad range of talks; almost every week featured a different major researcher in computer science.
- Giving lectures at the department's First Bytes program, a program designed to encourage women to study computer science via a lectures series featuring members of the UTCS department.

Institutional Context

The computer science department at the University of Texas at Austin is an excellent fit for the PI's research interests. The department comprises a world-class group of both theorists and machine-learning practitioners. Anna Gal and David Zuckerman share my strong interest in computational complexity, a field that has provided powerful techniques for developing algorithms in computational learning theory

(see, for example, Section 5). Along with Anna Gal and David Zuckerman, I will explore complexity-theoretic reductions for various applications in learning theory. In addition, algorithms researchers Vijaya Ramachandran and Greg Plaxton keep me informed on the latest tools in combinatorial optimization; these tools have traditionally been used by machine learning practitioners and theorists alike.

I am also fortunate to be surrounded by a strong group of AI researchers whose interests lie primarily in machine learning. Ray Mooney and Peter Stone have developed numerous systems in machine learning for applications ranging from computational biology to robot soccer. I frequently consult with them on the applicability of algorithms from machine learning theory, and I am eager to collaborate on a more applied project.

Finally, there are several faculty at UTCS who work in data mining and use tools from machine learning such as Support Vector Machines. In particular, Inderjit Dhillon and I have had several conversations regarding large margin classifiers and clustering. I am confident we will find new connections between our fields due to our mutual interest in techniques from machine learning.

1.2 Outline of Proposal

In Section 2, we formally define the learning models we will work in for the rest of the proposal. In Sections 3, 4, and 5, we describe the research component of this proposal, outlining work recently accomplished and avenues for future progress. In Section 6, we describe our educational plan.

2 Learning Models

In this section we formally define several learning models we will refer to throughout the proposal. Readers familiar with this material can skip to the next section where we begin to give statements of results and open questions.

2.1 PAC Learning

Here we describe the Probably Approximately Correct (PAC) model of learning due to Valiant [Val84]. A *concept class* $\mathcal{C}$ is any subset of Boolean functions mapping $\{0,1\}^n \to \{0,1\}$ with polynomial (in n) description length (e.g., polynomial-size circuits, DNF formulas with a polynomial number of terms). Fix a *target function* $f \in \mathcal{C}$ and a distribution $\mathcal{D}$ on $\{0,1\}^n$. The learner, who does not know f, receives labeled examples $(x^1, f(x^1)), (x^2, f(x^2)), \ldots, (x^n, f(x^n))$. Here each x^i in $\{0,1\}^n$ is chosen independently at random according to $\mathcal{D}$. An algorithm is said to learn $\mathcal{C}$ if, for any choice of $f \in \mathcal{C}$, on input $\epsilon \in (0,1)$, $\delta \in (0,1)$, the learner receives $\text{poly}(n, \frac{1}{\epsilon})$ labeled examples drawn from $\mathcal{D}$, and outputs, with probability at least $1-\delta$, a hypothesis h such that $\Pr_{x\sim\mathcal{D}}[f(x) \neq h(x)] < \epsilon$. The learner must run in time $\text{poly}(n, \frac{1}{\epsilon}, \frac{1}{\delta})$, and h must be computable in polynomial-time (again in the relevant paramaters). We often refer to *weak learning*, which relaxes the success criterion to $\Pr_{x\sim\mathcal{D}}[f(x) \neq h(x)] \leq \frac{1}{2} - \frac{1}{n^c}$ for a constant c.

The learner need not output an element of $\mathcal{C}$ as its hypothesis. If, however, the learner always outputs a hypothesis h that is an element of $\mathcal{C}$ we say that the algorithm is a *proper* PAC learning algorithm.

The PAC model has been intensely studied in computational learning theory. Still, few interesting concepts admit polynomial-time learning algorithms in the PAC model. The requirement that the algorithm works for *all* distributions $\mathcal{D}$ is often hard to achieve.

Learning interesting concept classes with respect to specialized distributions remains a challenging problem. For example, much work has gone into developing learning algorithms which work with respect to the uniform distribution over $\{0,1\}^n$ [Ver90, LMN93, FJS91, Jac97, BBL98, JKS02] or the uniform distribution over the unit sphere (S^{n-1}) [Bau90, BK97, Vem97, Lon03]. Still, many natural problems related to uniform distribution learning remain open (e.g., it is still open as to whether polynomial-size decision trees are learnable with respect to the uniform distribution [MOS03]).

2.2 Statistical Query Learning

Kearns [Kea98] invented the elegant Statistical Query (SQ) model of learning in which a learner receives various statistics about an unknown function f, as opposed to random examples (as in the PAC model). More precisely, the learner chooses a *statistic* g (a poly-time computable Boolean function) and is allowed to make queries of the following form: What is $E_{x\sim\mathcal{D}}[g(x, f(x))] \pm \tau$ (the expectation is taken over random examples drawn from $\mathcal{D}$)? The value τ is called the *tolerance* parameter. A concept class $\mathcal{C}$ is said to be efficiently learnable to within ϵ in the SQ model if for every $f \in \mathcal{C}$ after making $\text{poly}(n, 1/\epsilon)$ statistical queries (each with tolerance $\tau = 1/n^{O(1)}$), the learner outputs an h such that $Pr_{x\sim\mathcal{D}}[h(x) \neq f(x)] < \epsilon$. The learner must also run in time $\text{poly}(n, 1/\epsilon)$ and the hypothesis must be polynomial-time computable.

It is well known that if a concept class is learnable in the SQ model it is learnable in the PAC model. Kearns proved that if a concept class is learnable in the SQ model, then it can be PAC learned in the presence of *classification noise*, a setting where each example's label is flipped with some fixed probability η.

2.3 Coping with Noise: Agnostic Learning

The above models of learning assume that there is no noise in the data given to the learner. This is an unrealistic assumption, as data is usually corrupted/noisy to some extent. Kearns, Schapire, and Sellie introduced the challenging *Agnostic* model of learning in order to capture the problem of learning with noise. In this model, a learner receives examples from a distribution $\mathcal{D}$ on $X \times \{0, 1\}$. Fix a concept class $\mathcal{C}$; for a distribution $\mathcal{D}$ and concept class $\mathcal{C}$ let opt be the error rate of the optimal hypothesis in $\mathcal{C}$ with respect to $\mathcal{D}$. That is, $\mathsf{opt} = \min_{h\in\mathcal{C}}(Pr_{(x,y)\sim\mathcal{D}}[h(x) \neq y])$. The learner's task is to output a hypothesis h with error rate $\mathsf{opt} + \epsilon$ with respect to $\mathcal{D}$ in time polynomial in n and $1/\epsilon$.

Another way to think about agnostic learning is as follows: fix a concept $f \in \mathcal{C}$ and let an adversary flip an η fraction of f's values. That is, for an η fraction of the points in X, the output of f is negated. Let this corrupted function be f'. The learner is given examples from $\mathcal{D}$ on X labeled according to f' and must output a hypothesis h such that h has error rate less than $\eta + \epsilon$ on future randomly chosen examples drawn from $\mathcal{D}$ (labeled by f').

For example, consider the problem of agnostically learning *disjunctions*: a learner is given access to a set of labeled data points (chosen from some distribution on $X \times Y$) and must output a hypothesis whose error rate is within ϵ of the disjunction with minimal error rate with respect to $\mathcal{D}$ (the learner need not output a disjunction as its hypothesis).

Notice that in the *classification noise* model, each label is corrupted independently with probability η; this does not simulate *adversarial* noise.

2.4 Outline of Research Plan

The research plan comprises the following three goals: 1) construct a set of learning algorithms for learning halfspaces in the presence of noise (and the related goal of learning using with as few examples as possible); 2) develop algorithms for learning intersections of halfspaces, DNF formulas, and other related concept classes; 3) prove hardness results for common halfspace-based learning problems such as learning intersections of many halfspaces in the distribution free model and learning neural networks.

3 Learning Halfspaces in the Presence of Noise

As described in the introduction, efficient algorithms for learning halfspaces are perhaps the most important tools in machine learning. While the guarantees of these algorithms are strong if the data is noise-free, learning a halfspace from noisy examples remains a challenging and important problem. In fact, several open problems in computational learning theory can be reduced to the problem of learning a noisy halfspace [FJS91, BFKV97]. In addition, halfspace learning algorithms are often applied to data sets which are *not* linearly separable (a set of points is said to be linearly separable if there exists a halfspace separating the positive examples from the negative examples). This motivates the following question: what can one

provably say about the performance of halfspace-based learning methods in the presence of noisy data or distributions that are not separable (i.e., distributions on labeled data that cannot be separated by a single halfspace)? Can we develop learning algorithms which tolerate data generated from a "noisy" halfspace and output a meaningful hypothesis?

In joint work with Kalai *et al.* [KKMS05] (invited to appear in a special issue of SICOMP for the best papers of FOCS 2005), the PI has shown how to learn a halfspace with respect to various distributions when the labels are corrupted *adversarially*. Learning with respect to this type of noise is equivalent to learning in the *agnostic* model described in Section 2.3. Recall that in the agnostic learning model, a learner is given access to a set of labeled points chosen from some distribution *without* any guarantee that there exists a simple target function labeling the points. That is, the labeling is arbitrary. Still, for any concept class C there may exist a hypothesis which agrees with the labeling on more than half the points. The learner's goal is to find a hypothesis from a fixed class which maximizes this agreement.

More specifically, in the aforementioned joint work [KKMS05] the PI has solved the following agnostic learning problem: given a set of points chosen from the uniform distribution (over $\{0,1\}^n$ or S^{n-1}, the unit sphere) with *arbitrary labels*, find a hypothesis that approximates the optimal halfspace (the optimal halfspace is the halfspace which, over all halfspaces, classifies the largest fraction of points correctly). Our solution is the first positive result for learning an interesting concept class in the agnostic model. Previous work [BFKV97, Coh97] has focused on learning halfspaces with respect to a significantly easier noise model: the *classification noise* model, where each data point's label is flipped independently with probability $\eta < 1/2$.

The PI's algorithm can be described as follows: given a sufficiently large set S of (possibly corrupted) labeled examples, find the polynomial p of degree $O(1/\epsilon^4)$ that minimizes the quantity $\sum_{(x,y)\in S} |p(x) - y|$. The important thing to notice is that the degree of p is independent of the dimension n, and we prove that such a polynomial p will have error within ϵ of the optimal halfspace if the set S is chosen according to many natural distributions on S^{n-1} or $\{0,1\}^n$. We also show that this minimization can be carried out in time $n^{O(1/\epsilon^4)}$; this yields a polynomial-time algorithm for any $\epsilon = O(1)$.

3.1 A New Fourier-Based Algorithm and Applications

Our proof of correctness makes use of a non-standard Fourier transform over the unit sphere using the Hermite polynomials. Previous results using Fourier analysis in computational learning theory have used the parity functions as a basis. Here we have generalized the traditional parity-based learning algorithm to a problem in numerical approximation regarding how well multivariate orthogonal polynomials can approximate surfaces.

Given that the traditional parity-based approach led to a wealth of interesting learning results [KM93, LMN93, Jac95], the PI is certain that these new relationships to orthogonal polynomials will provide a set of mathematical tools for learning complex concept classes, even in the agnostic learning model. For example, the PI believes that applying a multivariate version of the Hermite polynomials (and considering an appropriate noise-stability analog from the traditional parity basis) results in an algorithm for agnostically learning *arbitrary functions* of halfspaces with respect to uniform distributions on the unit sphere. These concepts are not necessarily convex; such an algorithm would be a significiant improvement over traditional methods which require convexity of the concept class for efficient learning [Vem97].

3.2 Agnostically Learning Monomials

The PI has also given the first subexponential-time algorithm for learning *monomials* with respect to *any distribution*. A monomial is simply an AND over some subset of the n input variables and can be written as a simple halfspace, for example $\mathsf{AND}(x_1, x_3, x_4)$ is equivalent to $\mathsf{sign}(x_1 + x_3 + x_4 - 3)$. Learning noisy monomials is a well-studied problem in computational learning theory. Our algorithm runs in time $2^{\tilde{O}(n^{1/2}\log(1/\epsilon))}$ and uses the fact that the AND function can be computed by a degree $\tilde{O}(\sqrt{n})$ multivariate polynomial with respect to the ℓ_∞ norm.

An important question is whether it is possible to find a subexponential time algorithm for agnostically learning *any* halfspace with respect to an arbitrary distribution. This leads to an interesting geometric

question: what is the lowest degree polynomial (with respect to n and ϵ) on S^{n-1} that can pointwise ϵ-approximate halfspaces with respect to a distribution that obeys some small margin constraint? Techniques from rational approximation seem relevant here [BRS95], and we are convinced progress can be made.

Additionally, the subexponential time algorithm for agnostically learning monomials may be useful for learning small-depth Boolean circuits. The PI can reduce Boolean circuit learning problems to the following combinatorial conjecture: for any small-depth circuit C and distribution D, does there exist an AND or OR on some subset of the variables with non-trivial correlation (measured with respect to D) to C? If so, the above agnostic learning algorithm for monomials could find such an AND or OR. We believe the conjecture is true for any constant depth (the conjecture is true for depth-2 circuits) and will result in the first subexponential time algorithm for learning constant depth circuits with respect to any distribution.

3.3 Learning Using Few Labeled Examples

Given that properly labeled data is often expensive and difficult to obtain, it is important to develop algorithms requiring as few labeled examples as possible. If the unknown function we wish to learn has an extremely small representation (e.g., it is a DNF with only two terms) then, information theoretically, it is possible to learn the function using very few labeled examples. The PI intends to develop learning algorithms whose sample complexity depends only on the size of the unknown concept, as opposed to depending polynomially in n (the number of variables). This is the central problem of attribute efficient learning, studied e.g., in Blum *et al.* [BHL95] and Dhagat *et al.* [DH94]. Currently only low weight linear threshold functions are known to be learnable in this context by using Littlestone's Winnow algorithm [Lit88].

The PI has recently made progress on attribute efficient algorithms for learning decision lists (decision lists are essentially nested "if-then else" expressions and can be computed by a single halfspace) [KS04d]. This well-known problem was posed by A. Blum in 1989 [Blu90] and independently by Valiant ten years later [Val99] and has been studied by many researchers in computational learning theory [Blu96, BL97, DH94, NEY02, Ser00]. A solution to this problem would be a major step forward in our ability to make learning algorithms use few labeled examples.

In joint work with R. Servedio [KS04d], the PI has given an algorithm for learning decision lists of length $k << n$ using $2^{\tilde{O}(k^{1/3})} \log n$ examples and time $n^{\tilde{O}(k^{1/3})}$. The previous best bounds use the Winnow algorithm and require $\Omega(2^k \log n)$ examples and run in time $O(2^k n)$. We establish a new variant of Winnow and show how polynomial threshold functions can capture both the running time and sample complexity of learning concepts.

Is it possible to learn decision lists of length k using $O(k \log n)$ examples in time $n^{O(1)}$? It is also conceivable that polynomial size DNF are learnable using only a polynomial number of samples in time $2^{O(n^{1/2})}$. Since these concepts are small, they have an "effective" margin that is large. As such, techniques like random projection should yield algorithms with excellent sample complexity; the PI is currently investigating this approach. Recently, the PI has discovered attribute efficient learning algorithms for halfspaces with respect to the *uniform* distribution. The proof techniques use combinatorial properties of halfspaces, and we conjecture that the result can be extended to other distributions.

4 New Algorithms for Learning Intersections of Halfspaces and Related Classes

4.1 Learning Intersections of Halfspaces

Recall that a halfspace is a function $f = \mathsf{sign}(\sum_{i=1}^{n} a_i x_i - \theta)$ where the a_i's and θ are reals. The traditional halfspace learning problem is as follows: given a small data set labeled according to an unknown halfspace, produce a hypothesis that will approximate the unknown halfspace's classification of future data points.

This problem, studied since the 50s, has several solutions and, as mentioned in the introduction, is at the heart of the most important machine learning tools from the last three decades (algorithms for

learning neural networks, Boosting algorithms, and Support Vector Machines). Suprisingly, learning the *intersection* of two or more halfspaces remains a challenging open problem [Bau91, Vem97, KP98]. Intersections of halfspaces form a powerful concept class, as any convex body can be written as an intersection of halfspaces.

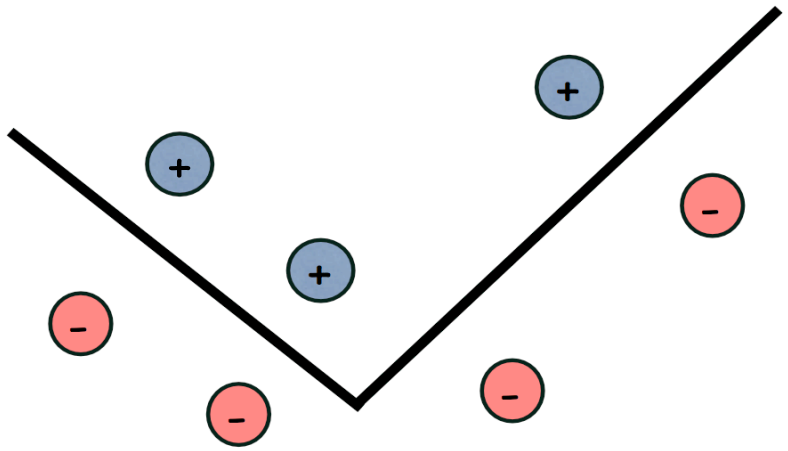

Figure 1: The Intersection of Two Halfspaces

The PI, in joint work with Ryan O'Donnell and Rocco Servedio, has given the *first polynomial-time algorithm* for learning the intersection of a constant number of halfspaces to within any constant error parameter with respect to the uniform distribution on $\{0,1\}^n$ [KOS04] (as explained in Section 2, the uniform distribution indicates that the examples in the small training set and future unlabeled examples are chosen uniformly at random). In this work we also give the first quasipolynomial time algorithm for learning the intersection of a constant number of low-weight halfspaces under any distribution.

Surprisingly, the general problem of PAC learning the intersection of two halfspaces (i.e. with respect to an arbitrary distribution) is still open. Is there a small degree polynomial threshold function which computes the intersection of two halfspaces? Again no upper bound on polynomial threshold function degree less than n is known. A natural and well-studied restriction to this problem is to assume the existence of a *margin* separating positive points from negative points (i.e. each point lies some distance away from the separating hyperplanes). In recent joint work with R. Servedio, the PI proved that if there is a margin of size at least $1/2^{\sqrt{\log n}}$ separating positive points from negative points then there exists a polynomial-time algorithm for learning the intersection of a constant number of halfspaces [KS04b].

To solve the general intersection of halfspaces learning problem we would need an algorithm which works with margins as small as $1/2^n$. It is still open as to whether one can learn intersections of halfspaces in polynomial time with margins as large as $1/n$ [KS04c]. The PI intends to work on this problem and find an algorithm that can work with margins that are polynomially small. Such a result would be analogous to the performance of the Perceptron algorithm [Blo62, Nov62, Ros58, STC00] which learns a *single* halfspace with margin $1/n^{O(1)}$ in polynomial-time and has found numerous applications in machine learning.

The PI's previous results use the Johnson-Lindenstrauss lemma [JL84, DG99] to reduce the problem to learning in a smaller dimension. It seems likely that a more sophisticated embedding will yield improved results. The PI is analyzing various embeddings with respect to metrics other than ℓ_2 (Euclidean space); applying recent techniques from the algorithmic field of embeddings [IM04, KOR01, ARV04, KLMN04] seems like a promising direction.

4.2 Learning DNF Formulas

An important subclass of intersections of halfspaces are DNF formulas. A DNF formula is a logical formula equal to the OR of ANDs, say $(x_1 \wedge x_3 \wedge x_4) \vee (x_5 \wedge x_2)$ for example. Assume that a data set of Boolean strings of length n is labeled according to some unknown polynomial size DNF formula. The above formula will label all $\{0,1\}^n$ strings with a one in position one, three, and four as positive or strings with a one in positions two and five positive. All other strings are labeled negative. A longstanding problem, posed in Valiant's original paper [Val84], is the following: given a small set of examples labeled by an unknown DNF formula, determine the unknown DNF formula (or output a polynomial-time computable hypothesis which approximates the DNF formula). This problem has been the object of intense study in computational learning theory [Ver90, Bsh96, BR95, Kus97, HM91, Jac97, HR02, TT99].

The PI, in joint work with Rocco Servedio, has proved that polynomial size DNF formulas on n variables can be computed by degree $\tilde{O}(n^{1/3})$ *polynomial threshold functions* [KS01a]. This means that there is a real polynomial in n variables of degree $\tilde{O}(n^{1/3})$ that is positive on every input making the unknown DNF true and negative on every input making the unknown DNF formula false. This fact yields a time $2^{\tilde{O}(n^{1/3})}$ algorithm for learning polynomial size DNF formulas; our algorithm is the fastest known DNF learning algorithm (this paper won the *Best Student Paper Award* at STOC 2001). It is possible that a smaller size threshold function can compute DNF formulas over a *different, non-polynomial* basis– the PI is investigating the possibility that a threshold of exponential functions will lead to an alternative construction. Additionally, interesting bounds on the degree of *modular* polynomials (rather than real polynomials) can be found for DNF formulas, and the PI is currently exploring potential learning applications.

4.3 More Complex Boolean Formulas

Very little is known about learning more complicated Boolean formulas. Polynomial size DNFs are equal to polynomial size depth-2 Boolean circuits, and there are no nontrivial PAC learning algorithms for polynomial size circuits of depth 3 or higher. An intriguing open question is to determine the smallest degree polynomial threshold function computing polynomial size Boolean circuits of higher depth (an upper bound on the degree of n turns out to be trivial). An asymptotic bound of $o(n)$ would immediately imply the first nontrivial algorithm for learning higher depth circuits.

Another approach to learning constant depth circuits starts from the observation [ABFR94] that there is a low-degree polynomial that exactly computes any polynomial-size constant depth circuit on *most* of the inputs from $\{0,1\}^n$. The problem reduces to *finding* one of these low-degree polynomial that approximately classifies most points correctly drawn from a suitably large data set. As a low-degree polynomial can be viewed as a halfspace in higher dimensions, we believe that our recent work on agnostic learning can be adopted to this scenario and will yield new results.

5 Lower Bounds: Which Learning Problems are Difficult?

In the previous two sections we outlined approaches for developing new learning algorithms for halfspace-related concept classes. In this section, we consider hardness results and describe several learning problems that are most likely intractable. For example, in Section 4, we discussed the possibility of finding a learning algorithm for the intersection of two halfspaces in n dimensions. In this section we will describe a recent result due to the PI and his student showing that PAC learning the intersection of n halfspaces in n dimensions is harder than solving well-studied *lattice problems* thought to be intractable. As such, it is unlikely we will find an efficient algorithm for learning the intersection of polynomially many halfspaces in n dimensions (unless several lattice-based cryptosystems are insecure).

We give three distinct avenues of research for proving the computational intractability of halfspace-based learning tasks. First, we consider reducing famously hard problems in computational complexity and cryptography to well-known learning problems. These are often referred to as *hardness results*. Secondly, we restrict the model of learning with the hope of proving explicit lower bounds on the resources required to learn. Our main example will involve the elegant Statistical Query model due to Kearns. Finally, we

show that efficient algorithms for learning neural networks, even in very restricted learning models, would resolve famously difficult open problems in complexity theory.

5.1 Lattice-Based Cryptographic Hardness for Learning Problems

In this section we will work within Valiant's PAC model of learning; the goal here is to show that an efficient learning algorithm for a particular concept class implies an efficient algorithm for solving a cryptographic problem thought to be intractable. There is a long history of results in computational learning theory showing that cracking cryptographic primitives based on *factoring* and other number theoretic problems reduces to solving particular learning problems (e.g., Kharitonov's quasipolynomial-time hardness result for learning AC^0 [Kha93]).

In recent work we departed from the usual crypto-learning paradigm based on number theoretic problems and showed that the existence of efficient learning algorithms for intersections of halfspaces implies that *lattice*-based public-key cryptosystems are insecure. We give an example of a lattice-based public-key cryptosystem (due to Oded Regev) [Reg04] below:

Private key: A real number H with $\sqrt{N} \leq H < 2\sqrt{N}$.

Public key: A vector $(A_1, \ldots, A_m, i_0)$, where $i_0 \in \{1, \ldots, m\}$ and each $A_i \in \{0, \ldots, N-1\}$.

Encryption: To encrypt 0, pick a random set $S \subseteq [m]$ and output $\sum_{i \in S} A_i \bmod N$. To encrypt 1, pick a random set $S \subseteq [m]$ and output $\lfloor A_{i_0}/2 \rfloor + \sum_{i \in S} A_i \bmod N$.

Decryption: On receipt of $W \in \{0, \ldots, N-1\}$, decrypt 0 if $\mathrm{frac}(WH/N) < 1/4$, and 1 otherwise. Here $\mathrm{frac}(a) \stackrel{\text{def}}{=} \min\{\lceil a \rceil - a, a - \lfloor a \rfloor\}$ denotes the distance from $a \in \Re$ to the closest integer. By a standard argument, the security and correctness of the cryptosystem are unaffected if we change the decryption function to $\mathrm{frac}(AW) < 1/4$, where A is a representation of H/N to within $\mathrm{poly}(n)$ fractional bits.

Regev [Reg04] proved that an efficient algorithm for distinguishing encryptions of 0 from encryptions of 1 would give an efficient algorithm for solving the unique shortest vector problem, a classical lattice problem thought to be intractable [Ajt96, AD97].

To relate this cryptosystem to learning, we state a general result due to Kearns and Valiant showing that if a concept class $\mathcal{C}$ can compute the decryption function of a public-key cryptosystem then cracking the cryptosystem reduces to PAC learning $\mathcal{C}$:

Theorem 1 *(Cryptography and Learning; cf. Kearns & Valiant [KV94]) Consider a public-key cryptosystem for encrypting individual bits into n-bit strings. Let $\mathcal{C}$ be a concept class that contains all the decryption functions $\{0,1\}^n \to \{0,1\}$ of the cryptosystem. If $\mathcal{C}$ is weakly PAC-learnable in polynomial time then there is a polynomial-time distinguisher between the encryptions of 0 and 1.*

A natural question follows: "what is the simplest concept class that can compute the decryption function for secure public-key cryptosystems?" More specifically, "can polynomial-size DNF formulas or intersections of halfspaces compute the decryption function for public-key cryptosystems?" Until now, the smallest concept class known to compute a decryption function for an interesting public-key cryptosystem were polynomial-size depth-3 threshold circuits (neural networks); these circuits can compute the decryption function of public-key cryptosystems based on the hardness of factoring.

The PI (with A. Sherstov) has shown that intersections of halfspaces *cannot* compute the decryption function of Regev's lattice-based cryptosystem [KS06a]. Nevertheless, The PI *has* shown (again with A. Sherstov) that an efficient algorithm for learning intersections of halfspaces would break two lattice-based cryptosystems due to Regev; we construct a distribution that "helps" an intersection of halfspaces compute the decryption function of Regev's cryptosystem.

This is the first cryptographic hardness result for learning intersections of halfspaces and gives strong evidence that even a subexponential time learning algorithm is unlikely to exist (we note here that intersections of halfspaces are strictly weaker than even depth-2 threshold circuits). All previous cryptographic

hardness results for learning have been based on the hardness of number-theoretic problems such as factoring. Ours is the first hardness result based on the intractability of classically hard lattice problems such as the shortest vector problem.

5.1.1 Hardness for Uniform Distribution Learning of Intersections of Halfspaces?

One drawback of our hardness results is that they make strong use of the PAC model's requirement that the learner must succeed with respect to every distribution. We construct a *non-uniform* distribution that enables intersections of halfspaces to compute the lattice-based cryptosystem's decryption function (as stated above we can prove that the uniform distribution will not suffice). Can we prove hardness results for uniform-distribution learning of intersections of halfspaces? Such results exist for concept classes that can compute strong pseudorandom generators. We believe that finding the appropriate lattice-based pseudorandom generator to solve this problem is within reach.

5.1.2 Lattice-Based Hardness for Learning DNF formulas?

Another central question is whether lattice-based cryptosystems, due to the elegance and simplicity of the associated decryption functions, can be used to prove hardness results for DNF formulas. We strongly believe the answer is yes and plan on finding new cryptosystems based on the *subset-sum* problem [IN96], a problem closely related to lattice problems, to prove a hardness result for learning poly-size DNF formulas. We believe an important first step would be to prove lattice-based hardness results for AC^0 circuits. Since AC^0 can approximately compute linear threshold functions such as majority [ABO84, Ajt87], we feel that this result is attainable in the near future.

5.2 NP-hardness of Properly PAC Learning DNF Formulas

In the last subsection we discussed how efficient PAC learning algorithms can be used to break public-key cryptosystems based on hard lattice problems. In this section we will consider *proper* PAC learning where the learner is restricted to output a hypothesis of the same form as the unknown concept to be learned. We will be able to show results of the form, "if *proper* PAC learning of DNF formulas is efficiently solvable then NP=RP (i.e., there is a randomized reduction from SAT to a problem in P)." This is strong evidence that the associated learning problem is intractable.

As mentioned in Section 4, DNF formulas are the most interesting subclass of intersections of halfspaces, and the DNF learning problem is one of the most important problems in computational learning theory. We do not know if DNF formulas can be learned in polynomial time. In joint work with Alekhnovich et al. [ABF$^+$04], the PI proved that if the output hypothesis of the learner must be a DNF formula then the problem is indeed NP-hard (a similar result holds for intersections of halfspaces). The result settles a long line of research initiated by Pitt and Valiant [PV88] (and studied by several others [HJLT95, KLPV87, NJS98]) on the difficulty of properly learning DNF formulas.

Note that this hardness result does not rule out a non-proper algorithm for learning DNF formulas in polynomial-time, but it does rule out any algorithm that outputs a DNF formula as a final hypothesis. In fact, the PI (along with Alekhnovich et al. [ABF$^+$04]) has proved that even if the output hypothesis is an intersection of halfspaces, the problem is still NP-hard.

The PI plans to extend these results by proving an NP-hardness result for properly *weakly* learning DNF formulas. The problem we plan to address is "is it NP-hard to properly PAC learn DNF formulas if the output hypothesis is only required to have 51% accuracy?" This would be an important result as it would rule out many popular *boosting*-based methods for learning concept classes.

Boosting-based methods use weak learning algorithms (PAC algorithms that output hypotheses with 51% accuracy) as subroutines to output a highly accurate hypothesis. The output of a boosting algorithm is rarely an element of the original concept class, so it is possible boosting based methods fall outside of NP-hardness results for proper learning. Thus, it is important to show that even accomplishing the weak-learning subroutine is NP-hard (with respect to randomized reductions). This would be strong evidence that many boosting-based methods will fail.

The techniques used by the PI in the above proper hardness result relies on strong inapproximability results for computing the chromatic number of a graph (which in turn rely on sophisticated PCP machinery). Given the flurry of recent technical innovations in hardness of approximation results, we expect complexity theory to be a source of useful techniques for this problem. Along these lines, the PI is investigating the possibility of basing hardness results for proper learning on Khot's unique games conjecture [Kho02]. Some proper learning problems are known to be harder than MAX-CUT (as shown by Bartlett *et al.* [BBD02]); Khot *et al.* [KKMO04] have shown a powerful new hardness result for MAX-CUT assuming the unique games conjecture. We are certain more connections can be made here.

5.3 Lower Bounds on the Statistical Query Dimension of Intersections of Halfspaces

The previous two sections have focused on proving *hardness* results for classification problems. In this section we detail our results and open questions for proving *unconditional* lower bounds on the statistical query dimension of well-studied concept classes. We begin with the definition of the statistical query dimension of a concept class:

Definition 2 *The* statistical query dimension *of $\mathcal{C}$ under distribution $\mathcal{D}$, denoted* $\mathbf{SQ\text{-}dim}_{\mathcal{D}}(\mathcal{C})$*, is the largest N for which there are N functions $f_1, \ldots, f_N \in \mathcal{C}$ with $|\mathbf{E}_{x\sim\mathcal{D}}[f_i(x) \cdot f_j(x)]| \leq \frac{1}{N}$ for all $i \neq j$. We denote* $\mathbf{SQ\text{-}dim}(\mathcal{C}) \stackrel{\text{def}}{=} \max_{\mathcal{D}}\{\mathbf{SQ\text{-}dim}_{\mathcal{D}}(\mathcal{C})\}$.

The SQ dimension of a concept class characterizes its weak learnability in the statistical query model: a low SQ dimension implies an efficient weak-learning algorithm, and a high SQ dimension rules out such an algorithm [BFJ$^+$94]. Since we are concerned here with lower bounds, we state only the latter theorem:

Theorem 3 [BFJ$^+$94] *Let $\mathcal{C}$ be a concept class and $\mathcal{D}$ a distribution s.t.* $\mathbf{SQ\text{-}dim}_{\mathcal{D}}(\mathcal{C}) = d \geq 16$*. Then if all queries are made with tolerance at least $1/d^{1/3}$, at least $d^{1/3}/2$ queries are required to learn $\mathcal{C}$ to error $1/2 - 1/d^3$ under $\mathcal{D}$ in the statistical query model.*

It is not difficult to see, for example, that the set of all parity functions on n bits is a concept class with SQ dimension 2^n, as every two distinct parity functions are orthogonal with respect to the uniform distribution. This immediately implies an $n^{\Omega(\log n)}$ lower bound on the SQ-dimension of polynomial-size DNF formulas and intersections of halfspaces, as a polynomial-size DNF formula can compute parity on a subset of $O(\log n)$ input bits. Prior to the PI's recent work with Alexander Sherstov, these were the only known lower bounds on the SQ-dimension of a concept class.

The PI (with A. Sherstov) has shown that intersections of $\sqrt{n}$ halfspaces have SQ-dimension at least $2^{\Omega(\sqrt{n})}$ (as stated above the previous best bound was $n^{\Omega(\log n)}$). This is the first non-trivial SQ-dimension lower bound for an interesting concept class.

The proof stems from considering properties of *bent* functions, Boolean functions whose Fourier spectrum is spread equally among all basis functions. Two immediate questions remain: can we improve the bound to $2^{\Omega(n)}$, and can we prove a new bound for polynomial-size DNF formulas or even constant depth circuits (nothing beyond the parity-inspired bound is known)? We have noticed a new connection to communication complexity: pairwise orthogonal functions correspond precisely to communication matrices with low discrepancy. We are confident that techniques from communication complexity will give new insights into the problem and are currently searching the literature.

5.4 Is Learning Equivalent to Proving Circuit Lower Bounds?

Here we describe a new approach (joint with Lance Fortnow) [FK06] for understanding the difficulty of designing efficient learning algorithms. The point of this section is to show that a PAC (or Exact) learning algorithm for learning depth-2 neural networks implies a *lower bound* against depth-2 neural networks. That is, learning depth-2 neural networks implies a solution to one of the most notoriously difficult problems in computational complexity: a circuit lower bound against depth-2 threshold circuits. More generally, the PI can prove that an efficient learning algorithm for a circuit class C implies a lower

bound against C. Note that depth-2 threshold circuits are a natural generalization of intersections of halfspaces (an intersection of halfspaces is a depth-2 threshold circuit with an AND gate at the root).

There are two interpretations of our result: 1) algorithm designers may wish to avoid working on problems that would resolve major challenges in complexity theory and 2) complexity theorists may want to design learning algorithms in order to prove lower bounds against very restricted models of computation.

Let us give a more specific statement of the result. Assume that a circuit class $\mathcal{C}$ (such as depth-2 threshold circuits) is PAC learnable with respect to the uniform distribution with membership queries in polynomial-time. Then BPEXP, the exponential time analog of BPP, is not contained in $\mathcal{C}$. Currently for depth-2 threshold circuits, we only know that $\mathsf{MA}_{\mathsf{EXP}}$, the exponential time analog of Merlin-Arthur interactive proofs, contains languages with superpolynomial circuit complexity. Improving this separation from $\mathsf{MA}_{\mathsf{EXP}}$ to BPEXP would be a major result and require non-relativizing techniques.

Note that the above result applies to uniform distribution learning and to learners who are allowed membership queries. The hardness results described in previous sections hold for non-uniform distributions and do not apply if the learner is allowed membership queries.

There has been an informal feeling within the learning community that learning algorithms are harder to obtain than circuit lower bounds; a brief glance at the learning literature shows that a lower bound for a circuit class usually precedes the discovery of a learning algorithm for that class. In several cases, the lower bound machinery is used to develop an associated learning algorithm. The PI's result indicates that this is not a coincidence– there is a precise relationship between PAC and Exact learning algorithms and circuit lower bounds.

Our approach is inspired by the results of Kabanets and Impagliazzo [KI03], who showed that derandomizing BPP implies a non-trivial arithmetic circuit lower bound. They further showed some equivalences between derandomizing BPP and proving circuit lower bounds. The PI believes a similar statment can be made for learning. That is, new strong lower bounds should automatically yield (weak) learning algorithms. We believe that further investigation of results from derandomization will be useful here, as derandomization tools have figured prominently in previous work on learning [KS03]. Since lower bounds imply the existence of objects from derandomization such as pseudorandom generators, we believe this is a fruitful avenue for research. Additionally, the PI will work to show finer separations of complexity classes under the assumption that strong learning algorithms exist.

6 Educational Plan

Integral to the PI's proposal is an educational plan that exposes students and faculty to computational learning theory and strengthens the relationship of the theory group to the rest of the department. The plan focuses on 1) creating a two-semester course on computational learning theory, 2) bringing together theorists and AI researchers/students with interests in machine learning, and 3) challenging undergraduates with both theoretical and applied projects in learning. The following sections detail this plan.

A Two-Semester Course in Computational Learning Theory. The PI is in the process of developing a two semester course aimed at both AI and theory students in computational learning theory. The first semester is inspired by a course the PI took at Harvard taught by Leslie Valiant and features classic learning concepts such as PAC learning, VC dimension, Occam's Razor, Boosting, the Perceptron Algorithm, and Statistical Queries. The course will require students to complete a project, as the course material leads naturally to a variety of theoretical and applied projects. The course webpage from Fall 2005 can be found at `http://www.cs.utexas.edu/~klivans/cl.html`. For example, students may test well known algorithms on new data sets (e.g., data from Computational Biology) or attempt to improve theoretical algorithms discussed in the course. Given the large AI community at UTCS, the PI is convinced that this type of course will attract both theory and AI students and result in an interesting exchange of ideas and perspectives.

The second semester of the course will be geared towards more advanced theoretical algorithms in computational learning theory. Because the PI started at UTCS in the Spring of 2005, this course has already been taught once, and the participants were indeed an even mix of theory and AI students. Students taking the course wrote a set of publicly available lecture notes

(`http://www.cs.utexas.edu/~klivans/395t.html`). Students also gave presentations on current research papers related to computational learning theory, and some students have followed up on these presentations by finding similar research topics to pursue. The PI continues to work on improving the two-course sequence.

Teaching and Advising Turing Scholars. The UTCS honors program or "Turing Scholars" program attracts many of the top computer science students in Texas. The PI plans to continue teaching a Turing Scholars course on analyzing programs; this course covers discrete mathematics and its applications to proving the correctness of algorithms. Computational learning theory provides a wealth of simple algorithms that can both be implemented and analyzed in a relatively straightforward way. Thus, students, via homeworks and projects, will be able to apply their discrete math knowledge to machine learning algorithms used in practice. The PI's hope is that this will enhance students' mathemtical maturity while exposing them to state-of-the-art tools in learning.

UTCS undergraduates often write an honors thesis. Currently the PI is advising a Turing Scholar for his thesis project which will involve implementing boosting based algorithms on a variety of data sets. The PI will continue to involve future undergraduates in aspects of the PI's own research regarding the implementation and analysis of machine learning algorithms. This will enable undergraduates to gain research experience at an early age.

Advising Graduate Students. Working with graduate students is central to the university experience. I am fortunate that the top student from my advanced computational learning theory course, Alexander Sherstov, has decided to work with me. Alexander recently won the *MLJ Best Student Paper Award* for our joint work "Improved Lower Bounds for Learning Intersections of Halfspaces" [KS06b] at the 2006 Conference on Learning Theory (COLT), and we have a paper in the upcoming FOCS 2006 conference [KS06a].

In addition, I regularly meet with several theory students. Many of the theory students at UTCS have taken several courses in computational complexity and are interested in the application of complexity-theoretic tools to problems in machine learning. The problem of learning parity with noise, for example, can be restated as a problem about the computational complexity of decoding random linear codes. Anup Rao, a student with interests in complexity and its applications to coding, frequently meets with me to discuss ways for attacking this problem.

UT-Austin's Computer Science department has traditionally been considered a top-ten department with a world-class faculty, and I believe many talented graduate students will continue to come to UTCS in the future.

Algorithms and Computation Theory Seminar. The PI plans to increase the activity of the theory group at UT-Austin. The PI organized a seminar series in Spring 2005 and Fall 2006 that featured many more talks than in previous semesters. In addition, the PI will invite outstanding researchers to visit and give distinguished lectures. Leslie Valiant from Harvard University, for example, visited in the Fall 2005 semester and Chris Umans from Caltech and Jaikumar Radhakrishnan from TTI-Chicago made extended visits. In the Spring 2006 semester, Parikshit Gopalan from Georgia Tech will be the PI's postdoc for 8 months. The PI believes that visits from top researchers increases the visibility of the theory group and promotes interaction between the theory group and the rest of the the department.

First Bytes. First Bytes is a UTCS program for encouraging women to participate in computer science. The program asks faculty to give lectures on an a topic accessible to high school students, and the audience is made up entirely of high school women. The PI believes this is an excellent way to excite young women about computer science and plans to be part of the First Bytes lecture series.

7 Previous Research and Educational Experience

7.1 Research supported by NSF Funding

The PI received an NSF Mathematical Sciences Postdoctoral Research Fellowship from July 2002 to July 2004. Work during this period resulted in five research papers. Each paper appeared in a major international computer science conference:

Learning Intersections and Thresholds of Halfspaces (FOCS 2002, Appears in a Special Issue of Journal of Computer and System Sciences, with R. O'Donnell and R. Servedio). We give the first polynomial time algorithm to learn the intersection of a constant number of halfspaces under the uniform distribution on $\{0,1\}^n$ to within any constant error parameter. We also give the first quasipolynomial time algorithm for learning the intersection of a constant number of halfspaces with polynomial bounded weights under *any* distribution. The only previous result for this problem is due to Baum [Bau91] who gave a polynomial time algorithm for learning the intersection of two halfspaces with respect to any symmetric distribution.

Learning Arithmetic Circuits (COLT 2003, with A. Shpilka). We consider the problem of learning an unknown polynomial that has a succinct encoding as an arithmetic circuit. We show that any arithmetic circuit whose partial derivatives induce a low-dimensional vector space is exactly learnable from membership and equivalence queries. As a consequence we obtain the first polynomial time algorithm for learning polynomial size, depth three multilinear arithmetic circuits. Our learning algorithm can be viewed as solving a generalization of the well known *polynomial interpolation problem* where the unknown polynomial has a succinct representation. We can learn representations of polynomials encoding *exponentially* many monomials. Our techniques combine a careful algebraic analysis of arithmetic circuits' partial derivatives with the "multiplicity automata" techniques due to Beimel et al. [BBB$^+$00].

Toward Attribute Efficient Learning Algorithms (COLT 2004 with R. Servedio). We give the first nontrivial attribute efficient algorithm for learning decision lists and parity functions. Decision lists are essentially nested "if-then else" expressions, for example "If x_1 is true then 1 else if x_2 is true then 0 else if x_3 is true then 1 else 0" is a decision list of length 3. This problem has been posed by A. Blum in 1989 and independently by Valiant ten years later. We can learn decision lists of length $k << n$ using $2^{\tilde{O}(k^{1/3})} \log n$ examples and time $n^{\tilde{O}(k^{1/3})}$. The previous best bounds use the Winnow algorithm and require $\Omega(2^k \log n)$ examples and run in time $O(2^k n)$. addition we can give the first polynomial time algorithm for learning an unknown parity function of length k using only $n^{1-\frac{1}{k}}$ examples.

Learning Intersections of Halfspaces with a Margin (COLT 2004, Invited to Appear in a Special Issue of Journal of Computer and System Sciences, with R. Servedio). We give a new algorithm for learning intersections of halfspaces with a margin, i.e. under the assumption that no example lies too close to any separating hyperplane. Our algorithm combines random projection techniques for dimensionality reduction, polynomial threshold function constructions, and kernel methods. The algorithm is fast and simple. It learns a broader class of functions and achieves an exponential runtime improvement compared with previous work on learning intersections of halfspaces with a margin.

Learnability and Automatizablity (FOCS 2004, Invited to Appear in a Special Issue of Journal of Computer and System Sciences, with M. Alekhnovich, M. Braverman, V. Feldman, T. Pitassi). We give the first hardness results for PAC learning the class of intersections of a polynomial number of halfspaces. We show that it is NP-hard to output the intersection of n^a halfspaces consistent with a data set labeled by an intersection of n^b terms for any constants $a > b > 0$. In addition we show that it is NP-hard to output the intersection of k halfspaces consistent with a data set labeled by the intersection of just two halfspaces for any constant k. This result can be extended to show new hardness results for learning neural networks with a constant number of hidden nodes. Our work improves on work due to Blum and Rivest who showed hardness results for learning the intersetion of two halfspaces by two halfspaces.

7.2 Other Previous Research Accomplishments and Awards

The PI has also published several papers in derandomization and computational complexity theory which have appeared in major international conferences ([KvM02, KS03, KS01b, Kli01, FK05]). Work the PI has done with Dieter van Melkebeek [KvM02] has been used in several recent papers on cryptography and pseudorandomness (e.g., [BOV03, IKW02, SU05]).

A complexity paper of particular relevance to this proposal is "Boosting and Hard-Core Sets" with R. Servedio [KS03] which makes a new connection between Boosting, a popular tool in machine learning [FS97, FS96], and hard-core set construction, a fundamental primitive in complexity and derandomization. Due to space limitations we omit details of the above papers.

The PI won the *Best Student Paper Award* at STOC 2001 for the paper "Learning DNF in Time $2^{\tilde{O}(n^{1/3})}$," written jointly with Rocco Servedio. This paper also received the *Charles and Jennifer Johnson Prize* given annually to the best paper written by an MIT Mathematics graduate student. Several of the PI's papers have been invited to special issues of the *Journal of Computer and System Sciences* [KS04a, KOS04, KS04b, ABF+04], *SICOMP Journal on Computing* [KKMS05], and *Machine Learning* [KS03, KS06b].

7.3 Service

The PI is currently an editor for the new *Theory of Computing* journal run by Laci Babai (see `http://www.theoryofcomputing.org` for more details) as well as the *Machine Learning Journal* (see `http://pages.stern.nyu.edu/~fprovost/MLJ/` for more details) and has served on the program committees for both STOC (Symposium on Theory of Computing) and ICML (International Conference on Machine Learning). The PI will serve on the 2007 Conference on Learning Theory (COLT) program committee, the 2007 Conference on Computational Complexity (CCC) program committee, and the 2007 Algorithmic Learning Theory (ALT) program committee. The PI has also presented papers at several major conferences (STOC, FOCS, COLT, CCC, RANDOM) and has given talks at a variety of major research labs and universities.

7.4 Previous Teaching Experience

I find teaching to be challenging and rewarding work; it is what distinguishes a university from being merely a research institution. I believe that the communication between a student and professor enhances the knowledge and understanding of both parties. Below I describe my previous teaching experience as a teaching assistant and as a professor.

During the summer of 1995,1996, and 1997 I helped Professor Steven Rudich teach his *Andrews' Leap* summer program for talented high school students. The Andrews' Leap program exposes high school students to a world of graduate level computer science and mathematics. Professor Rudich's enthusiasm for teaching has influenced me personally, as I was one of the first "Andrews' Leapers." His attitude has shaped my own desire to contribute to an academic discourse at all levels.

At MIT, I was a teaching assistant for Michael Sipser's Theory of Computation course for four consecutive years. Professor Sipser provided invaluable teaching experience by often allowing me to guest lecture. His lectures require enormous preparation, and I learned how to better plan and organize 75 minutes of instruction.

During the Spring 2005 semester (my first semester as a professor at UT-Austin), I taught an advanced course in computational learning theory. My course was attended by both theoretical computer science students and AI students with interests in machine learning, and I received an overall rating of 4.6/5.0 on my teaching evaluations. In the Fall 2005 semester, I taught a different graduate course in computational learning theory and received an overall rating of 4.3/5.0 on my teaching evaluations. These courses provide a comprehensive, one-year treatment of computational learning theory (see Section 6).

References

[ABF+04] M. Alekhnovich, M. Braverman, V. Feldman, A. R. Klivans, and T. Pitassi. Learnability and automatizability. In *Proceedings of the 45th Symposium on Foundations of Computer Science*, 2004.

[ABFR94] J. Aspnes, R. Beigel, M. Furst, and S. Rudich. The expressive power of voting polynomials. *Combinatorica*, 14(2):1–14, 1994.

[ABO84] M. Ajtai and M. Ben-Or. A theorem on probabilistic constant depth computations. In *Proceedings of the 16th Symposium on Theory of Computing*, 1984.

[AD97] M. Ajtai and C. Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *Proceedings of the 29th Symposium on Theory of Computing*, 1997.

[Ajt87] M. Ajtai. Approximate counting with uniform constant depth circuits. Technical Report RJ 5896, IBM Almaden Research Center, 1987.

[Ajt96] M. Ajtai. Generating hard instances of lattice problems. In *Proceedings of the 28th Symposium on Theory of Computing*, 1996.

[ARV04] S. Arora, S. Rao, and U. Vazirani. Expander flows, geometric embeddings and graph partitioning. In *Proceedings of the 36th Symposium on Theory of Computing*, 2004.

[Bau90] E. Baum. The Perceptron algorithm is fast for nonmalicious distributions. *Neural Computation*, 2:248–260, 1990.

[Bau91] E. Baum. A polynomial time algorithm that learns two hidden unit nets. *Neural Computation*, 2:510–522, 1991.

[BBB+00] A. Beimel, F. Bergadano, N. H. Bshouty, E. Kushilevitz, and S. Varricchio. Learning functions represented as multiplicity automata. *J. ACM*, 47(3):506–530, 2000.

[BBD02] P. Bartlett and S. Ben-David. Hardness results for neural network approximation problems. *TCS: Theoretical Computer Science*, 284, 2002.

[BBL98] A. Blum, C. Burch, and J. Langford. On learning monotone boolean functions. In *Proceedings of the 39th Symposium on Foundations of Computer Science*, 1998.

[BFJ+94] A. Blum, M. Furst, J. Jackson, M. Kearns, Y. Mansour, and S. Rudich. Weakly learning DNF and characterizing statistical query learning using Fourier analysis. In *Proceedings of the 26th Symposium on Theory of Computing*, 1994.

[BFKV97] A. Blum, A. Frieze, R. Kannan, and S. Vempala. A polynomial time algorithm for learning noisy linear threshold functions. *Algorithmica*, 22(1/2):35–52, 1997.

[BHL95] A. Blum, L. Hellerstein, and N. Littlestone. Learning in the presence of finitely or infinitely many irrelevant attributes. *Journal of Computer and System Sciences*, 50:32–40, 1995.

[BK97] A. Blum and R. Kannan. Learning an intersection of a constant number of halfspaces under a uniform distribution. *Journal of Computer and System Sciences*, 54(2):371–380, 1997.

[BL97] A. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1-2):245–271, 1997.

[Blo62] H. Block. The Perceptron: a model for brain functioning. *Reviews of Modern Physics*, 34:123–135, 1962.

[Blu90] A. Blum. Learning Boolean functions in an infinite attribute space. In *Proceedings of the Twenty-Second Symposium on Theory of Computing*, pages 64–72, 1990.

[Blu96] A. Blum. On-line algorithms in machine learning. available at `http://www.cs.cmu.edu/~avrim/Papers/pubs.html`, 1996.

[BOV03] B. Barak, S. Ong, and S. Vadhan. Derandomization in cryptography. In *Proceedings of the 23rd CRYPTO*, 2003.

[BR95] A. Blum and S. Rudich. Fast learning of k-term DNF formulas with queries. *Journal of Computer and System Sciences*, 51(3):367–373, 1995.

[BRS95] R. Beigel, N. Reingold, and D. Spielman. Pp is closed under intersection. *Journal of Computer and System Sciences*, 50(2):191–202, 1995.

[Bsh96] N. Bshouty. A subexponential exact learning algorithm for DNF using equivalence queries. *Information Processing Letters*, 59:37–39, 1996.

[Coh97] E. Cohen. Learning noisy perceptrons by a perceptron in polynomial time. In *Proceedings of the 38th Symposium on Foundations of Computer Science*, pages 514–521, 1997.

[DCJ+94] H. Drucker, C. Cortes, L. D. Jackel, Y. Lecun, and V. Vapnik. Boosting and other ensemble methods. *Neural Computation*, 6(6):1289–1301, 1994.

[DG99] S. Dasgupta and A. Gupta. An elementary proof of the johnson-lindenstrauss lemma. Technical Report TR-99-006, International Computer Science Institute, Berkeley, CA, March 1999.

[DH94] A. Dhagat and L. Hellerstein. PAC learning with irrelevant attributes. In *Proceedings of the 35th Symposium on Foundations of Computer Science*, 1994.

[FJS91] M. Furst, J. Jackson, and S. Smith. Improved learning of AC^0 functions. In *Proceedings of the Fourth Annual Workshop on Computational Learning Theory*, 1991.

[FK05] L. Fortnow and A. Klivans. NP with small advice. In *Proceedings of the 20th Conference on Computational Complexity*, 2005.

[FK06] L. Fortnow and A. Klivans. Efficient learning algorithms yield circuit lower bounds. In *Proceedings of the 19th Conference on Learning Theory*, 2006.

[FS96] Y. Freund and R. Schapire. Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on Machine Learning*, pages 148–156, 1996.

[FS97] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[HJLT95] T. R. Hancock, T. Jiang, M. Li, and J. Tromp. Lower bounds on learning decision lists and trees. In *Proceedings of the 12th Symposium on Theoretical Aspects of Computer Science*, 1995.

[HM91] T. Hancock and Y. Mansour. Learning monotone k-μ DNF formulas on product distributions. In *Proceedings of the Fourth Annual Conference on Computational Learning Theory*, 1991.

[HR02] L. Hellerstein and V. Raghavan. Exact learning of DNF formulas using DNF hypotheses. In *Proceedings of the 34th Symposium on Theory of Computing*, 2002.

[IKW02] R. Impagliazzo, V. Kabanets, and A. Wigderson. In search of an easy witness: Exponential time vs. probabilistic polynomial time. *JCSS: Journal of Computer and System Sciences*, 65, 2002.

[IM04] P. Indyk and J. Matousek. Discrete metric spaces. In Jacob E. Goodman and Joseph O'Rourke, editors, *Handbook of Discrete and Computational Geometry, CRC Press, 1997, 2004*, volume 2. 2004.

[IN96] R. Impagliazzo and M. Naor. Efficient cryptographic schemes provably as secure as subset sum. *Journal of Cryptology*, 9, 1996.

[Jac95] J. Jackson. *The Harmonic sieve: a novel application of Fourier analysis to machine learning theory and practice*. PhD thesis, Carnegie Mellon University, August 1995.

[Jac97] J. Jackson. An efficient membership-query algorithm for learning DNF with respect to the uniform distribution. *Journal of Computer and System Sciences*, 55:414–440, 1997.

[JKS02] J. Jackson, A. Klivans, and R. Servedio. Learnability beyond AC^0. In *Proceedings of the 34th ACM Symposium on Theory of Computing*, 2002.

[JL84] W. Johnson and J. Lindenstrauss. Extensions of lipschitz maps into a hilbert space. *Contemp. Math.*, 26:189–206, 1984.

[Kea98] M. Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM*, 45(6):983–1006, 1998.

[Kha93] M. Kharitonov. Cryptographic hardness of distribution-specific learning. In *Proceedings of the Twenty-Fifth Annual Symposium on Theory of Computing*, pages 372–381, 1993.

[Kho02] S. Khot. On the power of unique 2-prover 1-round games. In *Proceedings of the 34th Symposium on Theory of Computing*, 2002.

[KI03] V. Kabanets and R. Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. In *Proceedings of the 35th Symposium on Theory of Computing*, 2003.

[KKMO04] S. Khot, G. Kindler, E. Mossel, and R. O'Donnell. Optimal inapproximability results for max-cut and other 2-variable CSPs? In *Proceedings of the 45th Symposium on Foundations of Computer Science*, 2004.

[KKMS05] A. Kalai, A. Klivans, Y. Mansour, and R. Servedio. Agnostically learning halfspaces. In *Proceedings of the 46th Foundations of Computer Science*, 2005.

[Kli01] A. Klivans. On the derandomization of constant depth circuits. In *Proceedings of the Fifth International Workshop on Randomization and Approximation Techniques in Computer Science*, 2001.

[KLMN04] R. Krauthgamer, J. Lee, M. Mendel, and A. Naor. Measured descent: A new embedding method for finite metrics. In *Proceedings of the 45th Symposium on Foundations of Computer Science*, 2004.

[KLPV87] M. Kearns, M. Li, L. Pitt, and L. Valiant. On the learnability of Boolean formulae. In *Proceedings of the 19th Annual Symposium on Theory of Computing*, pages 285–295, 1987.

[KM93] E. Kushilevitz and Y. Mansour. Learning decision trees using the Fourier spectrum. *SIAM J. on Computing*, 22(6):1331–1348, 1993.

[KOR01] E. Kushilevitz, R. Ostrovsky, and Y. Rabani. Efficient search for approximate nearest neighbor in high dimensional spaces. *SIAM Journal on Computing*, 30(2):457–474, April 2001.

[KOS04] A. Klivans, R. O'Donnell, and R. Servedio. Learning intersections and thresholds of halfspaces. *Journal of Computer & System Sciences*, 68(4):808–840, 2004.

[KP98] S. Kwek and L. Pitt. PAC learning intersections of halfspaces with membership queries. *Algorithmica*, 22(1/2):53–75, 1998.

[KS01a] A. Klivans and R. Servedio. Learning DNF in time $2^{\tilde{O}(n^{1/3})}$. In *Proceedings of the 33rd Annual Symposium on Theory of Computing*, 2001.

[KS01b] A. Klivans and D. Spielman. Randomness efficient identity testing of multivariate polynomials. In *Proceedings of the 33rd Symposium on Theory of Computing*, 2001.

[KS03] A. Klivans and R. Servedio. Boosting and hard-core sets. *Machine Learning*, 53(3):217–238, 2003.

[KS04a] A. Klivans and R. Servedio. Learning DNF in time $2^{\tilde{O}(n^{1/3})}$. *Journal of Computer & System Sciences*, 68(2):303–318, 2004.

[KS04b] A. Klivans and R. Servedio. Learning intersections of halfspaces with a margin. In *Proceedings of the 17th Annual Conference on Learning Theory*, 2004.

[KS04c] A. Klivans and R. Servedio. Perceptron-like performance for intersections of halfspaces (short). In *Proceedings of the 17th Conference on Learning Theory*, 2004.

[KS04d] A. Klivans and R. Servedio. Toward attribute efficient learning of decision lists and parities. In *Proceedings of the 17th Conference on Learning Theory*, 2004.

[KS06a] A. Klivans and A. A. Sherstov. Cryptographic hardness results for learning intersections of halfspaces. In *Proceedings of the 47th Symposium on Foundations of Computer Science*, 2006. To appear.

[KS06b] A. Klivans and A. A. Sherstov. Improved lower bounds for learning intersections of halfspaces. In *Proceedings of the 19th Conference on Learning Theory*, 2006.

[Kus97] E. Kushilevitz. A simple algorithm for learning $o(\log n)$-term DNF. *Information Processing Letters*, 61(6):289–292, 1997.

[KV94] M. Kearns and L. Valiant. Cryptographic limitations on learning Boolean formulae and finite automata. *Journal of the ACM*, 41(1):67–95, 1994.

[KvM02] A. Klivans and D. van Melkebeek. Graph nonisomorphism has subexponential size proofs unless the polynomial-time hierarchy collapses. *SICOMP: SIAM Journal on Computing*, 31, 2002.

[Lit88] N. Littlestone. Learning quickly when irrelevant attributes abound: a new linear-threshold algorithm. *Machine Learning*, 2:285–318, 1988.

[LMN93] N. Linial, Y. Mansour, and N. Nisan. Constant depth circuits, Fourier transform and learnability. *Journal of the ACM*, 40(3):607–620, 1993.

[Lon03] P. Long. An upper bound on the sample complexity of pac learning halfspaces with respect to the uniform distribution. *Information Processing Letters*, 87(5):229–234, 2003.

[MO97] R. Maclin and D. Opitz. An empirical evaluation of bagging and boosting. In *AAAI/IAAI*, pages 546–551, 1997.

[MOS03] E. Mossel, R. O'Donnell, and R. Servedio. Learning juntas. In *Proceedings of the 35th Annual Symposium on Theory of Computing*, 2003.

[NEY02] Z. Nevo and R. El-Yaniv. On online learning of decision lists. *Journal of Machine Learning Research*, 3:271–301, 2002.

[NJS98] R. Nock, P. Jappy, and J. Sallantin. Generalized graph colorability and compressibility of boolean formulae. In *Proceedings of the 9th International Symposium on Algorithms and Computation (ISAAC)*, 1998.

[Nov62] A. Novikoff. On convergence proofs on perceptrons. In *Proceedings of the Symposium on Mathematical Theory of Automata*, volume XII, pages 615–622, 1962.

[PV88] L. Pitt and L. Valiant. Computational limitations on learning from examples. *Journal of the ACM*, 35(4):965–984, 1988.

[Reg04] O. Regev. New lattice-based cryptographic constructions. *JACM: Journal of the ACM*, 51, 2004.

[Ros58] F. Rosenblatt. The Perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–407, 1958.

[Ser00] R. Servedio. Computational sample complexity and attribute-efficient learning. *Journal of Computer and System Sciences*, 60(1):161–178, 2000.

[SSL+03] P. Stone, R. E. Schapire, M. L. Littman, J. A. Csirik, and D. A. McAllester. Decision-theoretic bidding based on learned density models in simultaneous, interacting auctions. *J. Artif. Intell. Res. (JAIR)*, 19:209–242, 2003.

[STC00] J. Shawe-Taylor and N. Cristianini. *An introduction to support vector machines*. Cambridge University Press, 2000.

[SU05] R. Shaltiel and C. Umans. Pseudorandomness for approximate counting and sampling. In *Proceedings of the 20th Annual Conference on Computational Complexity*, 2005.

[TT99] J. Tarui and T. Tsukiji. Learning DNF by approximating inclusion-exclusion formulae. In *Proceedings of the 14th Conference on Computational Complexity*, 1999.

[Val84] L. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[Val99] L. Valiant. Projection learning. *Machine Learning*, 37(2):115–130, 1999.

[Vem97] S. Vempala. A random sampling based algorithm for learning the intersection of halfspaces. In *Proceedings of the 38th Symposium on Foundations of Computer Science*, 1997.

[Ver90] K. Verbeurgt. Learning DNF under the uniform distribution in quasi-polynomial time. In *Proceedings of the Third Annual Workshop on Computational Learning Theory*, 1990.