On the Relationship between Models for Learning in Helpful Environments

Rajesh Parekh[1] and Vasant Honavar[2]

[1] Blue Martini Software
2600 Campus Drive
San Mateo, CA 94403. USA
rparekh@bluemartini.com
[2] Department of Computer Science
Iowa State University
Ames, IA 50011. USA
honavar@cs.iastate.edu
http://www.cs.iastate.edu/~honavar/aigroup.html

Abstract. The PAC and other equivalent learning models are widely accepted models for polynomial learnability of concept classes. However, negative results abound in the PAC learning framework (concept classes such as *deterministic finite state automata* (DFA) are not efficiently learnable in the PAC model). The PAC model's requirement of learnability under all conceivable distributions could be considered too stringent a restriction for practical applications. Several models for learning in more *helpful environments* have been proposed in the literature including: *learning from example based queries* [2], *online learning allowing a bounded number of mistakes* [14], *learning with the help of teaching sets* [7], *learning from characteristic sets* [5], and *learning from simple examples* [12,4]. Several concept classes that are not learnable in the standard PAC model have been shown to be learnable in these models. In this paper we identify the relationships between these different learning models. We also address the issue of unnatural collusion between the teacher and the learner that can potentially trivialize the task of learning in helpful environments.

Keywords:
Models of learning, Query learning, Mistake bounded learning, PAC learning, teaching sets, characteristic samples, DFA learning.

1 Introduction

Valiant's PAC learning model [20] provided a framework for extensive research on the computational complexity of various learning tasks. A concept class is said to be polynomially learnable if there exists an algorithm that can find a hypothesis approximating any concept in the class, when given a polynomial number of labeled examples and polynomially bounded computational resources. Further, the algorithm is expected to run in time that is polynomial in the parameters measuring the complexity of the target concept, size of the input

A.L. Oliveira (Ed.): ICGI 2000, LNAI 1891, pp. 207–220, 2000.

to the algorithm, and the accuracy of the resulting approximation. The specific assumptions and criteria used to define polynomial learnability have led to several variations of the basic PAC model. A unifying framework for proving the equivalence of these different models was presented in [8]. Despite the PAC model's acceptance as a standard model of polynomial learning, several negative results about PAC learning have been proven (for instance, even elementary concept classes such as DFA cannot be efficiently PAC learned [19,11]). Perhaps, the main reason for these negative results is the model's requirement that the concept class must be learnable under any arbitrary (but fixed) probability distribution. It is conceivable that most practical learning scenarios do not place such stringent restrictions on the learnability of concept classes. On the contrary, practical learning scenarios feature helpful learning environments (for example, a knowledgeable teacher might guide the learner by answering queries or by carefully selecting training examples that would enable the learner to learn quickly and efficiently).

Several models for learning in helpful environments have been proposed in the literature. These include: *learning from example based queries* [2,7], *online learning allowing a bounded number of mistakes* [14][1], *learning with the help of teaching sets* [7], *learning from characteristic sets* [5], and *learning from simple examples* [12,4]. A variety of concept classes whose learnability in the standard PAC model is unknown are shown to be learnable in the above models (see section 2 for the results on learning DFA).

In this paper, we study the relationships between these different models. Some of these relationships have been identified by earlier research whereas others are new. Fig. 1 gives a schematic representation of the relationships. The rest of this paper is organized as follows: Section 2 provides an overview of the different learning models. Section 3 proves the relationships outlined in Fig. 1. Section 4 addresses the issue of collusion in the models for learning in helpful environments. Section 5 concludes with a summary and some directions for future research.

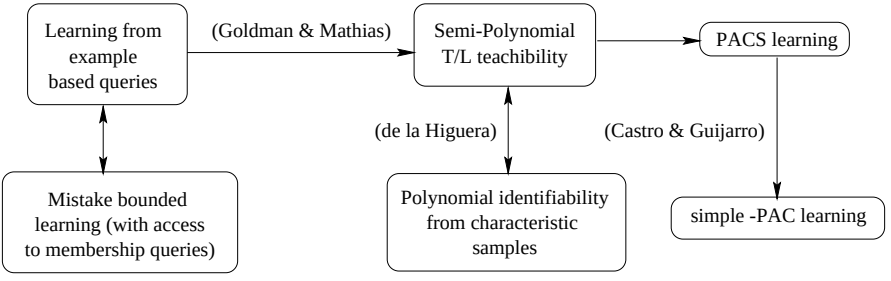

Fig. 1. Relationship between different learning models.

[1] We consider a variant of the *mistake bounded learning* model where the learner has access to a *membership oracle* or a teacher who answers membership queries.

2 Models for Learning in Helpful Environments

2.1 Preliminaries

Let Σ denote a finite alphabet. If $n \geq 1$ denotes the number of attributes then the set Σ^n is referred to as the sample space $\mathcal{X}$. If the learning domain involves examples of varying lengths then the sample space is denoted as $\mathcal{X} = \bigcup_{n \geq 1} \Sigma^n$. A concept class $\mathcal{C}$ is defined as $\mathcal{C} \subseteq 2^{\mathcal{X}}$. An individual concept $c \in \mathcal{C}$ is thus a subset of $\mathcal{X}$. A concept is usually associated with a classification function $c : \mathcal{X} \longrightarrow \{0, 1\}$ such that $c(x) = 1$ if an example x belongs to the concept and $c(x) = 0$ otherwise. The tuple $(x, c(x))$ represents a labeled example of c. If S is a set of labeled examples then $||S||$ denotes the size of S (i.e., the sum of the lengths of the individual examples in S). A representation $\mathcal{R}$ assigns a name to each concept in $\mathcal{C}$ and is defined as a function $\mathcal{R} : \mathcal{C} \longrightarrow \{0, 1\}^*$. Let $r = \mathcal{R}(c)$ be the representation of a concept c. $|r|$ (the length of the string r) denotes the size of the concept c. Let $\mathcal{D}$ be an arbitrary (but fixed) probability distribution defined over $\mathcal{X}$.

A concept class $\mathcal{C}$ is said to be *probably approximately correctly* (PAC) learnable if there exists (a possibly randomized) algorithm $\mathcal{A}$ such that on input of any parameters ϵ and δ, for any concept $c \in \mathcal{C}$ with corresponding representation r, and for any probability distribution $\mathcal{D}$ over $\mathcal{X}$, if $\mathcal{A}$ draws a set S of labeled examples of c, then $\mathcal{A}$ produces an approximation $\hat{c}$ of c such that with probability $\geq 1 - \delta$, $\Pr_{\mathcal{D}}(\{x | x \in \mathcal{X} \text{ and } c(x) \neq \hat{c}(x)\}) \leq \epsilon$. The run time of $\mathcal{A}$ is required to be polynomial in $1/\epsilon$, $1/\delta$, $|r|$, and $||S||$. If the algorithm $\mathcal{A}$ is such that any concept in $\mathcal{C}$ is learned exactly i.e., with probability $\geq 1 - \delta$, $\Pr_{\mathcal{D}}(\{x | x \in \mathcal{X} \text{ and } c(x) \neq \hat{c}(x)\}) = 0$ then $\mathcal{C}$ is said to be *probably exactly* learnable[2].

Kolmogorov complexity is a machine independent notion of *simplicity* of objects. Objects that have regularity in their structure (i.e., objects that can be easily compressed) have low Kolmogorov complexity. For any string $\alpha \in \{0, 1\}^*$, the *prefix Kolmogorov complexity* of α relative to a Turing Machine ϕ is defined as $K_\phi(\alpha) = min\{|\pi| \mid \phi(\pi) = \alpha\}$ where $\pi \in \{0, 1\}^*$ is a program input to the Turing machine. The *Optimality Theorem* for Kolmogorov Complexity guarantees that for any prefix Turing machine ϕ there exists a constant c_ϕ such that for any string α, $K_\psi(\alpha) \leq K_\phi(\alpha) + c_\phi$ where ψ is the Universal Turing Machine. Further, by the *Invariance Theorem* it can be shown that for any two universal Turing machines ψ_1 and ψ_2 there is a constant $\eta \in \mathcal{N}$ (where $\mathcal{N}$ is the set of natural numbers) such that for all strings α, $|K_{\psi_1}(\alpha) - K_{\psi_2}(\alpha)| \leq \eta$. Thus, fixing a single universal Turing machine U we denote $K(\alpha) = K_U(\alpha)$. The Kolmogorov complexity of a string is bounded by its length i.e., $K(\alpha) \leq |\alpha| + K(|\alpha|) + \zeta$ where ζ is a constant independent of α. The conditional Kolmogorov complexity of any string α given β is defined as $K_\phi(\alpha \mid \beta) = min\{|\pi| \mid \phi(\langle \pi, \beta \rangle) = \alpha\}$ where

[2] Note that in this case $\mathcal{A}$ takes in only δ as a parameter and is expected to run in time polynomial in $1/\delta$, $|r|$, and $||S||$.

$\pi \in \{0,1\}^*$ is a program and $\langle .,. \rangle$ is a standard pairing function. Fixing a single universal Turing machine U we denote the conditional Kolmogorov complexity by $K(\alpha|\beta) = K_U(\alpha|\beta)$.

The Solomonoff Levin universal distribution $\mathbf{m}$ is a *universal enumerable probability distribution* in that it multiplicatively dominates all enumerable probability distributions. Formally, $\forall i \in \mathcal{N}^+ \; \exists c > 0 \; \forall x \in \mathcal{N} \; [c\mathbf{m}(x) \geq P_i(x)]$ where P_1, P_2, ... is an enumeration of all enumerable probability distributions and $\mathcal{N}$ is the set of natural numbers. It can be shown that $\mathbf{m}(x) = 2^{-K(x)+O(1)}$. Thus, under $\mathbf{m}$, simple objects (or objects with low Kolmogorov complexity) have a high probability, and complex or random objects have a low probability. Given a string $r \in \Sigma^*$, the universal distribution conditional on the knowledge of r, $\mathbf{m}_r$, is defined as $\mathbf{m}_r(\alpha) = 2^{-K(\alpha|r)+O(1)}$ [4]. Further, $\forall r \in \Sigma^* \; \sum_\alpha \mathbf{m}_r(\alpha) < 1$. The interested reader is referred to [13] for a thorough treatment of Kolmogorov complexity, universal distribution, and related topics.

2.2 Learning from Example Based Queries

A variety of concept classes are known to be learnable in deterministic polynomial time when the learner is allowed access to a teacher (or an oracle) that answers *example based queries* [2]. Example based queries include *equivalence*, *membership*, *subset*, *superset*, *disjointedness*, *exhaustive*, *justifying assignment*, and *partial equivalence* queries. A membership query is of the form "does $x \in c$?" where $x \in \mathcal{X}$ is an example and $c \in \mathcal{C}$ is the target concept. The teacher's response is *yes* or *no* depending on whether $c(x) = 1$ or not. For all other types of queries the input is the learner's hypothesis $\hat{c}$ and the teacher's response is either a *yes* or a counterexample $x \in \mathcal{X}$. Thus, an equivalence query is of the form "$\forall x \in \mathcal{X}$ is $c(x) = \hat{c}(x)$?". The teacher's response is either *yes* or an example x such that $c(x) \neq \hat{c}(x)$.

Definition 1. *(Due to Goldman and Mathias [6])*
An example based query is any query of the form

$$\forall (x_1, x_2, \ldots, x_k) \in \mathcal{X}^k \text{ does } \phi_r(x_1, x_2, \ldots, x_k) = 1?$$

where r is the target concept and k is a constant.

ϕ may use the instances $(x_1, \ldots, x_k)$ to compute additional instances on which to perform membership queries. The teacher's response to example based queries is either *yes* or a counter example consisting of $(x_1, x_2, \ldots, x_k) \in \mathcal{X}^k$ (along with the correct classification corresponding to each of the x_i's) for which ϕ_r (x_1, x_2, ..., x_k) = 0 and the labeled examples for which membership queries were made in order to evaluate ϕ_r.

Definition 2. *A concept class $\mathcal{C}$ is said to be polynomially learnable from example based queries iff there exist polynomials $p_1()$ and $p_2()$, and an algorithm $\mathcal{A}$, such that for any concept $c \in \mathcal{C}$ with representation r, $\mathcal{A}$ returns a representation $\hat{r}$ of a concept $\hat{c}$ that is equivalent to c when it is allowed to pose a number of example based queries bounded by $p_1(|r|)$ and see a set of examples S (including counterexamples returned by the example based queries) of size at most $p_2(|r|)$.*

The L^* algorithm is a method for exactly learning DFA (in polynomial time) from membership and equivalence queries [1].

2.3 Mistake Bounded Learning

Littlestone's mistake bounded learning model deals with the online learning scenario. Instead of presenting the learner with a set of labeled examples, the online model presents one example at a time and asks the learner to predict the class of each example it receives. After making this prediction, the learner is told whether its prediction was correct. The learner uses this information to improve its hypothesis. The mistake bounded model considers bounding the number of (explicit) prediction errors made by the learner in the worst case while learning (to predict) a target concept. Several concept classes are known to be learnable with the help of membership queries in addition to other example based queries (such as equivalence queries, subset queries, etc.). We consider an augmented mistake bounded learning model where the learner has access to a teacher who answers membership queries.

Definition 3. *A concept class $\mathcal{C}$ is polynomially learnable in the augmented mistake bounded model iff there exist polynomials $p_1()$ and $p_2()$, a teacher T capable of answering membership queries, and an online learning algorithm $\mathcal{A}$ such that for any concept c with representation r, $\mathcal{A}$ learns a representation $\hat{r}$ of a concept $\hat{c}$ that is equivalent to c when it is allowed to make at most $p_1(|r|)$ prediction errors on the sequence of examples it sees and pose (if required) at most $p_2(|r|)$ membership queries.*

DFA are known to be exactly learnable in this augmented mistake bounded model for online learning with membership queries (see [18] for a description of the *Incremental ID* algorithm). We show the relationship of this augmented mistake bounded model to the model of learning from example based queries. Note that this result subsumes Littlestone's result depicting the relationship between mistake bounded learning and learning by posing a bounded number of equivalence queries [14].

2.4 Learning from Teaching and Characteristic Sets

Goldman and Mathias have developed a teaching model for efficient learning of target concepts [7]. Their model takes into account the quantity of information that a good teacher must provide to the learner. An additional player called the *adversary* is introduced in this model to ensure that there is no unnatural collusion whereby the teacher directly gives the learner an encoding of the target concept.

Definition 4. *(Due to de la Higuera [9])*
A concept class $\mathcal{C}$ is semi-polynomially T/L teachable iff there exist polynomials $p_1()$ and $p_2()$, a teacher T, and a learner L, such that for any adversary ADV

and any concept c with representation r that is selected by ADV, after the following teaching session the learner returns the representation $\hat{r}$ of a concept $\hat{c}$ that is equivalent to c:
- *ADV gives r to T.*
- *T computes a teaching set S of size at most $p_1(|r|)$.*
- *ADV adds correctly labeled examples to this set.*
- *The learner uses the augmented set S and outputs $\hat{r}$ in time $p_2(||S||)$.*

In this model, a concept class for which the computation of both the teacher and the learner takes polynomial time and the learner always learns the target concept is called *polynomially T/L teachable*. Without the restrictive assumption that the teacher's computations be performed in polynomial time, the concept class is said to be *semi-polynomially T/L teachable*.

While studying the identification of languages in the limit, Gold proposed a *model for learning from given data* [5]. In this model, the learner when presented with a set of examples S must return a representation of a concept consistent with S. Further, the model postulates that there exists a *characteristic set* of examples for each language such that the learning algorithm upon seeing the characteristic set must output a representation equivalent to that of the target concept. This condition should be monotonic in that even if correctly labeled examples are added to the characteristic set, the algorithm would still infer the same language. This leads to the model for polynomial identifiability of concept classes from characteristic sets. It is based on the availability of a polynomial sized characteristic set for any concept in the concept class and an algorithm which when given a superset of a characteristic set is guaranteed to return, in polynomial time, a representation of the target concept.

Definition 5. *(Due to de la Higuera [9])*
A concept class $\mathcal{C}$ is polynomially identifiable from characteristic sets iff there exist two polynomials $p_1()$ and $p_2()$ and an algorithm $\mathcal{A}$ such that:
- *Given any set S of labeled examples, $\mathcal{A}$ returns in time $p_1(||S||)$ a representation r of a concept $c \in \mathcal{C}$ such that c is consistent with S.*
- *For every concept $c \in \mathcal{C}$ with corresponding representation r there exists a characteristic set S_c such that $||S_c|| = p_2(|r|)$ and if $\mathcal{A}$ is provided with a set $S \supseteq S_c$ then $\mathcal{A}$ returns a representation $\hat{r}$ of a concept $\hat{c}$ that is equivalent to c.*

The framework of the RPNI algorithm for learning DFA identifies a practical notion of a *teaching* or *characteristic* set of a DFA and demonstrates how DFA can be exactly learned (in polynomial time) from a training set that includes a characteristic set of the target DFA as a subset [15].

2.5 Learning from Simple Examples

The standard PAC model's requirement of learnability under all conceivable distributions is often considered too stringent for practical learning scenarios. Li and Vitányi have proposed a *simple-PAC learning model* for efficiently learning *simple* concepts. A concept class is said to be simple-PAC learnable if it is PAC

learnable under the class of *simple* distributions [12]. A distribution is simple if it is multiplicatively dominated by some enumerable distribution. The class of simple distributions includes a variety of distributions (such as all computable distributions). Further, the *simple distribution independent learning theorem* says that a concept class is learnable under the universal distribution $\mathbf{m}$ *iff* it is learnable under the entire class of *simple distributions* provided the examples are drawn according to the universal distribution [12]. Thus, the simple-PAC learning model is sufficiently general. Concept classes such as $\log n$*-term DNF* and *simple k-reversible DFA* are learnable under the simple-PAC model whereas their PAC learnability in the standard sense is unknown [12].

Denis *et al* proposed a learning model (called the PACS model) where examples are drawn at random according to the universal distribution conditional on the knowledge of the target concept [4]. Under this model, examples with low conditional Kolmogorov complexity given a representation r of the target concept are called simple examples. Specifically, for a concept with representation r, the set $S^r_{sim} = \{\alpha \mid K(\alpha|r) \leq \mu lg(|r|)\}$ (where μ is a constant) is the set of simple examples for that concept. Further, $S^r_{sim,rep}$ is used to denote a set of simple and representative examples of r. The PACS model restricts the underlying distribution to $\mathbf{m}_r$ (where $\mathbf{m}_r(\alpha) = 2^{-K(\alpha|r)+O(1)}$).

The learnability of *logarithmic Kolmogorov Complexity* DFA in the simple-PAC model and that of the entire class of DFA in the PACS model are shown in [16,17].

3 Relationships between the Learning Models

In this section we show the relationships between the different models for learning in helpful environments (see Fig. 1).

Theorem 1. *A concept class $\mathcal{C}$ is learnable in deterministic polynomial time using example-based queries iff it is learnable in the augmented mistake bounded framework with a polynomial mistake bound.*

Proof: We prove this result by showing that an algorithm using example based queries can be simulated using a mistake bounded learning algorithm and vice-versa. A similar strategy was used to show the equivalence of the mistake bounded learning model with the model for learning by posing a bounded number of equivalence queries [14].

Let $\mathcal{A}$ be an algorithm for learning $\mathcal{C}$ from example based queries. We derive a mistake bounded learning algorithm $\mathcal{B}$ as follows:

Algorithm $\mathcal{B}$

1. simulate $\mathcal{A}$ until it outputs a hypothesis $\hat{c}_0$ as its query
2. use $\hat{c}_0$ as the initial hypothesis
   let $i = 0$

3. *for* each observed example x *do*
   predict $\hat{c}_i(x)$
   if $\hat{c}_i(x) \neq c(x)$ — (where $c(x)$ is the correct classification of x)
   then return x as a counterexample in response to $\mathcal{A}$'s query
   let the next query output by $\mathcal{A}$ be the updated hypothesis $\hat{c}_{i+1}$
   let $i = i + 1$
   end if
end for

Note that $\mathcal{A}$ might make use of additional membership queries to assist in the computation of its hypotheses $\hat{c}_i$. The number of membership queries and other example based queries posed by $\mathcal{A}$ is polynomially bounded (by the definition of polynomial learning from example based queries). The number of mistakes made by the algorithm $\mathcal{B}$ is thus polynomially bounded.

Let $\mathcal{B}$ be a mistake bounded learning algorithm for $\mathcal{C}$ (i.e., $\mathcal{B}$ makes at most a polynomial number of mistakes and possibly uses polynomial number of membership queries to learn any concept $c \in \mathcal{C}$). We derive an algorithm $\mathcal{A}$ for learning $\mathcal{C}$ from example based queries as follows:

Algorithm $\mathcal{A}$

1. let $i = 0$
   let $\hat{c}_0$ be the initial hypothesis of $\mathcal{B}$
2. *repeat*
   use $\hat{c}_i$ to pose an example based query
   if the teacher's response is *yes*
   then output $\hat{c}_i$ and *halt*
   else present the counterexample x to $\mathcal{B}$
   $\mathcal{B}$ predicts $\hat{c}_i(x)$ (which is $\neq c(x)$ since x is a counterexample)
   give $c(x)$ to $\mathcal{B}$
   let $\hat{c}_{i+1}$ be the next hypothesis of $\mathcal{B}$
   let $i = i + 1$
   end if
   until eternity

Note that $\mathcal{B}$ may pose a polynomial number of membership queries during the computation of its hypotheses $\hat{c}_i$. Further, since $\mathcal{B}$ makes a polynomial number of mistakes it is clear that $\mathcal{A}$ poses at most a polynomial number of example based queries. This proves the theorem. □

Theorem 2. *(Due to Goldman and Mathias [7])*
Any concept class $\mathcal{C}$ learnable in deterministic polynomial time using example-based queries is semi-polynomially T/L teachable.

Proof: (The result is proved by showing how a teaching set is constructed by simulating the query based learning algorithm. The teaching set captures all the counterexamples and the additional instances, if any, that are generated during the evaluation of example based queries. The learner then simulates the

execution of the query based algorithm. However, instead of posing queries to a teacher, the learner evaluates the responses to example based queries using the labeled instances that appear in the teaching set.) □

Theorem 3. *(Due to de la Higuera [9])*
A concept class $\mathcal{C}$ is semi-polynomially T/L teachable iff *it is polynomially identifiable from characteristic sets.*

Proof: (This result is proved by identifying the characteristic set with the teaching set.) □

Lemma 1. *Let $c \in \mathcal{C}$ be a concept with corresponding representation r. If there exists a characteristic set S_c for c and a polynomial $p_1()$ such that S_c can be computed from r and $||S_c|| = p_1(|r|)$ then each example in S_c is simple in the sense that $\forall \alpha \in S_c,\ K(\alpha|r) \leq \mu \lg(|r|)$ where μ is a constant.*

Proof: Fix an ordering of the elements of S_c and define an index to identify the individual elements. Since $||S_c|| = p_1(|r|)$, an index that is $O(\lg(p_1(|r|))) = O(\lg(|r|)) = \mu \lg(|r|)$ bits long is sufficient to uniquely identify each element of S_c[3]. Since S_c can be computed from r we can construct a Turing machine that given r reads as input an index of length $\mu \lg(|r|)$ and outputs the corresponding string of S_c. Thus, $\forall \alpha \in S_c,\ K(\alpha|r) \leq \mu \lg(|r|)$ where μ is a constant independent of α. □

Lemma 2. *(Due to Denis et al [4])*
Suppose that a sample S is drawn according to $\mathbf{m}_r$. For an integer $l \geq |r|$, and $0 < \delta \leq 1$, if $|S| \geq l^\mu\ (\ln(2) + \ln(l^\mu) + \ln(1/\delta))$ then with probability greater than $1 - \delta$, $S^r_{sim} \subseteq S$.

Proof:
Claim 1: $\forall \alpha \in S^r_{sim},\ \mathbf{m}_r(\alpha) \geq l^{-\mu}$

$$\begin{aligned}\mathbf{m}_r(\alpha) &\geq 2^{-K(\alpha|r)} \\ &\geq 2^{-\mu \lg |r|} \\ &\geq |r|^{-\mu} \\ &\geq l^{-\mu}\end{aligned}$$

Claim 2: $|S^r_{sim}| \leq 2l^\mu$

$$\begin{aligned}|S^r_{sim}| &\leq |\{\alpha \in \{0,1\}^* \mid K(\alpha|r) \leq \mu \lg(|r|)\}| \\ &\leq |\{\alpha \in \{0,1\}^* \mid K(\alpha|r) \leq \mu \lg(l)\}| \\ &\leq |\{\beta \in \{0,1\}^* \mid |\beta| \leq \mu \lg(l)\}| \\ &\leq 2^{\mu \lg(l)+1} \\ &\leq 2l^\mu\end{aligned}$$

[3] Note that if the sum of the lengths of the examples belonging to a set is k then clearly, the number of examples in that set is at most $k + 1$.

Claim 3: $|S| \geq l^{\mu}\ (\ln(2) + \ln(l^{\mu}) + \ln(1/\delta))$ then $\Pr(S^r_{sim} \subseteq S) \geq 1 - \delta$

$$\Pr(\alpha \in S^r_{sim} \text{ is not sampled in one random draw}) \leq (1 - l^{-\mu})$$

(claim 4.1)

$$\Pr(\alpha \in S^r_{sim} \text{ is not sampled in } |S| \text{ random draws}) \leq (1 - l^{-\mu})^{|S|}$$

$$\Pr(\text{some } \alpha \in S^r_{sim} \text{ is not sampled in } |S| \text{ random draws}) \leq 2l^{\mu}(1 - l^{-\mu})^{|S|}$$

(claim 4.2)

$$\Pr(S^r_{sim} \not\subseteq S) \leq 2l^{\mu}(1 - l^{-\mu})^{|S|}$$

We would like this probability to be less than δ.

$$2l^{\mu}(1 - l^{-\mu})^{|S|} \leq \delta$$

$$2l^{\mu}(e^{-l^{-\mu}})^{|S|} \leq \delta, \quad \text{since } 1 - x \leq e^{-x} \text{ if } x \geq 0$$

$$\ln(2) + \ln(l^{\mu}) - |S|l^{-\mu} \leq \ln(\delta)$$

$$|S| \geq l^{\mu}\ (\ln(2) + \ln(l^{\mu}) + \ln(1/\delta))$$

Thus, $\Pr(S^r_{sim} \subseteq S) \geq 1 - \delta$ □

Corollary 1. *Suppose that a sample S is drawn according to $\mathbf{m}_r$. For an integer $l \geq |r|$, and $0 < \delta \leq 1$, if $|S| \geq l^{\mu}\ (\ln(2) + \ln(l^{\mu}) + \ln(1/\delta))$ then with probability greater than $1 - \delta$, $S^r_{sim,rep} \subseteq S$.*

Proof: Follows from Lemma 2 since $S^r_{sim,rep} \subseteq S^r_{sim}$. □

Theorem 4. *Any concept class that is semi-polynomially T/L teachable (or equivalently polynomially identifiable from characteristic sets) is probably exactly learnable in the PACS model.*

Proof: Lemma 1 shows that if there exists a polynomial sized teaching (characteristic) set S_c of examples for a concept c then the individual examples belonging to the teaching set are simple (in that they have logarithmic Kolmogorov complexity). Lemma 2 shows that a polynomial sized sample S drawn according to the universal distribution $\mathbf{m}_r$ is sufficient to include all simple examples with a high probability. Further, corollary 1 shows that with high probability $S_c \subseteq S$ (we equate S_c with $S^r_{sim,rep}$). Since the concept class $\mathcal{C}$ is semi-polynomially T/L teachable, there exists an algorithm $\mathcal{A}$ that in polynomial time exactly learns any concept $c \in \mathcal{C}$ from any set of examples that includes S_c as a subset. The PACS learning algorithm can be formulated as follows. Draw a polynomial sized sample S according to $\mathbf{m}_r$ and use it as the training set for algorithm $\mathcal{A}$. Thus, $\mathcal{C}$ is probably exactly learnable in the PACS model. □

Theorem 5. *(Due to Castro and Guijarro [3])*
If a concept class $\mathcal{C}$ is learnable in the PACS model then the concept class $logK(\mathcal{C}) = \{c \in \mathcal{C} \mid \mathcal{R}(c) = r \text{ and } K(r) \leq \kappa \lg(|r|) \text{ where } \kappa \text{ is a constant}\}$ (i.e., the set of concepts whose corresponding representations have logarithmic Kolmogorov complexity) is learnable in the simple-PAC model.

Proof: (This result is proved by showing the relationship between the universal distributions $\mathbf{m}$ and $\mathbf{m}_r$.) □

4 Collusion and Learning in Helpful Environments

Learning models that involve interaction between a knowledgeable teacher (an oracle) and a learner are vulnerable to unnatural *collusion* wherein the teacher passes information about the representation of the target concept as part of the training set [10,7]. The teacher and learner can *a-priori* agree on some suitable binary encoding of concepts. The teacher can then pass the representation of the target concept r to the learner as a suitably labeled example. In the event that the target concept cannot be suitably encoded as a single labeled example, the teacher can break the representation r into smaller groups and pass these groups as appropriately labeled examples to the learner. For example, an encoding of the target concept could be passed via the counterexamples (in the case of learning from example based queries) or via the first few examples of a teaching set (in the case of learning from teaching sets or characteristic samples). The learner can thus quickly discover the target concept without even considering the labels of the training examples! The *teaching model* due to Jackson and Tomkins [10] prevents this coding of the target concept by requiring that the learner must still succeed if the teacher is replaced by an adversary (who does not code the target concept as the teacher above). Further, they argue that in their model the learner can stop only when it is convinced that there is only one concept consistent with the information received from the teacher i.e., the teacher does not tell the learner when to stop. Otherwise learning would be trivialized in that the teacher passes groups of n bits to the learner (as training examples) and when sufficient number of bits have been passed to the learner so as to reconstruct the representation r of the target concept, the teacher tells the learner to stop. Goldman and Mathias' work on *polynomial teachability* [7] shows that an *adversary* whose task is to embed the training set (also called *teaching set*) provided by the teacher into a larger set of correctly labeled examples is sufficient to prevent this type of collusion.

Another (perhaps more subtle) form of collusion is possible in the models for learning in helpful environments. For simplicity let us assume that the target representation can be encoded using a single labeled example. Consider the polynomial teachability model. The adversary augments the teaching set with correctly labeled examples. Assuming that the augmented set is suitably shuffled the learner cannot directly identify the target without even considering the class labels. However, the learner can decode each of the labeled examples in a fixed order (say lexicographic order). For each example that represents a valid concept (in $\mathcal{C}$), the learner checks whether the decoded concept is consistent with the teaching set and outputs the first concept that passes this consistency test. Note that a suitably formulated teaching set can ensure that one and only one concept is consistent with it. Here, the learner is provided with an encoding of the target concept but must perform some computation (the consistency check) in order to suitably identify the target. However, this method of identifying the target concept is potentially easier and thus more attractive than the typical algorithms that learn from a given teaching set (for instance the RPNI algorithm for learning DFA [15]).

The other models for learning in helpful environments are also vulnerable to this form of collusion. In the PACS learning model the target concept r is itself a simple example (since $K(r|r)$ is very small). Thus, r has a very high probability of being drawn under $\mathbf{m}_r$. By using the decoding and consistency check trick illustrated above the learner can efficiently identify the target. Counterexamples can be used to formulate a collusive learning scheme in the model for learning from example based queries. Here the teacher can encode the target concept in the counterexample it provides to the learner. The learner can attempt to decode the counterexample. If the counterexample does not represent a valid concept then the execution of the learning algorithm continues its normal execution. However, if the counterexample represents a valid concept then the learner can pose an equivalence query to determine if the example is the target concept. If the teacher replies *yes* then the learner outputs the target and halts. Otherwise it takes the counterexample and repeats the above process. This method is potentially more efficient in terms of computation time. From theorem 1 we know that if there exists a deterministic polynomial time algorithm for learning a concept class using example based queries then it is easy to construct an algorithm for learning the concept in the augmented mistake bounded learning framework. Thus, the collusive learning strategy for learning from example based queries can be used to design a strategy to learn the concept in the augmented mistake bounded learning framework.

It is clear that the frameworks for learning in helpful environments admit unnatural collusion. Any learnability results within models that admit collusion can be criticized on the grounds that the learning algorithm might be collusive. One method of avoiding collusive learning is to tighten the learning framework suitably. Collusion cannot take place if the representation of the target concept cannot be directly encoded as part of the training set or if the learner cannot efficiently decode the training examples and identify the one that is consistent with the training set. In the event that the learning framework cannot be suitably tightened to avoid collusion, one might provide a learning algorithm that does not rely on collusion between the teacher and the learner. For instance, the L^* algorithm for learning DFA from membership and equivalence queries [1], the IID algorithm for incremental learning of DFA using membership queries [18], and the RPNI algorithm for learning DFA from characteristic samples [15] are examples of non-collusive algorithms in learning frameworks that admit collusion. Obtaining a general answer to the question of collusion in learning would require the development of much more precise definitions of collusion and collusion-free learning than are currently available. A detailed exploration of these issues is clearly of interest.

5 Summary

We have presented above the inter-relationships between different models for learning in helpful environments. The PACS model for learning from simple examples naturally extends the results obtained for the deterministic learning

models (example based queries, mistake bounded learning, polynomial teachability, and polynomial identifiability from characteristic sets) to a probabilistic learning framework. This work opens up several interesting questions that remain to be answered. For instance, does PACS learnability imply learnability from example based queries or polynomial teachability? Or does there exist a concept class that is PACS learnable but is not learnable from example based queries or is not polynomially teachable? Similarly, does polynomial teachability imply learnability in the mistake bounded model? We have also addressed the important issue of collusion as it relates to the models for learning in helpful environments. We have shown how the models studied in this paper admit multiple learning algorithms including some seemingly collusive ones. Additional research is required to suitably address the issues of collusion and collusion-free learning.

Acknowledgements. This work was supported in part by grants from the National Science Foundation (9409580, 9982341) to Vasant Honavar. Rajesh Parekh would like to thank the Allstate Research and Planning Center for the support he has received while conducting this research.

References

1. D. Angluin. Learning regular sets from queries and counterexamples. *Information and Computation*, 75:87–106, 1987.
2. D. Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, 1988.
3. J. Castro and D. Guijarro. Query, PACS and simple-PAC learning. Technical Report LSI-98-2-R, Universitat Polytéctica de Catalunya, Spain, 1998.
4. F. Denis, C. D'Halluin, and R. Gilleron. Pac learning with simple examples. *STACS'96 - Proceedings of the* 13^{th} *Annual Symposium on the Theoretical Aspects of Computer Science*, pages 231–242, 1996.
5. E. M. Gold. Complexity of automaton identification from given data. *Information and Control*, 37(3):302–320, 1978.
6. S. Goldman and H. Mathias. Teaching a smarter learner. In *Proceedings of the Workshop on Computational Learning Theory (COLT'93)*, pages 67–76. ACM Press, 1993.
7. S. Goldman and H. Mathias. Teaching a smarter learner. *Journal of Computer and System Sciences*, 52:255–267, 1996.
8. D. Haussler, M. Kearns, N. Littlestone, and M. Warmuth. Equivalence of models for polynomial learnability. *Information and Computation*, 95:129–161, 1991.
9. Colin de la Higuera. Characteristic sets for polynomial grammatical inference. In L. Miclet and C. Higuera, editors, *Proceedings of the Third ICGI-96, Lecture Notes in Artificial Intelligence 1147*, pages 59–71, Montpellier, France, 1996.
10. J. Jackson and A. Tomkins. A computational model of teaching. In *Proceedings of the Workshop on Computational Learning Theory (COLT'92)*, pages 319–326. ACM Press, 1992.
11. M. Kearns and L. G. Valiant. Cryptographic limitations on learning boolean formulae and finite automata. In *Proceedings of the* 21^{st} *Annual ACM Symposium on Theory of Computing*, pages 433–444, New York, 1989.
12. M. Li and P. Vitányi. Learning simple concepts under simple distributions. *SIAM Journal of Computing*, 20(5):911–935, 1991.

13. M. Li and P. Vitányi. *An Introduction to Kolmogorov Complexity and its Applications,* 2^{nd} *edition*. Springer Verlag, New York, 1997.
14. N. Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2:285–318, 1988.
15. J. Oncina and P. García. Inferring regular languages in polynomial update time. In N. *et al* Pérez, editor, *Pattern Recognition and Image Analysis*, pages 49–61. World Scientific, 1992.
16. R. Parekh and V. Honavar. Simple DFA are polynomially probably exactly learnable from simple examples. In *Proceedings of the Sixteenth International Conference on Machine Learning (ICML'99)*, pages 298–306, Bled, Slovenia, 1999.
17. R. G. Parekh and V. G. Honavar. Learning DFA from simple examples. In *Proceedings of the Eighth International Workshop on Algorithmic Learning Theory (ALT'97), Lecture Notes in Artificial Intelligence 1316*, pages 116–131, Sendai, Japan, 1997. Also presented at the *Workshop on Grammar Inference, Automata Induction, and Language Acquisition* (ICML'97), Nashville, TN. July 12, 1997.
18. R. G. Parekh, C. Nichitiu, and V. G. Honavar. A polynomial time incremental algorithm for regular grammar inference. In V. Honavar and G. Slutzki, editors, *Proceedings of the Fourth ICGI-98, Lecture Notes in Artificial Intelligence 1433*, pages 37–49, Ames, IA, 1998.
19. L. Pitt and M. K. Warmuth. Reductions among prediction problems: on the difficulty of predicting automata. In *Proceedings of the* 3^{rd} *IEEE Conference on Structure in Complexity Theory*, pages 60–69, 1988.
20. L. Valiant. A theory of the learnable. *Communications of the ACM*, 27:1134–1142, 1984.