# A Computational Model of Teaching

**Jeffrey Jackson**
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
jcj@cs.cmu.edu

**Andrew Tomkins**
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
andrewt@cs.cmu.edu

## Abstract

Goldman and Kearns [GK91] recently introduced a notion of the *teaching dimension* of a concept class. The teaching dimension is intended to capture the combinatorial difficulty of teaching a concept class. We present a computational analog which allows us to make statements about bounded-complexity teachers and learners, and we extend the model by incorporating *trusted information*. Under this extended model, we modify algorithms for learning several expressive classes in the exact identification model of Angluin [Ang88]. We study the relationships between variants of these models, and also touch on a relationship with distribution-free learning.

## 1 INTRODUCTION

In the eight years since Valiant's seminal paper on learnability was published [Val84], computational learning theory has been an active and productive field. Several different learning models have been proposed, each attempting to model a different aspect of learning. Many of these models envision a teacher who interacts in some way with the learner (e.g., by providing counterexamples to hypotheses), but in virtually all of these models the learning process is solely the responsibility of the learner. The teacher is usually abstracted as an oracle of some form, neutral at best, and often adversarial.

Driven by the notion that real-world learning is often highly teacher-dependent, several researchers have suggested moving some of the computation from the learner to the teacher [Nat87, CVS88, GRS89, GK91]. It is appealing to allow some model of the cooperation that happens in, for instance, a classroom. Goldman and Kearns [GK91] take the limit of this process, and ask the question: What kind of teacher would be so smart that any reasonable student would understand the material at the end of the lecture? They introduce the notion of *teaching dimension*, which corresponds in the analogy above to the length of the shortest lecture a teacher can give that will force every reasonable student to understand the concept.

More precisely, in the Goldman-Kearns teaching model, a helpful teacher provides a set of examples to a learner. The teacher knows that the learner produces hypothesis concepts which are consistent with the examples seen but knows nothing else about the learner. The teaching dimension of a concept class is the minimum number $m$ such that for every concept in the class, there exists a set of $m$ examples consistent with that concept and no other. A teacher which can find such a set for each concept can thus teach any concept to any consistent learner with $m$ or fewer examples. (See [GK91] for a formal definition).

We modify the Goldman-Kearns model in several ways. Under our formulation we study teacher/learner pairs in which the teacher chooses examples tailored to a particular learner, rather than the Teaching Dimension paradigm in which the teacher constructs examples that work for any consistent learner. It is therefore meaningful to consider teacher-learner pairs that are time-bounded, our second change to the earlier formalism.

Finally, and perhaps most significantly, we introduce the notion of teaching with trusted information. That is, we allow the teacher to transmit a small number of bits about the target concept, information which the learner accepts without question. For example, the number of terms in a monotone DNF could be a form of trusted information. With the addition of such information some interesting concept classes, including monotone DNF and decision lists, are easily shown to be teachable in polynomial time.

We discuss the relative power of this model compared to exact identification [Ang87] with and without counterexamples and show that several interesting classes for which exact identification can be achieved, including regular sets represented by DFA's, read-once formulas, and read-once decision trees, are teachable in our model. We also show that if our teacher is allowed to be computationally unbounded, the set of classes that can be taught under our model contains the set that can be learned under exact identification with both membership and equivalence queries. We also briefly look

at a relationship between teaching and the distribution-free model.

## 2 PRELIMINARIES

A *concept* $c$ is a Boolean function on an *instance space* $X$. We consider only finite or countable instance spaces because our model of teaching currently considers only exact identification of concepts. A *concept class* $C$ is a set of concepts over some instance space. A *representation class* $R$ for a concept class $C$ is a set of *representations* (strings) $r$ such that there is some mapping $\mu$ of $R$ onto $C$. The *length* with respect to $R$ of a concept, $|c|$, is the length $|r|$ of the shortest representation $r \in R$ such that $\mu(r) = c$. At times we may refer to a concept class when the context indicates we mean a representation class; in such cases we have in mind any "reasonable" representation of the concept class.

An *instance* $x$ is an element of the instance space. Each instance has an associated complexity parameter referred to as the *length* of the instance. A pair $\langle x, c(x)\rangle$ is called an $example$ of the concept $c \in C$, and a set of examples is called a *teaching sequence*. The $c(x)$ portion of an example is called its *label*. The *length* of a teaching sequence $s$, $|s|$ is the sum of the lengths of all instances in the sequence.

We wish to bound the running time of our algorithms in terms of the complexity $n$ of the target concept $c$ chosen from $C$. For example, for DFA's, $n$ could be the number of states in the minimal DFA for the target language.

A representation class $R$ for $C$ is *exactly identifiable* if there exists a deterministic algorithm which can, with help from certain oracles, learn without error any concept in $C$. For every concept $c \in C$ the algorithm must run in time polynomial in $|c|$, in the complexity parameter $n$ of $C$, and in the length of the longest instance seen. The two most widely studied oracles have been membership and equivalence oracles. A *membership* oracle for a target concept $c$ when given an instance $x$ returns $c(x)$. An *equivalence* oracle for $c$ when given the representation of a hypothesis $h \in R$ returns either "yes" to indicate that $h = c$ or an instance $x$ such that $h(x) \neq c(x)$ (a *counterexample*). The combination of an equivalence and membership oracle is sometimes called a *minimally adequate teacher* [Ang87] (see [Ang87] and [Ang90] for formal definitions).

It is assumed that the reader is familiar with distribution-free (PAC) learning [Val84].

## 3 A NEW NOTION OF TEACHING

We develop two models. The first directly deals with computational issues, while the second addresses the limitations of teaching by examples alone.

### 3.1 TEACHING WITH EXAMPLES ONLY

We first propose the following notion of teaching, which incorporates both teacher and learner, and requires that each performs some of the work:

**Definition 1** *A representation class $R$ for a concept class $C$ is* polynomial-time teachable *if there exists a pair of algorithms $T$ and $L$ with the following properties:*

1. *When started on any representation $r \in R$, $T$ outputs a teaching sequence $s$ and terminates in time polynomial in $n$ and $|r|$.*
2. *Algorithm $L$, given $s$, runs for time polynomial in $n$ and $|s|$ (which is polynomial in $|r|$), outputting a representation $r'$ such that $\mu(r') = \mu(r) = c$.*
3. *If any adversarial teacher $A$ (not necessarily time-bounded) sends a set of examples $s'$ consistent with $c$ but different than $s$ then $L$ outputs either some $r''$ such that $\mu(r'') = c$ or $L$ outputs no concept at all. $L$ runs in time polynomial in $n$ and $|s'|$.*

Aside from time-boundedness, the primary difference between this model and that of Goldman-Kearns is with respect to avoiding what we will call *cheating*. To understand the problem, consider a teacher and learner which both have in mind some binary encoding of concepts. Then if each instance transmitted by the teacher corresponded to an appropriate $n$ bits of the encoding of the target concept, the learner could quickly discover what the concept was without learning in any real sense–it doesn't even look at the labels! Any reasonable model of teaching must therefore limit the teacher and/or learner in some way to forestall such cheating.

In the Goldman-Kearns model, cheating is avoided by requiring the teacher to produce a set of examples which will cause *any* consistent learner to hypothesize the correct concept. We propose phrasing the interaction between teacher and learner as a modified Prover-Verifier session [GMR85] in which the learner and teacher can collude, but no adversarial teacher (in the IP sense) can cause the learner to output an incorrect hypothesis.

On the surface, this approach to cheating avoidance might seem to give our teacher more power. While the previous model assumes an oblivious teacher, our teacher is tailored to the learner and is even allowed to simulate the learner. Thus, we can assume that the teacher knows the state of the learner at all times.[1] However, the following shows that our teacher is no more powerful than that of Goldman-Kearns.

**Fact 1** *If a representation class is polynomial-time teachable then for every representation the teacher must produce a teaching sequence which is consistent with exactly one concept.*

**Proof:** First, note that by definition every teaching sequence must be consistent with at least one concept. Assume that there is some representation $r$ for which the teacher $T$ produces a sequence $s$ consistent with two distinct concepts $\mu(r) = c$ and $c_2$. The learner $L$ must produce a representation $r'$ such that $\mu(r') = c$. But an adversarial teacher $A$ could choose to teach a representation $r_2$ such that $\mu(r_2) = c_2$

[1] This property would not hold if the model was extended to randomized learners; in this paper, as is common in exact identification research, we consider only deterministic learners.

using the same sequence $s$, in which case $L$ would be fooled into outputting an incorrect representation. □

Similarly, it is not hard to see that if there is a teacher that produces a sequence for every representation $r$ in some class such that the sequence is consistent with only the concept $\mu(r)$ then an arbitrary consistent learner can be used as the other member of the teacher/learner pair in our model. Thus the difference between the two models is that the explicit introduction of teacher/learner pairs facilitates the introduction of computational complexity issues.

In our model both teacher and learner must be realizable polynomial-time algorithms. As we will see, this immediately limits somewhat the classes which are teachable; removing the computational bound on the teacher, as in the definition of the class IP[GMR85], is an interesting alternative which we explore briefly as well.

### 3.2 TEACHING WITH TRUSTED INFORMATION

The above definition accomplishes two goals. First, it incorporates computational constraints on the teaching process. And second, it allows teacher-learner collusion while avoiding "cheating." However, the interaction between teacher and learner is still constrained to be a sequence of examples. This seems unnecessarily restrictive.

Consider the following example. Goldman and Kearns present an algorithm for teaching monotone $k$-term DNF for any fixed $k$ but note that the algorithm requires the learner to know $k$. If the teacher could transmit at least a little information—such as the value of $k$—that the learner simply "takes on faith" without verification, then the teaching could proceed without the restriction.

Thus a natural extension to our model is to allow the teacher to transmit a small amount of "trusted" information to the learner, information which the learner accepts as true. This information might take different forms for different classes. For monotone $k$-term DNF's the obvious information to send is $k$. For other classes, as we show below, it may be the number of relevant variables in a concept or a size measure of the concept. In every case the bits will be the output of some deterministic function applied to the representation the teacher has chosen. The problem, of course, is that the model must not give the teacher so much power that it can "cheat" with the learner as described above. The model we define appears to overcome this problem.

**Definition 2** *A representation class $R$ for a concept class $C$ is* polynomial-time teachable with trusted information *if there exist a pair of algorithms $T$ and $L$ and a deterministic function $f : R \to \{0,1\}^*$ with the following properties:*

1. *$|f(r)| = O(\log(|r|))$ for all $r$.*
2. *When started on any representation $r \in R$, $T$ outputs trusted bits $f(r) = b$ followed by a teaching sequence $s$ and terminates in time polynomial in $n$ and $|r|$.*
3. *Algorithm $L$, given $b$ and $s$, runs for time polynomial in $n$ and $|s|$ and outputs a representation $r'$ such that $\mu(r') = \mu(r) = c$.*
4. *If any adversarial teacher $A$ (not necessarily time-bounded) sends trusted bits $b$ and a set of examples $s'$ consistent with $c$ but different than $s$ then $L$ outputs either some $r''$ such that $\mu(r'') = c$, or $L$ outputs no concept at all. $L$ runs in time polynomial is $n$ and $|s'|$.*

The choice of "logarithmic" to represent the "small" amount of information that seemed intuitively reasonable is not arbitrary. If we wish to transmit the $k$ of, say, a $k$-term-DNF $F$ then we will in general need a least $O(\log(|F|))$ bits, so our definition is in some sense a minimal extension of the original. Furthermore, with logarithmic trusted bits it is possible to teach many if not all of the relatively natural concept classes which are known to be exactly identifiable with a polynomial-time minimally adequate teacher, that is, with teachers having equivalence oracles which run in polynomial time. Logarithmic trusted information is also enough to teach any exactly identifiable class if the teacher is unbounded, as shown later. Finally, this is not enough information to allow cheating: we now demonstrate a limit on how much can be taught by a polynomial-time teacher even with logarithmic trusted information.

**Theorem 2** *If $NP \not\subset P/\text{poly}$ then DNF is not teachable with trusted information.*

**Proof:** Assume otherwise, and let $T$, $L$, and $f$ be the teacher, learner, and trusted bit functions which can teach DNF. If $P \neq NP$ (true if $NP \not\subset P/\text{poly}$) then for $T$, a polynomial-time algorithm, there exist "hard" representations of DNF's each of which $T$ cannot find a falsifying assignment for but only some of which are truly tautologies. Furthermore, if $NP \not\subset P/\text{poly}$ then there must be some $r_+$ a tautology and $r_-$ not a tautology which are hard for $T$ and for which $f(r_+) = f(r_-)$. In other words, $f$ cannot partition the hard tautologies into one set of equivalence classes and hard falsifiable functions into another set.

To see this, assume otherwise, that is, that the values of $f$ partition the hard functions. Then $f$, a polynomial-time function producing a logarithmic number of bits, could be used by $T$ along with a polynomial-size "hint" (actually one for each size DNF) specifying which of the polynomially many values of $f$ represent hard tautologies in order to solve the $NP$-complete problem of DNF-falsifiability. But then $NP \subset P/\text{poly}$.

Thus to teach $r_-$, $T$ will transmit trusted bits $f(r_+)$ followed by examples all of which are labeled `true`, and $L$ will output a representation $r'$ such that $\mu(r') = \mu(r_-)$; that is, $r'$ is not a tautology. However, the same trusted bits and teaching sequence are consistent with $r_+$, so an adversarial teacher can force $L$ to output the wrong concept on target $r_+$. □

By duality, CNF is not teachable under the same complexity assumptions. It is easy to see that the proof can be extended to 3-DNF and 3-CNF as well.

# 4 TEACHING AND EXACT IDENTIFICATION

It is natural to explore the relationship between teachable concept classes, both with and without trusted information, and concept classes for which exact identification can be achieved.

By definition, the membership and equivalence oracles of exact identification are not computationally limited. For example, the class $k$-CNF can be exactly identified with equivalence queries alone [Ang88] even though finding counterexamples to the hypothesis `false` may require solving an NP-complete problem. Thus there are some classes which can be exactly identified but which are not teachable according to our definition. However, several interesting classes have polynomial-time minimally adequate teachers, as discussed below. In the sequel we will consider only such oracles unless otherwise noted.

## 4.1 EXACT IDENTIFICATION WITH MEMBERSHIP QUERIES

The qualitative differences between exact identification and teaching are worth highlighting. Under the assumption of polynomial-time oracles, the only difference between teaching and exact identification with membership queries is the information available to the algorithm making the query. A teacher can use the representation of the concept to drive its selection of examples, whereas a learner in the exact identification model must make queries based only the results of past queries. Thus, the classes learnable under exact identification with membership queries only is a subset of the classes teachable without trusted information.

This containment is in fact proper. First, define a *decision list* as a list of at most $n$ literals each with an associated Boolean value. A decision list defines a function over $n$-bit Boolean input strings as follows: output the value associated with the first satisfied literal, and output the complement of the last literal's value otherwise.[2] Figure 1 contains an example decision list. Define the class of *full decision lists* to be the class which for instances of $n$ bits consists of all possible decision lists of exactly $n$ variables (i.e. lists without any irrelevant variables). This class forms a natural separating class:

**Lemma 3** *The class of full decision lists is polynomial-time teachable (without trusted information).*

**Proof:** The teacher $T$ will provide examples to the learner $L$ which prove that some variable $v_i$ can legitimately be placed at the end of the decision list; repeating this process in the obvious way teaches the entire list. Assuming that $v_i$ is indeed the last variable in $T$'s representation of the list, $T$ first provides an instance $x$ which passes through to the 0 leaf at $v_i$. It then provides the (possibly empty) sequence of all examples whose instances differ from $x$ in just one bit (other than $v_i$) and which are labeled 1. Let $S$ represent the set of variables which were toggled in this sequence and $V$ the set of all $n$ variables. $T$ next flips the value of $v_i$ in $x$ and sends this instance, which will be labeled 1, to $L$. Call this instance $x'$. Finally, $T$ sends one example for each of the variables in $V - S - \{v_i\}$; the instance in each example will differ from $x'$ in one of these variable positions, and each example will be labeled 0. Figure 1 gives an example decision list and initial teaching sequence.

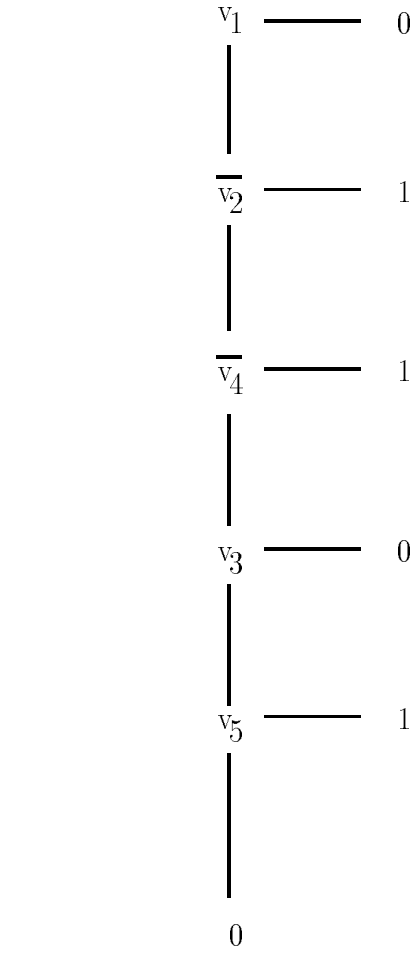


Figure 1: Example Decision List And Initial Portion Of Teaching Sequence.

To see that this sequence proves that $v_i$ can be placed at the end of the list, let $v_j$ be the variable which immediately follows $v_i$ in some decision list representing the target concept. Without loss of generality, assume $v_i$'s leaf value is 0. Then if $v_j$ has a leaf of value 0, the list having these two nodes reversed represents the same concept. But it is not hard to see that if $v_j$ does not have a 0 leaf then the teaching sequence

[2] This definition differs somewhat from the norm. Our definition forces irrelevant variables to be left out of the list, simplifying the presentation of our results.

described above cannot be produced. □

On the other hand, full decision lists can require exponentially many membership queries to learn: consider all possible conjuncts of $n$ literals. Thus we have the following:

**Theorem 4** *The set of representation classes which can be exactly identified with with a polynomial-time membership oracle is properly contained in the the set of representation classes which are polynomial-time teachable.*

### 4.2 EXACT IDENTIFICATION WITH A POLYNOMIAL-TIME MINIMALLY ADEQUATE TEACHER

We have shown in Theorem 2 that under a certain complexity assumption, there are classes which can be exactly identified with a minimally adequate teacher but which cannot be taught with trusted information. We can prove a similar result without complexity assumptions for teaching without trusted information.

**Fact 5** *Decision Lists are exactly identifiable with a minimally adequate teacher but are not teachable without trusted information.*

**Proof:** In order to teach the concept that is false on all inputs, a teacher/learner pair must eliminate the $2^n$ possible concepts that are true on exactly one input, all of which are representable as decision lists. But any example can eliminate at most one such concept. So no teaching algorithm can exist.

It is well-known that a variation of Rivest's decision list learning algorithm works in the exact identification model. □

We now discuss several classes which can be identified with a polynomial-time minimally adequate teacher and indicate how they can be taught in polynomial time with trusted information. These problems are included to demonstrate the power of trusted information and to give examples of various ways trusted bits can be used to encode a termination condition.

**Regular sets:** Angluin [Ang87] has given an algorithm for the exact identification of regular sets, represented by deterministic finite automata (DFA's), with membership and equivalence queries. A subroutine of this algorithm uses membership queries alone to find a minimal DFA consistent with any finite set of strings labeled according to whether or not they are in some regular language. The subroutine takes time polynomial in the number of states of the minimal DFA as long as the labeled strings are polynomially long. Angluin also notes that a polynomial-time equivalence oracle can be constructed for the class of regular sets represented by DFA's. Since a deterministic learner exists, there is a polynomial-time teacher which can determine all of the learner's queries for a given regular set and provide the appropriate examples for the learner. Furthermore, since the hypotheses the learner constructs are all minimal, the learner knows when to stop if it knows the number of states in the minimal DFA for the set. This can be sent by the teacher to the learner as a logarithmic number of trusted bits.

**Read-once formulas:** Hancock and Hellerstein [HH91], in an extension of several earlier results [AHK89, Han90, HK91], show that read-once Boolean formulas over a fairly rich basis that they call $\beta^k$ can be exactly identified with a minimally adequate teacher. $\beta^k$ consists of negations, thresholds, and mod $c$ gates for $c$ less than $k$. In fact, they actually prove a stronger result: read-once formulas over $\beta^k$ can be exactly identified with membership queries alone given the set of relevant variables and "justifying assignments" for each. For some function $f$ a justifying assignment for a variable $x_i$ is simply an assignment of values to all variables such that $f(x_1x_2 \ldots x_i \ldots x_n) \neq f(x_1x_2 \ldots \bar{x}_i \ldots x_n)$, where $\bar{x}_i$ denotes taking the complement of the value of $x_i$. For read-once formulas over $\beta^k$ it is clearly polynomial time to find justifying assignments for each relevant variable given the formula. Thus a teacher can be constructed which first sends as trusted information the number of relevant variables followed by justifying assignments for each and the responses to the membership queries which the learner would ask in the query model.

**$\mu$-formula decision trees:** $\mu$-formula decision trees are Boolean decision trees in which each node is a read-once formula over the AND/OR/NOT basis and no variable appears more than once in the entire tree. The tree is evaluated by first evaluating the root read-once formula, choosing the left or right subtree according to whether the formula evaluates to 0 or 1, and recursing on that subtree until a leaf is reached, at which time the leaf's value is output. Thus $\mu$-formula decision trees are a generalization of both read-once formulas over the standard basis and of Boolean decision trees with single variable nodes. Hancock [Han90] gives an algorithm for learning such trees which uses each counterexample it receives from an equivalence query to incorporate one or more new variables into its hypothesized tree. The hypothesis is correct once the processing which incorporates the last relevant variable has been completed. Thus once again the learner could terminate without help from an equivalence oracle if it was told how many relevant variables to expect.

#### 4.2.1 A Separating Class

As noted above, the class $k$-CNF can be exactly identified with a minimally adequate teacher, but under certain assumptions cannot be taught under our definition. Here we show a concept class which can be taught (without trusted information) but which cannot, under certain assumptions, be exactly identified using even a computationally unbounded minimally adequate teacher.[3] Avrim Blum[Blu90] has constructed a concept class for which exact identification is not possible given the existence of one-way functions but which is easily taught by giving the learner just one judiciously selected example (we regard this example as the representation of its concept, a slight departure from Blum's definition). Thus we have the following:

**Theorem 6** *Assuming the existence of one-way functions, the set of concept classes which are exactly identifiable with computationally unbounded membership and equiva-*

[3] This result holds even when restricted to Boolean domains.

*lence queries is incomparable to the set of those which are polynomial-time teachable.*

In fact, Blum's class was constructed to demonstrate a separation between approximate and exact learning models. Conceptually, the concept class is an ordered set of strings with the property that knowing string $i$ tells a learner how to detect and correctly label all strings $j > i$ from among the many "bad example" strings throughout which this set is pseudo-randomly scattered. By changing the class so that only the very first "good example" string contains this information we obtain a new class which is teachable but not learnable in either an approximate (PAC) or an exact sense.

### 4.3 EXACT IDENTIFICATION WITH AN UNBOUNDED TEACHER

On the surface, it seems that any language which can be exactly identified with a polynomial-time minimally adequate teacher should be polynomial-time teachable with trusted information. Given that an identification algorithm exists, all that a helpful teacher need do is supply its learner with the oracle responses that the identification algorithm would receive and somehow indicate when the learner can stop via the trusted information.

While there may be a general method for inferring an appropriate logarithmic amount of stopping information from identification algorithms, we have only been able to demonstrate such a method in a relaxed model in which the teacher is computationally unbounded:

**Theorem 7** *Any representation class which can be exactly identified is teachable with trusted information by a computationally-unbounded teacher.*

**Proof:** Given any representation $r$ of a concept $c$ in an identifiable class, the teacher $T$ first determines the *maximum* number of equivalence queries which some fixed identification algorithm $I$ would make before being told to halt, where the maximum is taken over all possible minimally adequate teachers (or more precisely over all possible sequences of valid counterexamples).

Since $I$ always halts in polynomial time regardless of the counterexample sequence, this maximum must be a polynomial in $|c|$. Thus logarithmic trusted bits are sufficient for $T$ to transmit this number to the learner $L$. $T$ then chooses such a maximal sequence and passes it, appropriately interleaved with responses to $I$'s expected membership queries, to $L$. $L$ then attempts to verify that the oracle sequence supplied by $T$ is valid for $I$ and contains the maximal number of counterexamples indicated by the trusted information. If this verification succeeds, $L$ outputs $I$'s final equivalence query hypothesis as its representation of $c$. □

Thus, showing that a representation class is not teachable with a computationally unbounded teacher also shows that the class cannot be exactly identified. Furthermore, negative results in the teaching model may shed light on algorithmic methods which will not achieve exact identification. For example, consider identification of DNF. A straightforward algorithm for exact identification of monotone DNF uses membership queries to produce one new term in the hypothesis from each counterexample; that is, equivalence queries are only used to identify additional terms. The following fact implies that equivalence queries must be used for more than this for general DNF identification:

**Fact 8** *DNF is not teachable by a computationally-unbounded teacher if the trusted information transmitted is the minimum number of terms in any DNF representation of the concept being taught.*

**Proof Sketch:** We show a set of DNF's all of which have the same minimal number of terms but for which the teaching dimension is exponential in the number of variables $n$. The set consists of the OR of $n$ variables, a function which has a single 0 value at the zero vector $\bar{0}$, and all the functions which have exactly two 0 values, one at $\bar{0}$ and the other at a vector which is at least Hamming distance 2 away (i.e. a vector which has at least two 1's).

All of these functions have $n$ terms in their minimal representation. The OR is obvious. As an example of what minimal representations of the other functions look like, consider the function which has 0's at $\bar{0}$ and $\bar{1}$: $x_1\bar{x_2} + x_2\bar{x_3} + \cdots + x_n\bar{x_1}$. The other functions will be similar but one or more of the terms will consist of single variables while the remainder of the terms will form a sort of cycle similar to the above. That every one of the functions in the set can be represented this way and that the representations are minimal can be proved by a simple induction based on the 2-clause CNF representation of the functions.

If the OR is chosen as the concept to be taught then a teaching sequence of length $2^n - n - 1$ will be required (cf. [GK91, Lemma 1]). □

Since the above argument is combinatorial in nature, Fact 8 holds even if the learner is also allowed to be computationally unbounded.

## 5 DISCUSSION

In the exact identification model, the learner must be able to succeed even when given an adversarial teacher who chooses the least helpful counterexamples. In our teaching model, the learner can assume that the teacher will send the best counterexamples that can be generated in polynomial time. So negative results in the exact identification model that rely on the adversarial nature of the teacher [Ang90] do not immediately carry over to our teaching model.

Thus, it seems possible that teaching with trusted information could be a more powerful model. However, this is offset by the difference in stopping criteria used by the learner in each model. In exact identification, the learner stops when the teacher says to stop; in teaching with trusted information, the learner stops when convinced that there is only one concept consistent with the information received from the teacher. A teaching model like ours is too powerful if the teacher can explicitly say when to stop: the teacher and learner simply agree on an encoding protocol, the teacher sends examples

Table 1: Summary of Exact Identification vs. Teaching

| Model: | Membership Queries | Poly-time MAT | Unbounded MAT |
|---|---|---|---|
| Poly-time Teachable | properly contains | incomparable | incomparable |
| Poly-time + trusted | properly contains | not subset of | incomparable |
| Unbounded + trusted | properly contains | properly contains | properly contains |

which encode for the representation, and the learner is told to stop after seeing enough bits to reconstruct the concept. This is another version of the "cheating" problem discussed in section 3.1.

A more subtle difference between the models has to do with the clear separation of the two types of information transmitted by the teacher with trusted information as opposed to the intermingling of membership and equivalence query responses from a minimally adequate teacher. This separation in the teaching model was useful, for example, in the proof of Theorem 2. It may also be helpful to first think in teaching terms when attempting to develop a new exact identification algorithm for a class.

# 6 CONCLUSIONS AND FURTHER RESEARCH

We have introduced a model of what it means for a concept class to be teachable in a computationally feasible way: informally, there must exist a polynomial-time teacher/learner pair such that the learner can always infer the correct concept from labeled examples and possibly a small amount of trusted information presented by the teacher, and the learner can never be fooled by an adversarial "teacher."

Known relationships between variations on our teaching model and versions of exact identification learners are summarized in Table 1. The table entries describe the relationship of the sets of classes which are teachable to those which can be exactly identified in each model modulo complexity assumptions for some entries. In addition to relaxing the complexity assumptions, the main open question is whether or not exact identification with a polynomial-time minimally adequate teacher implies polynomial-time teachability.

Perhaps one of the most intriguing aspects of this line of research is that there do not seem to be any natural classes which are obviously teachable with trusted information but not exactly identifiable with membership and equivalence queries. Intuitively, of course, teaching should enable us to "do more" than learning alone can do. Teaching certainly will in general speed up the learning process, at least by constant factors; does it do more than this? The negative results for exact identification mentioned above may be a fruitful field for finding natural classes which separate teaching from identification. This model also provides a framework within which time bounds for teaching various classes can be proved.

The fact that some classes are teachable but not PAC-learnable (under the assumption of one-way functions) raises the question of whether or not PAC-learnability of a concept class having a polynomial-time minimally adequate teacher implies polynomial-time teachability.

Some other research directions include considering concept classes over continuous domains and examining the effects of incorporating randomization at various points in the model. An analog model for approximate teaching would also be of interest.

**ACKNOWLEDGEMENTS**

Discussions with Avrim Blum have been fundamental in formulating these ideas. Thanks also to Merrick Furst for his comments and support, and to several anonymous referees for many helpful comments. We gratefully acknowledge the generous support of AT&T, Matsushita Information Technology Laboratory, and the CMU School of Computer Science.

## References

[AHK89] Dana Angluin, Lisa Hellerstein, and Marek Karpinski. Learning read-once formulas with queries. Technical Report 89/528, University of California at Berkeley, 1989.

[Ang87] Dana Angluin. Learning regular sets from queries and counterexamples. *Information and Computation*, 75:87–106, 1987.

[Ang88] Dana Angluin. Queries and concept learning. *Machine Learning*, 2:319–342, 1988.

[Ang90] Dana Angluin. Negative results for equivalence queries. *Machine Learning*, 5:121–150, 1990.

[Blu90] Avrim Blum. Separating distribution-free and mistake-bound learning models over the Boolean domain. In *31st Annual Symposium on Foundations of Computer Science*, pages 211–218, 1990.

[CVS88] John C. Cherniavsky, Mahendren Velauthapillai, and Richard Statman. Inductive inference: An abstract approach. In *Proceedings of the Twenty-Eighth Annual Symposium on Foundations of Computer Science*, pages 251–266, 1988.

[GK91] Sally A. Goldman and Michael J. Kearns. On the complexity of teaching. In *Proceedings of the Fourth Annual Workshop on Computational Learning Theory*, pages 303–314, 1991.

[GMR85] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proofs. In *Proceedings of the 17th ACM Symposium on Theory of Computing*, pages 291–304, 1985.

[GRS89] Sally A. Goldman, Ronald L. Rivest, and Robert E. Schapire. Learning binary relations and

total orders. In *Proceedings of the Twenty-Ninth Annual Symposium on Foundations of Computer Science*, pages 46–51, 1989.

[Han90] Thomas R. Hancock. Identifying $\mu$-formula decision trees with queries. In *Proceedings of the Third Annual Workshop on Computational Learning Theory*, pages 23–37, 1990.

[HH91] Thomas Hancock and Lisa Hellerstein. Learning read-once formulas over fields and extended bases. In *Proceedings of the Fourth Annual Workshop on Computational Learning Theory*, pages 326–336, 1991.

[HK91] Lisa Hellerstein and Marek Karpinski. Computational complexity of learning read-once formulas over different bases. Technical Report TR-91-014, International Computer Science Institute, 1991.

[Nat87] B. K. Natarajan. On learning Boolean functions. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, pages 296–304, 1987.

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