# Uniform Characterizations of Polynomial-query Learnabilities


Yosuke Hayashi[1], Satoshi Matsumoto[2],
Ayumi Shinohara[1], and Masayuki Takeda[1]

[1] Department of Informatics, Kyushu University 33, Fukuoka 812-8581, JAPAN
[2] Faculty of Science, Tokai University, Kanagawa 259-1291, JAPAN
{yosuke,ayumi,takeda}@i.kyushu-u.ac.jp, matumoto@ss.u-tokai.ac.jp



**Abstract.** We consider the exact learning in the query model. We deal with all types of queries introduced by Angluin: membership, equivalence, superset, subset, disjointness and exhaustiveness queries, and their weak (or restricted) versions where no counterexample is returned. For each of all possible combinations of these queries, we uniformly give complete characterizations of boolean concept classes that are learnable using a polynomial number of polynomial sized queries. Our characterizations show the equivalence between the learnability of a concept class $\mathcal{C}$ using queries and the existence of a good query for any subset $H$ of $\mathcal{C}$ which is guaranteed to reject a certain fraction of candidate concepts in $H$ regardless of the answer. As a special case for equivalence queries alone, our characterizations directly correspond to the lack of the approximate fingerprint property, which is known to be a sufficient and necessary condition for the learnability using equivalence queries.


## 1 Introduction

With the remarkable advances in computer and network technology, a large quantity of data obtained from scientific experiments is available. It is an urgent and very important problem to establish methods to discover some rules which explain such a large quantity of data. Because the data is so large, it is expected to use computers to analyze the data. One approach is to apply a machine learning system which learns concepts from examples, in order to discover rules automatically. Moreover, a successful learning algorithm using queries would give us a good strategy to make experiments within a reasonable amount of time, in order to identify underlying rules. For these purpose, we have to clarify the possibilities and limitations for computers to learn concepts from examples.

The exact learning model due to Angluin [1] is one of the most popular models in the field of learning theory. In this model, a learner is required to identify a target concept exactly using queries which give partial information about the target concept to the learner. Angluin introduced six kinds of queries, membership, equivalence, superset, subset, disjointness, and exhaustiveness. In some cases, *weak* (or *restricted*) versions of queries are often used, where no counterexample is provided to a learner.

Among these queries, membership and equivalence queries have been focused on especially, and there have been some individual approaches to show combinatorial properties in order to characterize the learnability using each of three combinations of membership and equivalence queries. For equivalence queries alone, Angluin [2] introduced a notion of *approximate fingerprint property* as a tool for proving non-learnability. Gavaldà [4] showed that the property (with a slight modification) can be used to prove the converse: if a concept class does not have an approximate fingerprint property, then the concept class is exactly learnable using a polynomial number of polynomial sized equivalence queries. (See also [3]). For membership queries alone, Goldman and Kearns [5] showed that the teaching dimension gives a lower bound for the number of membership queries required to learn. Hegedüs [6] generalized it so that the generalized teaching dimension of a concept class is polynomial if and only if the concept class is learnable using a polynomial number of membership queries alone. For the combination of membership and equivalence queries, Hellerstein et al. [7] and Hegedüs [6] independently gave an elegant combinatorial property called *polynomial certificates* as a necessary and sufficient condition for polynomial-query learning.

In this paper, we give combinatorial properties which uniformly characterize the learnability for *each of any possible combinations* of all queries introduced above. We will give the characterizations in an abstract form so that we can easily generalize it for another kind of queries, not specific to these six queries.

Our characterizations are based on the following two intuitions: The first one is that if a learner can ask a *good* question to a teacher about the unknown target concept, then the concept is easy to learn. Otherwise, it might be hard to learn. Here, we regard that a question is *good* if at least a certain fraction of concepts will be rejected, no matter how the answer is returned by the teacher. If there always exists a good question for any subset of a concept class, then the learner can use it to reduce the hypothesis space efficiently. Otherwise, that is, if there is no good question for some subset of a concept class, adversary teacher can answer maliciously so that little information will be given to the learner to identify the target concept. In fact, this was a key idea to prove that the lack of approximate fingerprint property is a necessary and sufficient condition for the learnability using equivalence queries alone [2, 4].

The second intuition is that a learner can identify any target concept *exactly* if and only if the learner can confirm that the hypothesis is absolutely correct by using queries. We introduce a notion of *specifying queries* in order to capture the intuition. When equivalence queries are available, it is a trivial task, since the learner can directly confirm whether the hypothesis is correct or not.

We apply these intuitions for each type of queries, and capture the essence of query complexity of exact learning using *each of any possible combinations* of all these queries. The technicalities of the proofs may not be quite new since they are rather straightforward extensions appeared in the literature [2, 4, 6]. However, our characterizations will be applied for any kind of queries, not restricted to the ones mentioned above. Since a query corresponds to an experiment in scientific discovery, we hope that our characterizations will lead us an efficient strategy to choose and perform experiments among a large number of possible experiments.

## 2 Preliminaries

We adopt the terminology from [7, 8]. Let $\Sigma$ be an alphabet. Then $\Sigma^*$ denotes the set of all finite length strings over $\Sigma$, and $\Sigma^n$ denotes the set of all strings over $\Sigma$ of length exactly $n$. For a string $w \in \Sigma^*$, $||w||$ denotes the length of $w$, for a set $S$, $|S|$ denotes the cardinality of $S$.

A *Representation of concepts* $\mathcal{R} = \langle \Sigma, \Delta, R, \mu \rangle$ is a 4-tuple where $\Sigma$ and $\Delta$ are finite alphabets, $R$ is a subset of $\Delta^*$, and $\mu$ is a map from $R$ to subsets of $\Sigma^*$, called *concepts*. $R$ is a set of *representations*, and $\mu$ is the map that specifies which concept is represented by a given representation. For any concept $c$, $\chi_c$ denotes the characteristic function of $c$. For any string $w$, $\chi_c(w) = 1$ if $w$ is in $c$ and $\chi_c(w) = 0$ otherwise. The *size* of a concept $c$ is $\min\{||r|| : \mu(r) = c\}$.

Throughout this paper, we assume that for any representation class $\mathcal{R}$, the following problems are computable:

1. For a given string $r \in \Delta^*$, decide if $r \in R$.
2. For a given string $w \in \Sigma^*$ and $r \in \Delta^*$, decide if $w \in \mu(r)$.

The *concept class* $\mathcal{C}$ by $\mathcal{R}$ is a set of concepts that have representations in $R$. For any positive integer $m$, $\mathcal{C}_m = \{\mu(r) : r \in R, ||r|| \leq m\}$, and $\mathcal{C} = \bigcup_{m \geq 1} \mathcal{C}_m$.

In this paper, we deal with *boolean concept classes* only. Thus let us assume that $\Sigma = \{0, 1\}$. A boolean concept $c$ is a subset of $\Sigma^n$ for any positive integer $n$. When it causes no confusion, we will use $c$ itself to denote $\chi_c$. If $\mathcal{R}$ is a *boolean representation class*, each $r \in R$ will represent a boolean formula over $n$ variables, and the concept is a set of assignments to the variables that satisfies the function. For any positive integers $m$ and $n$, let $\mathcal{C}_{m,n} = \{\mu(r) : ||r|| \leq m \text{ and } \mu(r) \subseteq \Sigma^n\}$, and $\mathcal{C}_n = \bigcup_{m \geq 1} \mathcal{C}_{m,n}$. In the sequel, we identify a concept $c \in \mathcal{C}$ with its representation $r \in \mathcal{R}$ with $\mu(r) = c$ when it is clear from the context.

We assume several oracles which give some information about a target concept $c^*$ to a learner. We may regard them as experiments to identify the target concept. In the literatures, six oracles have been introduced as follows. For each string $v \in \Sigma^n$, the *membership oracle* MEM returns "Yes" if $c^*(v) = 1$ and "No" otherwise. Moreover, for each concept $h \in \mathcal{C}$, we define *Equivalence* (EQU), *Superset* (SUP), *Subset* (SUB), *Disjointness* (DIS), *Exhaustiveness* (EXH) oracles and their *weak* versions (wEQU, wSUP, wSUB, wDIS, wEXH) as in Table 1. However, in this paper, we do not have to restrict the queries to those ones, since our characterizations would not be specific to these oracles. For a query $\sigma$ and a concept $c$, we denote by $c[\sigma]$ the set of possible answers for $c$ when asking $\sigma$. We denote by $||\sigma||$ the length of a query $\sigma$. For example, for a membership query $\sigma_1$, $c[\sigma_1]$ is {"Yes"} or {"No"}, and for an equivalence query $\sigma_2$, $c[\sigma_2]$ is {"Yes"} or the set of all counterexamples.

**Table 1.** The definitions of oracles. The first row represents the types of oracles. The second row represents conditions when each oracle returns "Yes", and the third row "No". The last row shows the condition which a counterexample should satisfy. For instance, the weak equivalence oracle wEQU answers "Yes" if $h = c^*$, and answers "No" if $h \neq c^*$. The equivalence oracle EQU answers "Yes" if $h = c^*$, and returns a counterexample $w$ with $w \in (h \cup c^*) - (h \cap c^*)$ if $h \neq c^*$.

| | Yes | No | $w$ |
|---|---|---|---|
| MEM | $c^*(v) = 1$ | $c^*(v) = 0$ | |
| EQU, wEQU | $h = c^*$ | $h \neq c^*$ | $w \in (h \cup c^*) - (h \cap c^*)$ |
| SUP, wSUP | $h \supseteq c^*$ | $h \not\supseteq c^*$ | $w \in c^* - h$ |
| SUB, wSUB | $h \subseteq c^*$ | $h \not\subseteq c^*$ | $w \in h - c^*$ |
| DIS, wDIS | $h \cap c^* = \emptyset$ | $h \cap c^* \neq \emptyset$ | $w \in h \cap c^*$ |
| EXH, wEXH | $h \cup c^* = \Sigma^n$ | $h \cup c^* \neq \Sigma^n$ | $w \in \Sigma^n - (h \cup c^*)$ |

The *query complexity* of learning algorithm $\mathcal{A}$ is the sum of the lengths of queries and counterexamples returned by oracles. Note that the length of a counterexample is always $n$, since we consider only boolean concepts.

**Definition 1.** *Let $\mathcal{Q}$ be a set of queries. A concept class $\mathcal{C}$ is* polynomial-query learnable using $\mathcal{Q}$ *if there exists an algorithm $\mathcal{A}$ and a polynomial $p(\cdot,\cdot)$ such that, for any positive integers $m$, $n$ and an unknown target concept $c^* \in \mathcal{C}_{m,n}$:*

1. *$\mathcal{A}$ gets $n$ as input.*
2. *$\mathcal{A}$ may ask queries in $\mathcal{Q}$.*
3. *$\mathcal{A}$ eventually halts and outputs $r \in R$ with $\mu(r) = c^*$.*
4. *The total query complexity of $\mathcal{A}$ is at most $p(m,n)$.*

In Section 3, we consider the case where the size $m$ of a target concept is additionally given to a learner.

## 3 Good Queries

We introduce a notion of good queries in order to characterize polynomial-query learnability where the size of a target concept is known to a learner. Intuitively, a query is good for a set $\mathcal{T}$ of concepts, if a certain fraction of $\mathcal{T}$ are eliminated by the query no matter how the answer is returned.

**Definition 2.** *For a concept class $\mathcal{T}$, a query $\sigma$ and its answer $\alpha$, we define $Cons(\mathcal{T},\sigma,\alpha)$ be the set of concept in $\mathcal{T}$ that is consistent with $\sigma$ and $\alpha$. That is,*

$$Cons(\mathcal{T},\sigma,\alpha) = \{h \in \mathcal{T} \mid \alpha \in h[\sigma]\}.$$

**Definition 3.** *A query $\sigma$ is $\delta$-*good *for a concept class $\mathcal{T}$ if for any answer $\alpha$,*

$$|Cons(\mathcal{T},\sigma,\alpha)| \leq (1-\delta)|\mathcal{T}|.$$

**Algorithm** LEARNER1($m, n$ : positive integers)
**Given** $\mathcal{Q}$ : available queries
**begin**
  $H := \mathcal{C}_{m,n}$;
  **while** $|H| \geq 2$ **do**
    Find a query $\sigma \in \mathcal{Q}$ that is $1/q(m,n)$-good for $H$;    $(*)$
    Let $\alpha$ is the answer to the query $\sigma$;
    $H := Cons(H, \sigma, \alpha)$
  **endwhile**;
  **if** $|H| = 1$ **then** output the unique hypothesis $h$ in $H$
  **else** output "Target concept is not in $\mathcal{C}_{m,n}$"
**end.**

**Fig. 1.** Algorithm LEARNER1, where the size $m$ of a target concept is known.

**Theorem 1.** *Assume the size $m$ of the target concept is known to a learner. A concept class $\mathcal{C}$ is polynomial-query learnable using $\mathcal{Q}$ if and only if there exist polynomials $q(\cdot,\cdot)$ and $p(\cdot,\cdot)$ such that for any positive integers $m$, $n$ and any $\mathcal{T} \subseteq \mathcal{C}_{m,n}$ with $|\mathcal{T}| \geq 2$, there exists a query $\sigma$ in $\mathcal{Q}$ with $||\sigma|| \leq p(m,n)$ that is $1/q(m,n)$-good for $\mathcal{T}$.*

*Proof.* (if part) Let $p(\cdot,\cdot)$ and $q(\cdot,\cdot)$ be polynomials such that for any positive integers $m$,$n$ and any $\mathcal{T} \subseteq \mathcal{C}_{m,n}$ with $|\mathcal{T}| \geq 2$, there exists a query $\sigma$ that is $1/q(m,n)$-good for $\mathcal{T}$. We show a learning algorithm using queries in $\mathcal{Q}$ in Figure 1. It is not hard to verify that all procedures in the algorithm, such as *Cons*, are computable, since we only deal with boolean concepts.

First we show the correctness of the algorithm. Since $H$ is initialized as $\mathcal{C}_{m,n}$, and the target concept $c^*$ is assumed to be in $\mathcal{C}_{m,n}$, $H$ contains $c^*$ before the first stage. Since $c^*$ is consistent with any answer returned by the oracles in $\mathcal{Q}$, and at any stage $H$ is updated so that only inconsistent concepts are eliminated from $H$, $c^*$ is never eliminated. Moreover, whenever $|H| \geq 2$, we can find a query that is $1/q(m,n)$-good for $H$ in $\mathcal{Q}$. Thus the output of the algorithm is guaranteed to be exactly equal to the target concept $c^*$.

We now show that the total number of queries is $O(m \cdot p(m,n))$. We denote the set $H$ at $i$-th stage of the algorithm by $H_i$, and $l$ be the number of the stages. We can show that for any stage $i = 1, 2, ..., l-1$,

$$|H_i| \leq \left(1 - \frac{1}{p(m,n)}\right) \cdot |H_{i-1}|,$$

regardless of the answer from an oracle in $\mathcal{Q}$. Since $H_0$ is initialized as $\mathcal{C}_{m,n}$, we have

$$|H_i| \leq \left(1 - \frac{1}{p(m,n)}\right)^i \cdot |\mathcal{C}_{m,n}|$$

for any $i$. We can show that the right part becomes at most one if $i > p(m,n) \cdot \ln |\mathcal{C}_{m,n}|$ by simple calculations, which ensures the termination of the algorithm. Recall that $|\mathcal{C}_{m,n}| \leq (|\Delta| + 1)^{m+1}$ for any $m$ and $n$, since any concept in $\mathcal{C}_{m,n}$ is represented by a string over $\Delta$ of length at most $m$. Since at each stage, exactly one query is asked to an oracle, the total number of query is $O(m \cdot p(m,n))$. Since the length of each query is at most $p(m,n)$ and the length of each counterexample returned by oracles is $n$, the query complexity of the algorithm is $O(m \cdot p(m,n)(p(m,n) + n))$, which is a polynomial with respect to $m$ and $n$.

(only if part) Assume that for any polynomials $p(\cdot,\cdot)$ and $q(\cdot,\cdot)$, there exist positive integers $m, n$ and a set $\mathcal{T} \subseteq \mathcal{C}_{m,n}$ with $|\mathcal{T}| \geq 2$ such that there exists no query $\sigma$ that is $1/q(m,n)$-good for $\mathcal{T}$. Suppose to the contrary that there exists a learning algorithm $\mathcal{A}$ that exactly identifies any target concept using queries in $\mathcal{Q}$, whose query complexity is bounded by a polynomial $p'(m,n)$ for any $m$ and $n$. Let $p(m,n) = p'(m,n)$ and $q(m,n) = 2p'(m,n)$.

We construct an adversary teacher who answers for each query $\sigma$ in $\mathcal{Q}$ as follows: If $||\sigma|| > p(m,n)$, the teacher may answer arbitrarily, say "Yes". (Since the query complexity of $\mathcal{A}$ is bounded by $p'(m,n) = p(m,n)$, actually $\mathcal{A}$ can never ask such a query.) If $||\sigma|| \leq p(m,n)$, the teacher answers $\alpha$ such that $|Cons(\mathcal{T},\sigma,\alpha)| > \left(1 - \frac{1}{q(m,n)}\right)|\mathcal{T}|$. By the assumption, there always exists such a *malicious* answer. The important point is that for any query $\sigma$, its answer $\alpha$ returned by the teacher contradicts less than $1/q(m,n)$ fraction of concepts in $\mathcal{T}$. That is,

$$|\mathcal{T}| - |Cons(\mathcal{T},\sigma,\alpha)| < \frac{1}{q(m,n)}|\mathcal{T}|.$$

Since the query complexity of $\mathcal{A}$ is $p'(m,n)$, at most $p'(m,n)$ queries can be asked to the teacher.

Thus the learner can eliminate less than $(p'(m,n)/q(m,n))|\mathcal{T}|$ concepts after $p'(m,n)$ queries. Since $q(m,n) = 2p'(m,n)$, more than $(1/2)|\mathcal{T}|$ concepts in $\mathcal{T}$ are consistent with all the answers. Moreover, since $|\mathcal{T}| \geq 2$, at least two distinct concepts from $\mathcal{T}$ are consistent with all the answers so far. Since $\mathcal{A}$ is deterministic, the output of $\mathcal{A}$ will be incorrect for at least one concept in $\mathcal{T}$, which is a contradiction. □

## 4 Specifying Queries

This section deals with the case where the size $m$ of a target concept is unknown to a learner. The standard trick to overcome this problem is to guess $m$ incrementally and try to learn: initially let $m = 1$, and if there is no concept in $\mathcal{C}_{m,n}$ that is consistent with the answers given by oracles, we double $m$ and repeat. For some cases, such that the equivalence query is available, or both subset and superset queries are available, we can apply the trick correctly, since the learner can confirm the hypothesis is correct or not by asking these queries. The next definition is an abstraction of the notion.

**Algorithm** LEARNER2 ($n$ : positive integer)
**Given** $\mathcal{Q}$ : available queries
**begin**
  $m = 1$;
  **repeat**
    simulate LEARNER1$(m, n)$ using $\mathcal{Q}$;
    **if** LEARNER1 outputs a hypothesis $h$ **then**
      Let $Q$ be a set of specifying queries for $h$ in $\mathcal{C}_n$;
      **if** $h$ is consistent with the answers for all queries in $Q$ **then**
        **output** $h$ and **terminate**
    $m = m * 2$
  **forever**
**end.**

**Fig. 2.** Algorithm LEARNER2, where the target size $m$ is unknown

**Definition 4.** *A set $Q$ of queries is called* specifying queries *for a concept $c$ in $\mathcal{T}$ if the set of consistent concept in $\mathcal{T}$ is a singleton of $c$ for any answer. That is,*

$$\{h \in \mathcal{T} \mid h[\sigma] = c[\sigma] \text{ for all } \sigma \in Q\} = \{c\}.$$

For instance, if the equivalence oracle is available, the set $\{\text{EQU}(c)\}$ is a trivial specifying queries for any $c$ in $\mathcal{C}$. Moreover, if both the superset and subset oracles are available, the $\{\text{wSUP}(c), \text{wSUB}(c)\}$ is also specifying queries for any $c$ in $\mathcal{C}$. If the only membership oracle is available, our notion corresponds to the notion of *specifying set* [6].

**Theorem 2.** *A concept class $\mathcal{C}$ is polynomial-query learnable using $\mathcal{Q}$ if and only if there exist polynomials $q(\cdot,\cdot)$, $p(\cdot,\cdot)$ and $r(\cdot,\cdot)$ such that for any positive integers $m$ and $n$, the following two conditions hold:*

*(1) for any $\mathcal{T} \subseteq \mathcal{C}_{m,n}$ with $|\mathcal{T}| \geq 2$, there exists a query $\sigma$ in $\mathcal{Q}$ with $||\sigma|| \leq p(m,n)$ that is $1/q(m,n)$-good for $\mathcal{T}$.*
*(2) for any concept $c \in C_n$, there exist specifying queries $Q \subseteq \mathcal{Q}$ for $c$ in $C_n$ such that $||Q|| \leq r(m,n)$.*

*Proof.* (if part) We show a learning algorithm LEARNER2 in Fig. 2, assuming that the two conditions hold. The condition (2) guarantees that the output of LEARNER2 is exactly equal to the target concept, while the condition (1) assures that LEARNER1 will return a correct hypothesis as soon as $m$ becomes greater than or equal to the size of the target concept.

(only if part) Assume that the concept class $\mathcal{C}$ is polynomial-query learnable by a learning algorithm $\mathcal{A}$ using queries in $\mathcal{Q}$. We have only to show the condition (2), since Theorem 1 implies the condition (1). Let $n > 0$ and $c \in \mathcal{C}_n$ be arbitrarily fixed. Let $Q$ be the set of queries asked by $\mathcal{A}$ when the target concept is $c$. We can verify that $Q$ is specifying queries, since the output of $\mathcal{A}$ is always equal to the target concept $c$. Since the size of $Q$ is bounded by a polynomial, the condition holds. □

Let us notice that the above theorem uniformly gives complete characterizations of boolean concept classes that are polynomial-query learnable for each of all possible combinations of the queries such as membership, equivalence, superset, subset, disjointness and exhaustiveness queries, and their weak versions.

Moreover, as a special case, we get the characterization of learning using equivalence queries alone in terms of the approximate fingerprint property. We say that a concept class $\mathcal{C}$ has an *approximate fingerprint property* if for any polynomials $p(\cdot,\cdot)$ and $q(\cdot,\cdot)$, there exist positive integers $m, n$ and a set $\mathcal{T} \subseteq \mathcal{C}_{m,n}$ with $|\mathcal{T}| \geq 2$ such that for any concept $h \in \mathcal{C}_{p(m,n),n}$, we have $|\{c \in \mathcal{T} \mid h(w) = c(w)\}| < \frac{1}{q(m,n)}|\mathcal{T}|$ for some $w \in \Sigma^n$. Since equivalence queries contain a trivial single specifying query for each concept, we get the following result.

**Corollary 1 ([2, 4]).** *A concept class $\mathcal{C}$ is polynomial-query learnable using equivalence queries if and only if $\mathcal{C}$ does not have an approximate fingerprint property.*

## 5 Conclusion

We have shown uniform characterizations of the polynomial-query learnabilities using each of any combinations of all queries, such as membership, equivalence, superset queries, etc. Our results reveal that the polynomial-query learnability using a set of oracles is equivalent to the existence of a good query to the oracles which eliminate a certain fraction of any hypothesis space. This is quite intuitive.

In this paper, we only dealt with boolean concepts. We will generalize our results to treat general concepts in future works. Moreover, it is also interesting to investigate the computational complexity of the learning task for honest concept classes with polynomial query-complexity, in the similar way as shown by Köbler and Lindner [8], where they showed that $\Sigma_2^p$ oracles are sufficient for the learning using each of three possible combinations of membership and equivalence queries.

## References

1. D. Angluin. Queries and concept learning. *Machine Learning*, 2:319–342, 1988.
2. D. Angluin. Negative results for equivarence queries. *Machine Learning*, 5:121–150, 1990.
3. N. Bshouty, R. Cleve, R. Gavalda, S. Kannan, and C. Tamon. Oracles and queries that sufficient for exact learning. *Journal of Computer and System Sciences*, 52:421–433, 1996.
4. R. Gavaldà. On the power of equivarence queries. In *Proceedings of the 1st European Conference on Computational Learning Theory*, pages 193–203, 1993.

5. S. Goldman and M. Kearns. On the complexity of teaching. *Journal of Computer and System Sciences*, 50:20–31, 1995.
6. T. Hegedüs. Generalized teaching dimensions and the query complexity of learning. In *Proceedings of the 8th Annual Conference on Computational Learning Theory*, pages 108–117, 1995.
7. L. Hellerstein, K. Pillaipakkamnatt, V. Raghavan, and D. Wilkins. How many queries are needed to learn? *Journal of the ACM*, 43(5), 1996.
8. J. Köbler and W. Lindner. Oracles in $\Sigma_2^p$ are sufficient for exact learning. In *Proceedings of 8th Workshop on Algorithmic Learning Theory*, pages 277–290. Springer-Verlag, 1997.