From Equivalence Queries to PAC Learning: The Case of Implication Theories

Ramil Yarullin[a,b,*], Sergei Obiedkov[a]

[a] *National Research University Higher School of Economics, Pokrovskii bulvar 11, Moscow, Russia*
[b] *Yandex.Taxi, 82/2 Sadovnicheskaya Ulitsa, Moscow, Russia*

Abstract

In Angluin's exact-learning framework, equivalence queries can be simulated by stochastic equivalence testing to achieve a probably approximately correct identification of an unknown concept. We present an analysis of the number of samples that need to be generated in the process leading to a theoretical improvement on an earlier approach. We apply this modification to a previously known probably approximately correct algorithm for computing implication bases with an implication oracle and evaluate its performance in terms of the number of queries to the oracle on artificial and real-world data.

Keywords: Attribute Exploration, Concept Learning, Formal Concept Analysis, Implications, PAC Learning, Query Learning

1. Introduction

An important computational problem in formal concept analysis (FCA) [1] is finding a representation of the implication theory of a given formal context, i.e., the set of attribute implications valid in the context. Solving this problem allows extracting valuable knowledge from data by discovering hidden dependencies between attributes. Since the number of all valid implications over an attribute set M can be as large as $4^{|M|}$, it is reasonable to compute an implication basis,

*Corresponding author
Email addresses: ramly@ya.ru (Ramil Yarullin), sergei.obj@gmail.com (Sergei Obiedkov)

i.e., a set of valid implications from which all other valid implications follow. It is still a hard problem: in general, even the size of a minimum basis can be exponential in the context size [2, 3]. Furthermore, currently known algorithms for computing minimum implication bases take time exponential in the size of the output in the worst case.

One possible approach is to resort to the approximate computation of implication bases. In probably approximately correct (PAC) learning introduced by Valiant, the goal is to learn a good approximation of a target concept with high probability [4]. As a related practical example, there have been research efforts in PAC learning applied to rule discovery and association-rule mining [5, 6]. PAC learning of implication bases has been previously considered in FCA, as well as in similar settings such as learning Horn formulas [2, 7, 8, 9]. In particular, Kautz *et al.* present a PAC algorithm for computing a Horn formula from its so-called characteristic models, a problem equivalent to that of computing an implication basis of a formal context [2]. The algorithm is based on results obtained in the framework of learning with queries [10, 11]. In this framework, it is assumed that there is no direct access to data (the formal context, in our case), but it is possible to interact with a fixed set of oracles, or teachers, or domain experts, capable of answering specific types of queries.

The most widely used setting in this framework is that of a *minimally adequate teacher* capable of answering so-called membership and equivalence queries [12]. Equivalence queries are used to check if a learner's hypothesis is equivalent to the target concept; if not, the teacher must provide a counterexample that satisfies the hypothesis but does not fall under the target concept, or vice versa. Whether an example satisfies the target concept can be checked with a membership query. A practically relevant alternative to equivalence queries is stochastic equivalence testing, which makes use of a sampling oracle: a call to such an oracle returns an example from the domain with an indication of whether it falls under the target concept. Provided that a certain number of examples are considered in simulation of each equivalence query, stochastic testing makes it possible to achieve a PAC identification of the target concept. Accord-

ing to the procedure outlined in [10], the number of examples needed to simulate the ith equivalence query issued by the original algorithm grows linearly with i. Our main contribution is a tighter upper bound: we describe a procedure where this dependence is only logarithmic.

This tighter upper bound is applicable to any algorithm that uses equivalence queries. We apply it to the algorithm for learning a Horn formula with membership and equivalence queries from [11]. This algorithm runs in time polynomial in the number of variables and the size of the formula. Its PAC version that uses implication queries, i.e., queries to check the validity of a given implication, instead of membership queries is presented in [9] as a PAC version of attribute exploration, a well-known knowledge-acquisition procedure from formal concept analysis [13]. In attribute exploration, one has access to the underlying formal context $\mathbb{K}$ only through implication queries; the goal is to learn the implication theory of $\mathbb{K}$. The algorithm in [9] uses the stochastic equivalence testing–procedure from [10] to eliminate equivalence queries. Replacing this with our improved procedure results in an asymptotically smaller number of samples that need to be generated during the execution of the algorithm.

The paper is structured as follows. In Section 2, we introduce the necessary definitions and state the problem formally. In particular, in Section 2.1, we recall the basics of formal concept analysis and implication learning, including the notion of approximate implication bases. This is followed by a brief introduction to concept learning with queries and probably approximately correct learning in Section 2.2. We then state the problem of approximate learning of implications with queries precisely and survey the results from previous works in Section 3. In Section 4, we recall a technique for transforming an exact-learning algorithm with equivalence queries into a PAC algorithm that does not use equivalence queries and suggest improvements that make the resulting algorithm asymptotically more efficient. We transfer the results from the previous section to learning implications with queries in Section 5 and present some empirical evaluation of the original and modified techniques in Section 6.

2. Preliminaries

2.1. Formal Contexts and Implications

Let G be a set of *objects*, M be a finite set of *attributes*, and $I \subseteq G \times M$ be a binary relation called the *incidence relation* that indicates which objects possess which attributes. A *formal context* is a triple $\mathbb{K} := (G, M, I)$. We say that an object $g \in G$ has an attribute $m \in M$ if $(g, m) \in I$. Finite formal contexts are usually represented by cross-tables with object names as row headers and attribute names as column headers; see Figure 1 for an example.

For a set of objects $A \subseteq G$ and a set of attributes $B \subseteq M$, we define the *derivation operators* $(.)'$ in $\mathbb{K}$ as follows:

$$A' = \{m \in M \mid \forall g \in A \colon (g, m) \in I\}$$

$$B' = \{g \in G \mid \forall m \in B \colon (g, m) \in I\}$$

The consecutive application of the two derivation operators define two *closure operators* on objects and attributes:

$$A \mapsto A''$$

$$B \mapsto B''$$

Sets A'' and B'' are called the *closures* of, respectively, A and B in $\mathbb{K}$.

A pair (A, B) where $A \subseteq G$ and $B \subseteq M$ is called a *formal concept* of $\mathbb{K}$ if $A' = B$ and $B' = A$. In this case, for the *extent* A and the *intent* B of the formal concept (A, B), the following holds: $A = A''$ and $B = B''$, and A and B are said to be *closed*. The set $\operatorname{Int}\mathbb{K} \subseteq 2^M$ of all subsets $A \subseteq M$ such that $A = A''$ is called the *set of intents* of $\mathbb{K}$.

For $X \subseteq M$ and $Y \subseteq M$, we define an *implication* over M with *premise* X and *conclusion* Y as the expression $X \to Y$. We denote the set of all implications over M with $\operatorname{Imp} M := \{X \to Y \mid X \subseteq M, Y \subseteq M\}$.

A set $B \subseteq M$ *does not respect* (or *refutes*) the implication $X \to Y$ if $X \subseteq B$ and $Y \nsubseteq B$. Otherwise, i.e., if either $X \nsubseteq B$ or $Y \subseteq B$, set B is said to *respect* $X \to Y$ and to be a *model* of $X \to Y$.

Implications are closely related to propositional Horn clauses: a Horn clause is a disjunction of literals at most one of which is without negation, and every implication can be represented as a conjunction of Horn clauses with the same negated variables. More precisely, an implication $A \to B$ corresponds to the following *Horn formula*:

$$\bigwedge_{b \in B} (b \vee \bigvee_{a \in A} \neg a)$$

in the sense that, by setting the variables of a model of $A \to B$ to 1 and all other variables from M to 0, we obtain a satisfying assignment for the Horn formula, and all its satisfying assignments are obtained in this way from the models of $A \to B$.

A set $B \subseteq M$ *respects* (or is a *model* of) the set of implications $\mathcal{L} \subseteq \operatorname{Imp} M$ if it respects all the implications from $\mathcal{L}$. The set of all models of $\mathcal{L}$ is denoted by $\operatorname{Mod} \mathcal{L}$. A set of implications $\mathcal{L}$ over M defines a closure operator on M mapping $B \subseteq M$ to the unique inclusion-minimal model $A \in \operatorname{Mod} \mathcal{L}$ such that $B \subseteq A$, which is called the *closure* of B under $\mathcal{L}$ and is denoted by $\mathcal{L}(B)$.

The implication $X \to Y \in \operatorname{Imp} M$ is *valid* in a context $\mathbb{K} = (G, M, I)$ if, for any $g \in G$, its *object intent* $\{g\}'$ respects $X \to Y$. It can be shown that $X \to Y$ is valid in $\mathbb{K}$ iff $Y \subseteq X''$ or, equivalently, $X' \subseteq Y'$ [1]. A set $\operatorname{Th} \mathbb{K}$ of all implications valid in $\mathbb{K}$ is called the *theory* of $\mathbb{K}$.

Let $\mathcal{L} \subseteq \operatorname{Imp} M$ be a set of implications and $\mathbb{K} = (G, M, I)$ be a formal context. We say that $\mathcal{L}$ is *valid* in $\mathbb{K}$ if $\mathcal{L} \subseteq \operatorname{Th} \mathbb{K}$.

An implication $X \to Y \in \operatorname{Imp} M$ *follows* from $\mathcal{L}$ if every subset $B \subseteq M$ that respects $\mathcal{L}$, respects $X \to Y$ as well. In this case, we write $\mathcal{L} \models X \to Y$.

Two implication sets $\mathcal{L}_1$ and $\mathcal{L}_2$ are called *equivalent* if every implication of $\mathcal{L}_2$ follows from $\mathcal{L}_1$ and vice versa, or, equivalently, if $\operatorname{Mod} \mathcal{L}_1 = \operatorname{Mod} \mathcal{L}_2$.

If all implications following from $\mathcal{L}$ are contained in $\mathcal{L}$, then $\mathcal{L}$ is called *closed*. If every implication valid in $\mathbb{K}$ follows from $\mathcal{L}$, then $\mathcal{L}$ is *complete* for $\mathbb{K}$.

An implication set $\mathcal{L}$ is called an *implication basis* of $\mathbb{K}$ if it is complete for $\mathbb{K}$ and valid in $\mathbb{K}$. In other words, $\mathcal{L}$ is a basis of $\mathbb{K}$ if $\mathcal{L}$ is equivalent to $\operatorname{Th} \mathbb{K}$. A basis $\mathcal{L}$ of $\mathbb{K}$ is *non-redundant* if there is no $\mathcal{L}' \subset \mathcal{L}$ such that $\mathcal{L}'$ is also a

basis. Sometimes, non-redundancy is considered to be a part of the definition of an implication basis. A basis $\mathcal{L}$ of $\mathbb{K}$ is *minimum* if there is no basis $\mathcal{L}'$ such that $|\mathcal{L}'| < |\mathcal{L}|$.

We say that a subset $P \subseteq M$ is a *pseudo-intent* of $\mathbb{K}$ if $P \neq P''$ and $Q'' \subseteq P$ for any pseudo-intent $Q \subset P$. The *canonical* (or *Duquenne–Guigues*) basis $\mathrm{Can}(\mathbb{K})$ of $\mathbb{K}$ is defined as a set of all implications of the form $P \to P''$ where P is a pseudo-intent of $\mathbb{K}$ [14].

2.2. Queries and Concept Learning

Our goal in this paper is to provide algorithms that efficiently learn a Horn approximation of a formal context in a setting when the context is not given explicitly but is accessible only through queries of certain types. We introduce the general setting of learning with queries [10] in this subsection and use it to formally state our problem in the following subsection.

Let U be a *universal set* of objects. A subset $C \subseteq U$ is called a *concept* over U. A *concept class* over U is a set of concepts over U.

The problem of concept learning can be formulated as follows: we need to identify an unknown concept $C \subseteq U$ from a finite or countable set of concepts $\mathcal{H}$ called the *set of hypotheses* [10]. We assume that U is finite and $\mathcal{H} = 2^U$. Besides, we have access to a number of oracles that can answer certain types of queries. These may include, in particular, the following oracles for a concept C:

- *Equivalence oracle.* The input is a hypothesis $H \subseteq U$. The output is `True` if $H = C$. Otherwise, the output is a *counterexample* $x \in H \triangle C$, where $\triangle$ denotes the symmetric difference. A counterexample $x \in U$ is called *positive* if $x \in C \setminus H$ and *negative* if $x \in H \setminus C$.

- *Membership oracle.* The input is an object $x \in U$. The output is `True` if $x \in C$. Otherwise, the output is `False`.

- *Sampling oracle* EX. Given an unknown distribution $\mathcal{D}$ on U, the oracle EX takes no input and returns an object $x \in U$ generated according to the distribution $\mathcal{D}$ with an indication of whether x belongs to C.

A learning algorithm having access to certain oracles for a concept C *exactly identifies* C if it always halts and outputs C. A concept class $\mathcal{C}$ is *exactly learnable* with certain oracles if there is an algorithm that, when given access to such oracles for an arbitrary concept $C \in \mathcal{C}$, exactly identifies C.

Besides exact identification, probably approximately correct (PAC) identification is considered. The criterion of PAC identification, first introduced by [4], can be summarised as follows.

Let $\mathcal{D}$ be a probability distribution on the universal set U and let $\mathcal{C} \subseteq 2^U$ be a concept class. We define the *distance* between two concepts $C_1, C_2 \in \mathcal{C}$ as the probability that an object $x \in U$ is drawn according to the distribution $\mathcal{D}$ is in $C_1 \triangle C_2$:

$$\operatorname{dist}(C_1, C_2) = \Pr_{\mathcal{D}}(x \in C_1 \triangle C_2).$$

For $0 < \epsilon \leq 1/2$, we call $H \subseteq U$ an *ϵ-approximation* of $C \subseteq U$ if $\operatorname{dist}(H, C) \leq \epsilon$. A concept class $\mathcal{C}$ is *PAC learnable* if there is an algorithm A such that, for every $C \in \mathcal{C}$, every distribution $\mathcal{D}$ on U, and arbitrary $0 < \epsilon \leq 1/2$ and $0 < \delta \leq 1/2$, it uses EX for C and $\mathcal{D}$ to compute, with probability at least $1 - \delta$, an ϵ-approximation of C. It is usually required that the number of calls that the algorithm makes to EX is bounded by a polynomial in $1/\epsilon$ and $1/\delta$. If the algorithm performs in time polynomial in $1/\epsilon, 1/\delta$, and the size of the representation of C, we call the concept class $\mathcal{C}$ *efficiently PAC learnable*.

3. Learning Implications with Queries

3.1. Horn Approximation

We define the *Horn distance* between an implication set $\mathcal{L}$ and a context $\mathbb{K}$ as the normalized size of the symmetric difference between $\operatorname{Mod}\mathcal{L}$ and $\operatorname{Int}\mathbb{K}$:

$$\operatorname{dist}(\mathcal{L}, \mathbb{K}) := \frac{|\operatorname{Mod}\mathcal{L} \triangle \operatorname{Int}\mathbb{K}|}{2^{|M|}}.$$

The *strong Horn distance* between $\mathcal{L}$ and $\mathbb{K}$ is defined as follows:

$$\operatorname{dist}_{\text{STRONG}}(\mathcal{L}, \mathbb{K}) := \frac{|\{A \subseteq M \mid A'' \neq \mathcal{L}(A)\}|}{2^{|M|}}.$$

It reflects the difference between $\mathcal{L}$ and $\mathbb{K}$ in terms of the number of attribute subsets with different closures in $\mathbb{K}$ and under $\mathcal{L}$.

For $0 < \epsilon < 1$, we say that an implication set $\mathcal{H}$ is an *ϵ-Horn approximation* of $\mathbb{K}$ if $\mathrm{dist}(\mathcal{H}, \mathbb{K}) \leq \epsilon$ and an *ϵ-strong Horn approximation* of $\mathbb{K}$ if $\mathrm{dist}_{\text{STRONG}}(\mathcal{H}, \mathbb{K}) \leq \epsilon$ [9].

When $\operatorname{Int}\mathbb{K}$ and $\operatorname{Mod}\mathcal{H}$ are small in comparison with $2^{|M|}$, the Horn distance between them is also small no matter how different they are. In this case, $\mathcal{H} = \{\varnothing \to M\}$ is likely to be an ϵ-Horn approximation even for a rather small ϵ, because $|\operatorname{Mod}\mathcal{H}| = 1$ and the distance $\mathrm{dist}(\mathcal{H}, \mathbb{K}) \leq \dfrac{|\operatorname{Int}\mathbb{K}| + |\operatorname{Mod}\mathcal{H}|}{2^{|M|}}$ is small since $\dfrac{|\operatorname{Int}\mathbb{K}|}{2^{|M|}}$ is small.

A common practical case when $\mathbb{K}$ has a relatively small set of intents is categorical data sets transformed into binary ones using one-hot encoding (or, in FCA terms, nominally scaled many-valued contexts): a categorical feature is replaced by a number of binary attributes each corresponding to a single value of the categorical attribute. Such attributes derived from a single feature will be mutually exclusive. Consider, for example, the Zoo data set, where there are 101 objects (animals) described by 16 binary features and 2 categorical features (the number of legs and the type of the animal) [15]. After one-hot encoding, the attributes $legs = 0, legs = 2, legs = 4, legs = 5, legs = 6$, and $legs = 8$ are mutually exclusive, and so are the attributes $type = 1, type = 2, type = 3, type = 4, type = 5, type = 6$, and $type = 7$. It means that an attribute subset chosen uniformly at random is not an intent of $\mathbb{K}$ with probability at least $1 - \dfrac{6 \cdot 7}{2^{6+7}} \approx 0.995$; in other words, at most $\approx 0.5\%$ of all attribute subsets are concept intents. Hence, for ϵ-Horn approximation to be meaningful, we have to set ϵ at 0.005 at most.

The strong Horn distance is designed to alleviate this issue. Since $\mathrm{dist}(\mathcal{H}, \mathbb{K}) \leq \mathrm{dist}_{\text{STRONG}}(\mathcal{H}, \mathbb{K})$ for any $\mathcal{H}$ and $\mathbb{K}$, an ϵ-strong Horn approximation is always an ϵ-Horn approximation, but the reverse is not necessarily true.

3.2. Problem Statement: The Case of Implication Theories

Our goal is to be able to efficiently learn implication bases of formal contexts or their approximations in a setting when we are not given the context explicitly but can gain some information about it via queries. Let M be a known finite set of attributes, $\mathbb{K} = (G, M, I)$ be an unknown formal context, and $\mathcal{L}$ be its canonical basis (also unknown). To represent our problem in terms of query learning, we treat 2^M as the universal set and identify implication sets with sets of their models. Consider the following oracles:

- *Equivalence oracle.* The input is a set of implications $\mathcal{H}$ (hypothesis). The output is `True` if $\mathcal{H}$ forms an implication basis of $\mathbb{K}$. Otherwise, the output is a *counterexample* $X \in \operatorname{Mod}\mathcal{H} \triangle \operatorname{Mod}\mathcal{L} = \operatorname{Mod}\mathcal{H} \triangle \operatorname{Int}\mathbb{K}$. A counterexample X is called *positive* if $X \in \operatorname{Int}\mathbb{K} \setminus \operatorname{Mod}\mathcal{H}$ and *negative* if $X \in \operatorname{Mod}\mathcal{H} \setminus \operatorname{Int}\mathbb{K}$.

- *Membership oracle.* The input is a subset $X \subseteq M$. The output is `True` if $X \in \operatorname{Mod}\mathcal{L}$ or, equivalently, if $X = X''$. Otherwise, the output is `False`.

- *Implication oracle.* The input is an implication $A \to B \in \operatorname{Imp} M$. The output is `True` if $A \to B$ is valid in $\mathbb{K}$. Otherwise, the output is `False` (*restricted* version) or a *counterexample* $X \subseteq M$ from $\{\{g\}' \mid g \in G\}$ such that X does not respect $A \to B$ (*full* version).

An algorithm, Horn1, for learning Horn formulas with equivalence and membership queries is described in [11], where it is proved that it requires time polynomial in the number of variables, n, and the number of clauses, m, of the target Horn formula, making $O(mn)$ equivalence queries and $O(m^2n)$ membership queries in the process. This algorithm is straightforwardly translated into our setting by replacing the target Horn formula by the canonical basis of the unknown formal context $\mathbb{K}$. Interestingly, the output of the algorithm is not just any implication set equivalent to the canonical basis: it is always precisely the canonical basis [16]. Hence, if membership and equivalence queries

are available, the canonical basis can be computed in total polynomial time, that is, time polynomial in the combined size of input and output.

It is easy to show that the membership query for $X \subseteq M$ can be simulated by means of at most $|M| - |X|$ (restricted) implication queries [17, 9]. One possible procedure is described in Algorithm 1. A different procedure requiring a full version of the implication oracle is presented in [8].

Algorithm 1 Membership query simulated with implication queries

Require: A set M, its subset X, and the implication oracle *IsValid* for a context over attribute set M

```
1: function IsMember(X, IsValid, M)
2:     for m ∈ M \ X do
3:         if IsValid(X → {m}) then
4:             return False
5:     return True
```

Therefore, if equivalence and implication oracles for context $\mathbb{K}$ are available, the canonical basis of $\mathbb{K}$ can be learned in total polynomial time.

In the setting when only the implication oracle is available, formal concept analysis offers a technique called *attribute exploration* [13]. Its implementation is usually based on a modification of NextClosure, a standard algorithm for computing the canonical basis of an explicitly given formal context [18]. Unfortunately, the algorithm may require time exponential not only in the size of the context but also in the number of implications in its canonical basis. No total–polynomial time algorithm is known for the case when the context is given explicitly; designing such an algorithm is a major open question related to other important problems such as enumerating minimal transversals in a hypergraph [19, 20].

A total–polynomial time algorithm that uses only implication queries would solve this problem since implication queries can be simulated in polynomial time if the context is available, but, of course, no such algorithm is known. In this paper, we consider approximate computation of the canonical basis with implication queries, thus aiming at an efficient probably approximately correct

version of attribute exploration. More precisely, our problem is as follows:

> Given a (restricted) implication oracle for a context $\mathbb{K} = (G, M, I)$ and parameters $0 < \epsilon \leq 1$ and $0 < \delta \leq 1$, compute, with probability at least $1 - \delta$, an ϵ- or ϵ-strong Horn approximation of $\mathbb{K}$.

It is known that this problem is solvable in total polynomial time [9, 8]. The solution is based on a general technique from [10] for transforming an exact-learning algorithm using equivalence queries into a PAC algorithm without equivalence queries. The idea is to use a random-sampling strategy to search for a counterexample instead of relying on equivalence queries to provide such counterexamples. As described in [10], the number of samples required to simulate an ith equivalence query grows linearly with i. We recall this technique in Section 4, and then, using tighter analysis, we show that, in fact, the number of queries logarithmic in i is sufficient. Note that this result is not specific to learning implications; it is applicable to any algorithm with equivalence queries. Our experiments in Section 6 provide first evidence that, applied to the problem of learning implications with implication queries, an improved sampling strategy can make it possible to significantly reduce the overall number of queries.

4. Stochastic Equivalence Testing

Consider an algorithm that uses the equivalence oracle for an unknown concept C and exactly identifies C. Its simplified schema is shown in Algorithm 2.

Algorithm 2 Exact concept identification using an equivalence oracle

Require: The equivalence oracle *IsEquivalent* for an unknown concept C, ...

```
1: function Identify(IsEquivalent, ...)
2:     H ← H_0                          ▷ where H_0 is some initial hypothesis
3:     while IsEquivalent(H, ...) returns a counterexample x do
4:         Modify(H, x, ...)
5:     return H
```

A polynomial-time algorithm using equivalence queries can be transformed into a PAC algorithm for the same learning problem [10]. The idea is that, in-

stead of using an equivalence oracle to check if a hypothesis H is identical to the target concept C, we call the sampling oracle EX a number of times and check if any of the objects it returns belongs to $H \triangle C$. If so, such an object is returned as a counterexample. Otherwise, in case none of the generated samples belongs to $H \triangle C$, the distance between H and C is assumed small enough and the algorithm is terminated. This is achieved by replacing calls to the *IsEquivalent* function in Algorithm 2 with calls to the *IsApproximatelyEquivalent* function given in Algorithm 3.

Algorithm 3 Approximate-equivalence testing

Require: A hypothesis H, the sampling oracle EX for an unknown concept C, and $q_i \in \mathbb{N}$

```
1: function IsApproximatelyEquivalent(H, EX, q_i)
2:     for q_i times do
3:         (x, ℓ) ← EX()                    ▷ ℓ ∈ {True, False} indicates if x ∈ C
4:         if ℓ ≢ (x ∈ H) then
5:             return x
6:     return True
```

Angluin shows in [10] that, to obtain an ϵ-approximation of C with probability at least $1 - \delta$, it is sufficient to replace the ith equivalence query by

$$q_i = \left\lceil \frac{1}{\epsilon}(\ln \frac{1}{\delta} + i \ln 2) \right\rceil \tag{1}$$

calls to the EX oracle and terminate the algorithm if none of them returns a counterexample.

We derive this upper bound from a more general result that we prove next. We then provide a better bound.

4.1. How Many Calls to EX Are Sufficient?

The following theorem shows how to choose the number q_i of calls to EX to simulate the ith equivalence query so that the probability of obtaining an ϵ-approximation of C is at least $1 - \delta$.

Theorem 1. *Given* $0 < \epsilon \leq 1/2$ *and* $0 < \delta \leq 1/2$*, let* $\{\lambda_i(\delta)\}_{i \geq 1}$ *be an arbitrary sequence satisfying the following conditions:*

$$\begin{aligned} &\sum_{i=1}^{+\infty} \lambda_i(\delta) \leq \delta; \\ &\forall i \in \mathbb{N} \colon \lambda_i(\delta) > 0. \end{aligned} \tag{2}$$

Set

$$q_i \coloneqq \left\lceil \log_{1-\epsilon} \lambda_i(\delta) \right\rceil.$$

Suppose that there is an exact-learning algorithm using equivalence queries with respect to a target concept $C \subseteq U$*. We modify it by replacing the* i*th equivalence query by* q_i *calls to the sampling oracle* EX *for* C*. We claim that, with probability at least* $1 - \delta$*, the outcome of the modified algorithm is an* ϵ*-approximation of the target concept* C*.*

Proof. We refer to the sequence of calls to EX replacing a particular equivalence query as a *simulated equivalence query*. Denote by N the random variable representing the number of simulated equivalence queries performed by the algorithm and by H_i the hypothesis obtained after the ith simulated query. It is reasonable to assume that the exact-learning algorithm terminates as soon as the equivalence oracle confirms that the current hypothesis is equal to the target concept. Then, in our simulation, $H_{N-1} = H_N = H$, the outcome of the algorithm.

Let p_n be the probability that the algorithm terminates after at most n simulated queries and returns $H \subseteq U$ such that $\operatorname{dist}(H, C) > \epsilon$. Assuming that the original exact-learning algorithm always makes at least one equivalence query (as this may be the only way for it to check the correctness of its hypothesis), we have $p_0 = 0$ and, for $n \geq 1$,

$$\begin{aligned} p_n &= \Pr(N \leq n, \operatorname{dist}(H, C) > \epsilon) \\ &= \Pr(N \leq n-1, \operatorname{dist}(H, C) > \epsilon) + \Pr(N = n, \operatorname{dist}(H, C) > \epsilon) \\ &= p_{n-1} + \Pr(N = n, \operatorname{dist}(H, C) > \epsilon \mid N \geq n) \Pr(N \geq n). \end{aligned}$$

Note that

$$\begin{aligned}\Pr(N = n, \mathrm{dist}(H, C) > \epsilon \mid N \geq n) &= \Pr(N = n, \mathrm{dist}(H_{n-1}, C) > \epsilon \mid N \geq n) \\ &\leq \Pr(N = n \mid N \geq n, \mathrm{dist}(H_{n-1}, C) > \epsilon) \\ &=: \delta_n\end{aligned}$$

and

$$\Pr(N \geq n) = 1 - \Pr(N \leq n-1) \leq 1 - \Pr(N \leq n-1, \mathrm{dist}(H, C) > \epsilon) = 1 - p_{n-1}.$$

Thus,

$$p_n \leq p_{n-1} + \delta_n(1 - p_{n-1}).$$

We have

$$\begin{aligned}\delta_n &= \Pr(N = n \mid N \geq n, \mathrm{dist}(H_{n-1}, C) > \epsilon) \\ &= (1 - \Pr(EX() \in H_{n-1} \triangle C \mid N \geq n, \mathrm{dist}(H_{n-1}, C) > \epsilon))^{q_n} \\ &< (1 - \epsilon)^{q_n}.\end{aligned}$$

By setting $q_n := \lceil \log_{1-\epsilon} \lambda_n(\delta) \rceil$, we ensure $\delta_n < \lambda_n(\delta)$:

$$\delta_n < (1 - \epsilon)^{q_n} = (1 - \epsilon)^{\lceil \log_{1-\epsilon} \lambda_n(\delta) \rceil} \leq \lambda_n(\delta).$$

Hence,

$$p_n < p_{n-1} + \lambda_n(\delta)(1 - p_{n-1}) \leq p_{n-1} + \lambda_n(\delta) \leq \sum_{i=1}^{n} \lambda_i(\delta)$$

and

$$\Pr(\mathrm{dist}(H, C) > \epsilon) = \lim_{n \to +\infty} p_n \leq \sum_{i=1}^{+\infty} \lambda_i(\delta) \leq \delta.$$

□

It is instructive to think of $\lambda_i(\delta)$ in Theorem 1 as an upper bound on the probability of failing to find a counterexample with the ith simulated equivalence query given that there are still enough (w.r.t. ϵ) counterexamples at the time of the simulation. Proposition 1 gives a lower bound on the probability of computing an ϵ-approximation in terms of $\lambda_i(\delta)$.

Proposition 1. *In the setting of Theorem 1, the probability that the algorithm returns an ϵ-approximation of the target concept C is at least* $\prod_{i=1}^{+\infty}(1-\lambda_i(\delta))$.

Proof. The sequence of probabilities $\{p_n\}_{n\geq 1}$ from the proof of Theorem 1 is bounded by the sequence $\{x_n\}_{n\geq 1}$ defined as $x_n = x_{n-1} + \lambda_n(\delta)(1-x_{n-1})$ with $x_1 = \lambda_1(\delta)$. Thus,

$$\Pr(\text{dist}(H,C) > \epsilon) = \lim_{n\to+\infty} p_n \leq \lim_{n\to+\infty} x_n.$$

From the definition of $\{x_n\}_{n\geq 1}$:

$$\frac{1-x_n}{1-x_1} = \prod_{i=2}^{n} \frac{1-x_i}{1-x_{i-1}} = \prod_{i=2}^{n}(1-\lambda_i(\delta)).$$

Therefore, $x_n = 1 - \prod_{i=1}^{n}(1-\lambda_i(\delta))$ and $\lim_{n\to+\infty} x_n = 1 - \prod_{i=1}^{+\infty}(1-\lambda_i(\delta))$ meaning that

$$\Pr(\text{dist}(H,C) \leq \epsilon) \geq \prod_{i=1}^{+\infty}(1-\lambda_i(\delta)).$$

□

Even though this limit can hardly be calculated in general, in some particular cases, it is possible. For example, for $\lambda_n(\delta) = \frac{6\delta}{\pi^2 n^2}$ Weierstrass factorization theorem provides a closed form for the infinite product: $1 - \prod_{i=1}^{+\infty}\left(1 - \frac{6\delta}{\pi^2 n^2}\right) = 1 - \frac{\sin(\sqrt{6\delta})}{\sqrt{6\delta}}$.

An immediate consequence of Theorem 1 is that the number of calls to the EX oracle given in (1), as suggested in [10], is indeed sufficient for transforming an exact-learning algorithm with equivalence queries into a PAC-learning algorithm without them. We obtain this bound by using $\lambda_i(\delta) = \frac{\delta}{2^i}$, in which case,

$$q_i = \left\lceil \log_{1-\epsilon} \frac{\delta}{2^i} \right\rceil = \left\lceil \frac{\ln \frac{\delta}{2^i}}{\ln(1-\epsilon)} \right\rceil = \left\lceil \frac{\ln \frac{2^i}{\delta}}{-\ln(1-\epsilon)} \right\rceil \leq \left\lceil \frac{1}{\epsilon}\left(\ln 2^i + \ln\frac{1}{\delta}\right) \right\rceil$$
$$= \left\lceil \frac{1}{\epsilon}\left(\ln\frac{1}{\delta} + i\ln 2\right) \right\rceil.$$

The inequality is due to $-\ln(1-\epsilon) \geq \epsilon$.

A smaller bound is obtained when using $\lambda_i(\delta) = \dfrac{\delta}{i(i+1)}$. Note that this sequence satisfies conditions (2). In this case,

$$q_i = \left\lceil \log_{1-\epsilon} \frac{\delta}{i(i+1)} \right\rceil = \left\lceil \frac{\ln \frac{i(i+1)}{\delta}}{-\ln(1-\epsilon)} \right\rceil \leq \left\lceil \frac{1}{\epsilon} \left(\ln(i(i+1)) + \ln \frac{1}{\delta} \right) \right\rceil$$
$$= O\left(\frac{1}{\epsilon} \left(\ln \frac{1}{\delta} + \ln i \right) \right).$$

Therefore, the number of calls to EX needed to simulate the ith equivalence query depends on i only logarithmically rather than linearly, as suggested by the previously known bound (1). As the following theorem shows, using $\lambda_i(\delta) = \dfrac{\delta}{i(i+1)}$ instead of $\lambda_i(\delta) = \dfrac{\delta}{2^i}$ also leads to an asymptotically smaller bound on the total number of samples needed.

Theorem 2. *Assume the notation and conditions of Theorem 1. Suppose that the exact learning algorithm requires at most k calls to the equivalence oracle on a particular input (and, therefore, the modified algorithm makes at most k simulated queries). Then, on this input,*

1. *if $\lambda_i(\delta) = \dfrac{\delta}{2^i}$, the total number of calls to EX in the modified algorithm in the worst case is*
$$\Theta\left(\frac{k}{\epsilon} \left(k + \log \frac{1}{\delta} \right) \right);$$
2. *if $\lambda_i(\delta) = \dfrac{\delta}{i(i+1)}$, the total number of calls to EX in the modified algorithm in the worst case is*
$$\Theta\left(\frac{k}{\epsilon} \left(\log k + \log \frac{1}{\delta} \right) \right).$$

Proof. In the worst case, the modified algorithm makes k simulated queries and the ith query requires precisely q_i calls to EX with only the last call within each simulated query returning a counterexample. Therefore, the total number of calls to EX in the worst case is $\sum_{i=1}^{k} q_i$.

Define $P_k(\lambda) := \prod_{i=1}^{k} \lambda_n(\delta)$. The total number of samples generated in the

modified algorithm is bounded by

$$\sum_{i=1}^{k} q_i = \sum_{i=1}^{k} \left\lceil \log_{1-\epsilon} \lambda_i(\delta) \right\rceil \geq \sum_{i=1}^{k} \log_{1-\epsilon} \lambda_i(\delta) = \log_{1-\epsilon} \left(\prod_{i=1}^{k} \lambda_i(\delta) \right)$$
$$= \log_{1-\epsilon} P_k(\lambda) =: Q_k(\lambda).$$

From the above, it also follows that

$$Q_k(\lambda) \leq \sum_{i=1}^{k} q_i \leq Q_k(\lambda) + k$$

The lower bound can be restated as follows:

$$Q_k(\lambda) = \log_{1-\epsilon} P_k(\lambda) = \frac{1}{\ln(1-\epsilon)} \ln P_k(\lambda) \sim -\frac{1}{\epsilon} \ln P_k(\lambda)$$

We show futher in Section 4.2 that $k = o(-\ln P_k(\lambda))$. Assuming that for now, we have:

$$\sum_{i=1}^{k} q_i \sim Q_k(\lambda) \sim -\frac{1}{\epsilon} \ln P_k(\lambda)$$

Now we consider two cases:

1. If $\lambda_i(\delta) = \dfrac{\delta}{2^i}$, we have

$$P_k(\lambda) = \prod_{i=1}^{k} \lambda_i(\delta) = \prod_{i=1}^{k} \frac{\delta}{2^i} = \delta^k 2^{-\frac{k(k+1)}{2}}$$

It implies that

$$\sum_{i=1}^{k} q_i \sim \frac{1}{\epsilon} \left(\frac{k(k+1)\ln 2}{2} + k \ln \frac{1}{\delta} \right) = \Theta\left(\frac{k}{\epsilon} \left(k + \log \frac{1}{\delta} \right) \right)$$

2. If $\lambda_i(\delta) = \dfrac{\delta}{i(i+1)}$, we have

$$P_k(\lambda) = \prod_{i=1}^{k} \lambda_i(\delta) = \prod_{i=1}^{k} \frac{\delta}{i(i+1)} = \delta^k \frac{1}{(k!)^2(k+1)}$$

Then

$$\sum_{i=1}^{k} q_i \sim \frac{1}{\epsilon} \left(\ln\left((k!)^2(k+1) \right) + k \ln \frac{1}{\delta} \right) \sim \frac{1}{\epsilon} \left(2k \ln k + k \ln \frac{1}{\delta} \right)$$
$$= \Theta\left(\frac{k}{\epsilon} \left(\log k + \log \frac{1}{\delta} \right) \right)$$

□

An important question is whether we can find an optimal sequence $\{\lambda_n(\delta)\}_{n\geq 1}$ that minimizes the number of generated samples in the expected worst-case scenario, when we still make q_i queries to EX at the ith step but treat the number of stochastic equivalence queries N as a random variable. Since $\sum_{i=1}^{N} q_i \sim Q_N(\lambda)$, we can aim at minimizing the expectation of $Q_N(\lambda)$. We provide details in the following theorem.

Theorem 3. *For a given $0 < \epsilon \leq 1/2$, let N be a random variable with probability distribution* Pr *representing the number of simulated equivalence queries performed by the algorithm. Then the minimum value of $\mathbb{E}Q_N(\lambda)$ is reached for the sequence $\{\lambda_n(\delta)\}_{n\geq 1}$ defined as*

$$\lambda_n(\delta) := \frac{\delta \Pr(N \geq n)}{\mathbb{E}N}.$$

Proof.

$$\mathbb{E}Q_N(\lambda) = \sum_{n=1}^{+\infty} \Pr(N=n) Q_n(\lambda) = \sum_{n=1}^{+\infty} \left(\Pr(N=n) \log_{1-\epsilon} \prod_{i=1}^{n} \lambda_i(\delta) \right)$$

Our goal is to minimize $\mathbb{E}Q_N(\lambda)$, so we can write down the following constrained optimization problem:

$$\begin{aligned} \max_{\lambda} \quad & \sum_{n=1}^{+\infty} \left(\Pr(N=n) \ln \prod_{i=1}^{n} \lambda_i(\delta) \right) \\ \text{s.t.} \quad & \sum_{n=1}^{+\infty} \lambda_n(\delta) = \delta \\ & \forall n \in \mathbb{N}\ \forall \delta \in (0, 1/2] : \lambda_n(\delta) > 0 \end{aligned}$$

We find the solution using the method of Lagrange multipliers, subject to only the first condition. The Lagrangian function is as follows:

$$\mathcal{L}(\lambda, \mu) = \sum_{n=1}^{+\infty} \left(\Pr(N=n) \sum_{i=1}^{n} \ln \lambda_i(\delta) \right) - \mu \left(\sum_{i=1}^{+\infty} \lambda_n(\delta) - \delta \right)$$

We require that all the partial derivatives of the unconstrained problem are equal to 0:

$$\frac{\partial \mathcal{L}}{\partial \lambda_j} = \frac{1}{\lambda_j(\delta)} \sum_{n=j}^{+\infty} \Pr(N=n) - \mu = 0 \Rightarrow \lambda_j(\delta) = \frac{1}{\mu} \Pr(N \geq j)$$

Hence, the first condition can be expressed as follows:

$$\delta = \sum_{n=1}^{+\infty} \lambda_n(\delta) = \frac{1}{\mu} \sum_{n=1}^{+\infty} \Pr(N \geq n) = \frac{\mathbb{E}N}{\mu}$$

Thus,

$$\lambda_n(\delta) = \frac{\delta \Pr(N \geq n)}{\mathbb{E}N}$$

To complete the proof, we shall note that $\lambda_n(\delta) > 0$, so the obtained solution is feasible indeed.

□

4.2. Can We Do Better?

It can be proved that it is not possible to choose the sequence $\{\lambda_i(\delta)\}_{i\geq 1}$ so as to get an asymptotically better upper bound than the one obtained in the second part of Theorem 2. Indeed, by the inequality of arithmetic and geometric means, for any sequence $\{\lambda_i(\delta)\}_{i\geq 1}$ satisfying the constraints (2), we have

$$P_k(\lambda) = \prod_{i=1}^{k} \lambda_i(\delta) \leq \left(\frac{1}{k} \sum_{i=1}^{k} \lambda_i(\delta) \right)^k \leq \left(\frac{\delta}{k} \right)^k. \tag{3}$$

Using this inequality and notation from the proof of Theorem 2, we obtain

$$\sum_{i=1}^{k} q_i \geq Q_k(\lambda) = \log_{1-\epsilon} P_k(\lambda) > -\frac{1}{\epsilon} \ln P_k(\lambda) \geq \frac{1}{\epsilon} \left(k \ln k + k \ln \frac{1}{\delta} \right).$$

This lower bound for $\sum_{i=1}^{k} q_i$ has the same growth rate as the second sequence from Theorem 2, which means that the latter is optimal in that it ensures the asymptotically smallest number of sampling queries.

It is also worth noting that there is no sequence $\{\lambda_i(\delta)\}_{i\geq 1}$ such that $P_k(\lambda) = \left(\frac{\delta}{k} \right)^k$ if k is unknown in advance. Moreover, there is no sequence λ such that for any other valid sequence λ' and any large enough k, $P_k(\lambda) \geq P_k(\lambda')$. Let us provide a constructive proof.

Proposition 2. *Let Λ be the set of all sequences satisfying the constraints from (2). For two sequences $\{\lambda_n(\delta)\}_{n\geq 1} \in \Lambda$ and $\{\lambda'_n(\delta)\}_{n\geq 1} \in \Lambda$, we say that $\lambda \prec \lambda'$*

if $\exists K \in \mathbb{N}\ \forall k > K\colon P_k(\lambda) < P_k(\lambda')$. *Then, for any sequence* $\lambda \in \Lambda$, *there exists* $\lambda' \in \Lambda$ *such that* $\lambda \prec \lambda'$.

Proof. The series $\sum\limits_{i=1}^{+\infty} \lambda_i(\delta)$ converges and consists of positive numbers; so there must exist $K \in \mathbb{N}$ such that, among the first K numbers of $\{\lambda_n(\delta)\}_{n\geq 1}$, at least two are distinct. Let us construct a new sequence $\{\lambda'_n(\delta)\}_{n\geq 1}$ as follows:

$$\forall i \leq K\colon \quad \lambda'_i(\delta) := \frac{1}{K}\sum_{i=1}^{K} \lambda_i(\delta)$$

$$\forall i > K\colon \quad \lambda'_i(\delta) := \lambda_i(\delta)$$

Then, for $k > K$, we have $P_k(\lambda) < P_k(\lambda')$ since, by the inequality of arithmetic and geometric means,

$$\prod_{i=1}^{K} \lambda_i(\delta) < \left(\frac{1}{K}\sum_{i=1}^{K} \lambda_i(\delta)\right)^K.$$

Therefore, $\lambda \prec \lambda'$. □

Even though the upper bound in (3) is unattainable, it may be possible to decrease the hidden constant in the obtained estimate of $\sum\limits_{i=1}^{k} q_i$. In Theorem 2, the constant before $k \ln k$ is 2. We reduce it by considering the telescoping sequence $\lambda_n(\delta) = \delta(n^{-r} - (n+1)^{-r})$ for some $0 < r \leq 1$:

$$P_k(\lambda) = \prod_{n=1}^{k} \delta(n^{-r} - (n+1)^{-r}) = \delta^k (k!)^{-r} \prod_{n=1}^{k} \left(1 - \left(1 + \frac{1}{n}\right)^{-r}\right).$$

Here we can note that according to a generalized version of Bernoulli's inequality:

$$\left(1 + \frac{1}{n}\right)^{-r} = \left(1 - \frac{1}{n+1}\right)^{r} \leq 1 - \frac{r}{n+1}$$

Therefore,

$$P_k(\lambda) \geq \delta^k (k!)^{-r} \prod_{n=1}^{k} \frac{r}{n+1} = \delta^k (k!)^{-r-1} \frac{r^k}{k+1}$$

Finally,

$$\begin{aligned}\sum_{i=1}^{k} q_i \sim Q_k(\lambda) &\sim -\frac{1}{\epsilon}\ln P_k(\lambda)\\ &\leq -\frac{1}{\epsilon}\ln\left(\delta^k (k!)^{-r-1}\frac{r^k}{k+1}\right)\\ &\sim \frac{1}{\epsilon}\left(k\ln\frac{1}{\delta} + (r+1)k\ln k + k\ln\frac{1}{r}\right)\end{aligned}$$

Thus, we obtained that the constant before $k\ln k$ can be set arbitrarily close to 1 by selecting r close enough to 0. However, note that this lower constant before the term $k\ln k$ comes at the price of adding the term $k\ln\frac{1}{r}$, which grows with r decreasing.

5. Probably Approximately Correct Learning of Implication Bases

As explained in Section 2.1, implications are closely related to Horn clauses. An algorithm, called Horn1, capable of learning conjunctions of Horn clauses through membership and equivalence queries was proposed in [11]. It was later shown that the implication set corresponding to the output of the algorithm is the canonical basis of the underlying formal context [16]. A PAC version of this algorithm using an implication oracle instead a membership oracle is presented below as Algorithm 4 adapted from [9]. There, $UEX\langle M, IsValid\rangle$ is a function that generates a subset of M uniformly at random and checks if this subset is closed in the context underlying the implication oracle $IsValid$. Since it uses an implication oracle, Algorithm 4 can be regarded as a probably approximately correct version of attribute exploration [13].

A notation-related clarification would be in order. In line 4 of Algorithm 3, we check if $x \in H$, which should be read as 'object x satisfies hypothesis H' rather than as 'x belongs to H'. Here, H is some representation of a hypothetical concept, but it may be given in a form different from the set of objects belonging to the concept. In the case of implication learning, H is a set of implications representing a set of models, and $x \in H$ should be read as 'attribute set X is a model of the implication set H'.

Algorithm 4 PAC identification of an implication basis with implication queries

Require: An attribute set M; the implication oracle *IsValid* for a formal context over M; $0 < \epsilon < 1$ and $0 < \delta < 1$

function *HornApproximation*(M, *IsValid*, ϵ, δ)
  $\mathcal{H} \leftarrow []$
  $i \leftarrow 1$
  while *IsApproximatelyEquivalent*($\mathcal{H}$, $UEX\langle M, IsValid\rangle$, $q_i(\epsilon, \delta)$) returns C **do**
    if $C \models \mathcal{H}$ **then**
      found $\leftarrow$ `False`
      for $A \to B \in \mathcal{H}$ **do**
        if $A \not\subseteq C$ **and not** *IsMember*($A \cap C$, *IsValid*, M) **then**
          found $\leftarrow$ `True`
          replace $A \to B$ **by** $A \cap C \to B$ in $\mathcal{H}$
          exit for
      if not *found* **then**
        append $C \to M$ to the end of $\mathcal{H}$
    else
      for $A \to B \in \mathcal{H}$ **do**
        if $C \not\models A \to B$ **then**
          replace $A \to B$ **by** $A \to B \cap C$
    $i \leftarrow i + 1$
  return $\mathcal{H}$

Algorithm 5 A simulated sampling oracle that generates a subset of M uniformly at random and returns it together with an indication whether it is closed in the underlying context

Require: A attribute set M and the implication oracle *IsValid* for a formal context over M

function $UEX\langle M, IsValid\rangle$( )
  generate $X \subseteq M$ uniformly at random
  return $(X, IsMember(X, IsValid, M))$

In the previous section, we showed that, if the sequence $\{q_i(\epsilon,\delta)\}_{i=1}^{+\infty}$ is given by $q_i(\epsilon,\delta) \coloneqq \left\lceil \log_{1-\epsilon} \frac{\delta}{i(i+1)} \right\rceil$, then $\Theta\left(\frac{k}{\epsilon}\left(\log k + \log\frac{1}{\delta}\right)\right)$ samples will be generated in total, where k is the maximum number of counterexamples used by the exact-learning algorithm. For the Horn1 algorithm, it was shown in [11] that $k = O(|\hat{\mathcal{H}}||M|)$, where $\hat{\mathcal{H}}$ is the value of $\mathcal{H}$ upon termination of the algorithm. In equivalence testing, we use one simulated membership query $IsMember(\cdot,\cdot,\cdot)$ per simulated sampling query $UEX\langle\cdot,\cdot\rangle()$. Besides, in cases when a negative counterexample is returned, at most $|\hat{\mathcal{H}}| \leq |\mathcal{L}|$ simulated membership queries are issued inside the **while** loop. So, in total, Algorithm 4 requires $O\left(k|\hat{\mathcal{H}}| + \frac{k}{\epsilon}\left(\log k + \log\frac{1}{\delta}\right)\right) = O\left(|\hat{\mathcal{H}}|^2|M| + \frac{|\hat{\mathcal{H}}||M|}{\epsilon}\log\frac{|\hat{\mathcal{H}}||M|}{\delta}\right) = O\left(|\hat{\mathcal{H}}||M|\left(|\hat{\mathcal{H}}| + \frac{1}{\epsilon}\log\frac{|\hat{\mathcal{H}}||M|}{\delta}\right)\right)$ simulated membership queries or, equivalently, $O\left(|\hat{\mathcal{H}}||M|^2\left(|\hat{\mathcal{H}}| + \frac{1}{\epsilon}\log\frac{|\hat{\mathcal{H}}||M|}{\delta}\right)\right)$ implication queries to achieve PAC identification. We obtain the following theorem:

Theorem 4. *Let* $0 < \epsilon < 1$ *and* $0 < \delta < 1$. *Define*

$$q_i(\epsilon,\delta) \coloneqq \left\lceil \log_{1-\epsilon} \frac{\delta}{i(i+1)} \right\rceil.$$

Algorithm 4 takes an implication oracle for a formal context $\mathbb{K}$ *and computes* $\hat{\mathcal{H}}$*, which, with probability at least* $1-\delta$*, is an* ϵ*-Horn approximation of* $\mathbb{K}$*, using* $O\left(|\hat{\mathcal{H}}||M|^2\left(|\hat{\mathcal{H}}| + \frac{1}{\epsilon}\log\frac{|\hat{\mathcal{H}}||M|}{\delta}\right)\right)$ *implication queries.*

Example 1. The formal context "Living Beings and Water" from [1] shown in Figure 1. In this context, eight living organisms are described by nine attributes. Its canonical basis and the basis returned by Algorithm 4 using the *IsValid* oracle corresponding to this context with $\epsilon = 0.05$, $\delta = 0.1$, and $q_i(\epsilon,\delta) = \left\lceil \log_{1-\epsilon} \lambda_n(\delta) \right\rceil$, where $\lambda_i(\delta) = \frac{\delta}{\sqrt{i}} - \frac{\delta}{\sqrt{i+1}}$, are presented in Figure 2. The sequence λ was selected to be a telescoping sequence $\lambda_i(\delta) = \delta(i^{-r} - (i+1)^{-r})$ with $r = 1/2$, in accordance with our discussion in Section 4.2.

The Horn distance between this formal context and its PAC basis from Figure 2 is 0.0078125, which means that only 4 out of 512 possible subsets

	(a) needs water to live	(b) lives in water	(c) lives on land	(d) needs chlorophyll	(e) dicotyledon	(f) monocotyledon	(g) can move	(h) has limbs	(i) breast feeds
Fish leech	×	×					×		
Bream	×	×					×	×	
Frog	×	×	×				×	×	
Dog	×		×				×	×	×
Water weeds	×	×		×		×			
Reed	×	×	×	×		×			
Bean	×		×	×	×				
Corn	×		×	×		×			

Figure 1: Formal context "Living Beings and Water"

Canonical basis	**PAC basis**
1. $\varnothing \to \{a\}$	1. $\varnothing \to \{a\}$
2. $\{a,\ b,\ d\} \to \{f\}$	2. $\{a,\ b,\ d\} \to \{f\}$
3. $\{a,\ d,\ g\} \to \{b,\ c,\ e,\ f,\ h,\ i\}$	3. $\{a,\ d,\ g\} \to \{b,\ c,\ e,\ f,\ h,\ i\}$
4. $\{a,\ f\} \to \{d\}$	4. $\{a,\ f\} \to \{d\}$
5. $\{a,\ h\} \to \{g\}$	5. $\{a,\ h\} \to \{g\}$
6. $\{a,\ e\} \to \{c,\ d\}$	6. $\{a,\ e\} \to \{b,\ c,\ d,\ f,\ g,\ h,\ i\}$
7. $\{a,\ i\} \to \{c,\ g,\ h\}$	7. $\{a,\ i\} \to \{b,\ c,\ d,\ e,\ f,\ g,\ h\}$
8. $\{a,\ b,\ c,\ g,\ h,\ i\} \to \{d,\ e,\ f\}$	
9. $\{a,\ c,\ d,\ e,\ f\} \to \{b,\ g,\ h,\ i\}$	
10. $\{a,\ c,\ g\} \to \{h\}$	

Figure 2: The canonical basis and a PAC basis of the formal context from Figure 1

of M disprove the theory we obtained: $\{a, c, d, e\}$ and $\{a, c, g, h, i\}$ are positive counterexamples to the PAC basis (they are concept intents refuting some implications in the PAC basis), while $\{a, c, g\}$ and $\{a, b, c, g\}$ are negative counterexamples (closed under the PAC basis but not in the context). To compute this basis, the algorithm ran for 25 iterations (equivalence testing) and made 637 membership queries that were simulated by 951 implication queries.

6. Experiments

Let us now examine different modifications of the PAC learning algorithm[1]. Within the experiments we need to compare the following approaches:

- equivalence testing with different sequences $\lambda_n(\delta)$. We consider $\lambda_n(\delta) = \delta(n^{-q} - (n+1)^{-q})$ for $q = 1, 0.5, 0.25$ and $\lambda_n(\delta) = \dfrac{\delta}{2^n}$
- strong approximation in stochastic equivalence testing [9]
- the attribute exploration algorithm

The metrics we focus on are as follows:

- the total number of implication queries
- the number of iterations in Horn1 (equivalence testing)
- Horn distance

First, let us test different modifications in artificially generated datasets. We generated 80 formal contexts with 10-18 attributes and 30-80 objects. For every context, the number of attributes and objects was selected uniformly, and the incidence relation I was selected so that for each object g and each attribute m the indicator $(g, m) \in I$ is true with probability 0.5 (uniform distribution over the set of all contexts with given sets of attributes and objects). Figures

[1]The source code and examples of usage are publicly available at `https://github.com/Ramil0/pac-basis`

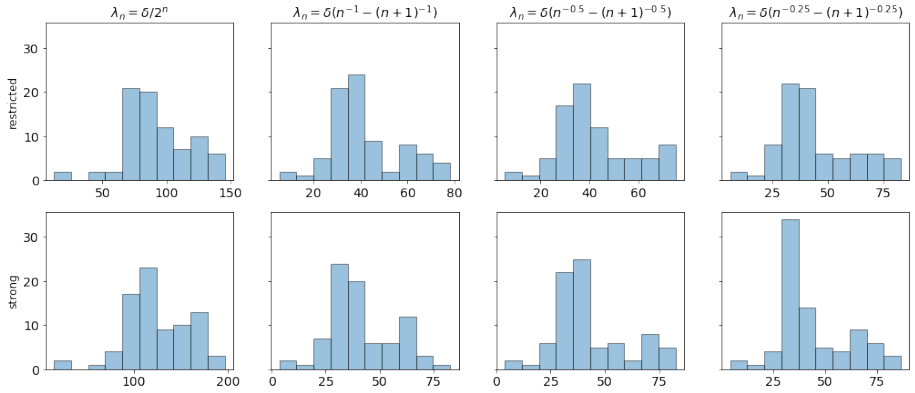

Figure 3: Distribution of the number of implication queries for different sequences (in thousands).

	$\boldsymbol{\lambda_n = \delta/2^n}$	$\boldsymbol{q = 1}$	$\boldsymbol{q = 0.5}$	$\boldsymbol{q = 0.25}$
	Horn Approximation (restricted)			
mean	91248.7	44027.5	41650.1	41436.2
std	25905	16621.8	15224.4	15204.1
	Strong version of Horn Approximation			
mean	124940.7	43689.6	42870.8	41130.3
std	34310.6	17023.6	16368.9	15257.1

Table 1: Mean and standard deviation of the number of implication queries for different sequences $\lambda_n = \delta(n^{-q} - (n+1)^{-q})$

	$\epsilon = 0.01$		$\epsilon = 0.05$		$\epsilon = 0.1$	
Setting	N_q	N_{it}	N_q	N_{it}	N_q	N_{it}
Attribute Exploration	101	-	101	-	101	-
$\boldsymbol{\lambda_n = \delta/2^n}$	248305	201	13523	66	54279	147
$\boldsymbol{\lambda_n = \delta/2^n}$ **(strong)**	172352	292	133801	272	83640	299
$\boldsymbol{q = 1}$	45836	137	3041	40	3372	40
$\boldsymbol{q = 1}$ **(strong)**	67263	263	28092	197	16422	146
$\boldsymbol{q = 0.5}$	19905	90	2374	28	1792	32
$\boldsymbol{q = 0.5}$ **(strong)**	56163	276	31025	232	22067	176
$\boldsymbol{q = 0.25}$	35372	120	2655	31	1826	28
$\boldsymbol{q = 0.25}$ **(strong)**	46934	241	14305	147	12758	135

Table 2: Results for the Zoo dataset, $\delta = 0.1$. Number of implication queries (N_q), number of iterations (N_{it}).

2 illustrates the distribution of the number of implication queries performed by the algorithms depending on the sequence $\lambda_n(\delta)$ selected. Mean and standard deviation values of these distributions are shown in Table 1.

We shall also examine the behavior of the learning algorithms in real datasets. First, we consider the Zoo dataset, but because of the previously mentioned issue with one-hot encoding of multi-valued features, we drop the categorical features (*legs* and *type*) from the data and leave only the binary ones. Table 2 demonstrates the performance of the considered algorithms on this dataset. The Horn distance in all of the setups was observed to be in the $[0.00003, 0.003]$, $[0.00003, 0.02]$, $[0.00028, 0.02]$ intervals for $\epsilon = 0.01$, $\epsilon = 0.05$, and $\epsilon = 0.1$ respectively, with minimal value in the cases with $\lambda_n = \delta/2^n$.

For this dataset, for $\epsilon = 0.01$, the algorithm returned PAC bases containing 29.4 implications on average (with 6.24 as a standard deviation) after 5 trials performed, while the exact canonical basis consists of 58 implications.

The results above demonstrate that the number of implication queries $\lambda_n(\delta) = \delta(n^{-q} - (n+1)^{-q})$ is quite close for different q, apparently because the contexts used were too small to notice a difference. However, $\lambda_n(\delta) = \dfrac{\delta}{2^n}$ works fairly worse than the other sequences. We can also note that the strong versions usually ask more queries if compared with the corresponding restricted versions.

Attribute exploration asked significantly fewer queries than all the considered

	Example 1			*Example 2*		
n	$\boldsymbol{\lambda_n = \delta/2^n}$	$\boldsymbol{q = 0.25}$	$\boldsymbol{AE}$	$\boldsymbol{\lambda_n = \delta/2^n}$	$\boldsymbol{q = 0.25}$	$\boldsymbol{AE}$
2	1048.8 ± 43.6	730.8 ± 53.8	10	145.2	93.2	10
3	159.4 ± 36.5	105.0 ± 18.6	36	282.2	109.0	17
4	346.0 ± 22.9	248.6 ± 39.2	268	572.0	216.0	28
5	66.0 ± 2.5	56.0 ± 8.48	1417.8	3140	232.4	47
6	2559.6 ± 936.6	1943.8 ± 497.09	46674	3524.0	248.2	82
7	-	-	-	5068.2	414.2	149
8	-	-	-	10776.6	219.2	280
9	-	-	-	21688.6	67.4	539

Table 3: Results for contexts from *Example 1* and *Example 2* for different n and attribute exploration (AE).

modifications of Horn1. However, if the number of pseudo-intents of the context or the number of objects is large enough, attribute exploration may perform worse since it has to issue at least $|G| + |\mathcal{L}|$ implication queries where $\mathcal{L}$ is the canonical basis. We consider contexts of the following two types:

1. *Example 1* [13]. A context with n^n objects and n^2 attributes $M_1 \cup \cdots \cup M_n$ with $|M_i| = n, M_i \cap M_j = \varnothing$ for all $1 \leq i < j \leq n$, where the object intents $\{g\}'$ are all possible subsets of the attribute set such that $|\{g\}' \cap M_i| = n-1$ for all $1 \leq i \leq n$. This context has only n implications in its canonical basis, namely, all implications of the type $M_i \to \bigcup_j M_j$ for $1 \leq i \leq n$.
2. *Example 2* [3]. A context with $3n$ objects $g_1, g_2, \ldots, g_{3n}$ and $2n+1$ attributes $m_0, m_1, \ldots, m_{2n}$, where object g_i has attribute m_j if $i \leq n, j \neq 0, j \neq i$, and $j \neq i+n$, or if $i > n$ and $j \neq i-n$. This context has exactly 2^n pseudo-intents of the form $\{m_{i_1}, m_{i_2}, \ldots, m_{i_n}\}$ where $i_j \in \{j, j+n\}$.

The results for these contexts are presented in the Table 3.

7. Summary and Outlook

Our analysis of a general technique for transforming an arbitrary exact-learning algorithm using equivalence queries into a PAC algorithm without equivalence queries has revealed that the asymptotic number of samples needed for simulating equivalence queries is lower than it was suggested previously: it

is sufficient to use the number of samples logarithmic in i for simulating the ith equivalence query as opposed to the number that depends on i linearly as suggested in [10]. This leads to an asymptotically lower overall number of samples needed during the execution of the algorithm.

Based on these results, we suggested several sampling strategies and experimentally compared them, along with the strategy from [10], in application to the problem of computing probably approximately correct implication bases through implication queries with respect to artificial and real-world data sets. Although our improvements significantly reduce the number of samples that have to be generated and, consequently, the number of queries to the oracle, the latter is still often much higher than needed by attribute exploration, a technique from formal concept analysis that makes it possible to learn the basis through implication queries exactly. One of the reasons for this is that attribute exploration stores all counterexamples obtained from the oracle in a formal context that is part of the underlying context $\mathbb{K}$ and uses them to falsify an implication before attempting to confirm it with the oracle. A similar approach can be easily integrated into our algorithm: it is possible to maintain a context composed of intents of $\mathbb{K}$ generated along the way (either in the approximate equivalence–testing procedure or during the search for an implication to be refined in the **if** part of the main loop of Algorithm 4). Since the algorithm runs in time polynomial in all relevant quantities, the number of such intents is also polynomial.

It may also be worthwhile to cache implications confirmed by the oracle since not all of them would follow from the current hypothesis $\mathcal{H}$ at every subsequent step of the algorithm. Not only they will help answer some of the implication queries without resorting to the oracle; they can also be used to generate negative counterexamples without resorting to the potentially costly sampling procedure: if a valid implication $A \to B$ does not follow from $\mathcal{H}$, then $\mathcal{H}(A)$ is a negative counterexample to $\mathcal{H}$.

As a possible next step, it would be interesting to explore some other notions of approximation. Horn distance and strong Horn distance as defined in Section

3.1 assume the universal distribution over attribute subsets, which may not always be appropriate, in particular, when there are only few closed attribute sets (see the discussion in the end of the same section). The inter-concept distance from Section 2.2 is compatible with an arbitrary distribution over objects from the universal set. Similarly, one may assume an arbitrary distribution $\mathcal{D}$ over attribute subsets $A \subseteq M$ and generalize the definitions of Horn distance and strong Horn distance as follows:

$$\operatorname{dist}(\mathcal{L}, \mathbb{K}) := \Pr_{\mathcal{D}}(A \in \operatorname{Mod} \mathcal{L} \triangle \operatorname{Int} \mathbb{K})$$

and

$$\operatorname{dist}_{\text{STRONG}}(\mathcal{L}, \mathbb{K}) := \Pr_{\mathcal{D}}(A'' \neq \mathcal{L}(A)).$$

For example, one may assume that attribute subsets are distributed according to their frequency. For a context $\mathbb{K} = (G, M, I)$ with finite G and $A \subseteq M$, this means

$$\Pr(A) = \frac{|A'|}{\sum_{B \subseteq M} |B'|}.$$

Plugging this distribution into the generalized definition of (strong) Horn distance, one obtains a sort of *frequency-aware approximation* of $\mathbb{K}$: it is biased towards ensuring the correct status w.r.t. being closed (or, in case of strong approximation, the correct closure) of subsets A with large *support* $|A'|$. In particular, sets A containing mutually exclusive attributes will have no effect on the quality of approximation, since $\Pr(A) = 0$ for such A. As discussed in Section 3.1, such sets form the bulk of all attribute subsets in contexts $\mathbb{K}$ obtained through one-hot encoding, which makes $\{\varnothing \to M\}$ a trivial ϵ-Horn approximation of $\mathbb{K}$ even for small ϵ. Using frequency-aware approximation instead solves the problem.

Algorithm 4 can be modified to produce frequency-aware approximations by substituting an oracle generating attribute subsets according to their frequency for the UEX algorithm, which generates subsets uniformly at random. When the context is available, this oracle can be simulated by polynomial-time Algorithm 1 from [21], resulting in a total–polynomial time PAC algorithm for computing frequent implications, also known as exact association rules. Whether

this approach is competitive with other algorithms for computing implications and association rules is a matter of further research.

Acknowledgments

This research was supported by SPARC, a Government of India Initiative under grant no. SPARC/2018-2019/P682/SL. We thank Aimene Belfodil for letting us know of the paper [21].

References

[1] B. Ganter, R. Wille, Formal Concept Analysis: Mathematical Foundations, Springer, Berlin/Heidelberg, 1999.

[2] H. A. Kautz, M. J. Kearns, B. Selman, Horn approximations of empirical data, Artif. Intell. 74 (1) (1995) 129–145.

[3] S. Kuznetsov, On the intractability of computing the Duquenne–Guigues base, Journal of Universal Computer Science 10 (8) (2004) 927–933.

[4] L. G. Valiant, A theory of the learnable, Commun. ACM 27 (11) (1984) 1134–1142. doi:10.1145/1968.1972.
URL http://doi.acm.org/10.1145/1968.1972

[5] E. Suzuki, Worst case and a distribution-based case analyses of sampling for rule discovery based on generality and accuracy, Appl. Intell. 22 (1) (2005) 29–36. doi:10.1023/B:APIN.0000047381.08666.c9.
URL https://doi.org/10.1023/B:APIN.0000047381.08666.c9

[6] C. Jia, X. Gao, Multi-scaling sampling: An adaptive sampling method for discovering approximate association rules, J. Comput. Sci. Technol. 20 (3) (2005) 309–318. doi:10.1007/s11390-005-0309-5.
URL https://doi.org/10.1007/s11390-005-0309-5

[7] D. Borchmann, T. Hanika, S. Obiedkov, On the usability of probably approximately correct implication bases, in: K. Bertet, D. Borchmann, P. Cellier, S. Ferré (Eds.), Formal Concept Analysis. Proceedings ICFCA 2017, Vol. 10308 of Lecture Notes in Computer Science, 2017, pp. 72–88.

[8] S. Obiedkov, Learning implications from data and from queries, in: D. Cristea, F. Le Ber, B. Sertkaya (Eds.), Formal Concept Analysis, Springer International Publishing, Cham, 2019, pp. 32–44.

[9] D. Borchmann, T. Hanika, S. Obiedkov, Probably approximately correct learning of Horn envelopes from queries, Discrete Applied Mathematics 273 (2020) 30 – 42. doi:https://doi.org/10.1016/j.dam.2019.02.036.

[10] D. Angluin, Queries and concept learning, Machine learning 2 (4) (1988) 319–342.

[11] D. Angluin, M. Frazier, L. Pitt, Learning conjunctions of Horn clauses, Machine Learning 9 (1992) 147–164.

[12] D. Angluin, Learning regular sets from queries and counterexamples, Information and Computation 75 (2) (1987) 87 – 106. doi:https://doi.org/10.1016/0890-5401(87)90052-6.
URL http://www.sciencedirect.com/science/article/pii/0890540187900526

[13] B. Ganter, S. Obiedkov, Conceptual Exploration, Springer, 2016.

[14] J. L. Guigues, V. Duquenne, Famille minimale d'implications informatives résultant d'un tableau de données binaires, Mathématiques et Sciences Humaines 24 (95) (1986) 5–18.

[15] K. Bache, M. Lichman, UCI machine learning repository (2013).
URL http://archive.ics.uci.edu/ml

[16] M. Arias, J. L. Balcázar, Construction and learnability of canonical Horn formulas, Machine Learning 85 (3) (2011) 273–297. doi:10.1007/

s10994-011-5248-5.
URL http://dx.doi.org/10.1007/s10994-011-5248-5

[17] M. Arias, J. L. Balcázar, C. Tîrnăucă, Learning definite Horn formulas from closure queries, Theoretical Computer Science 658 (Part B) (2017) 346 – 356.

[18] B. Ganter, Two basic algorithms in concept analysis, in: Proceedings of the 8th International Conference on Formal Concept Analysis, ICFCA'10, Springer-Verlag, Berlin, Heidelberg, 2010, pp. 312–340. doi:10.1007/978-3-642-11928-6_22.
URL http://dx.doi.org/10.1007/978-3-642-11928-6_22

[19] R. Khardon, Translating between Horn representations and their characteristic models, J. Artif. Intell. Res. (JAIR) 3 (1995) 349–372.

[20] F. Distel, B. Sertkaya, On the complexity of enumerating pseudo-intents, Discrete Applied Mathematics 159 (6) (2011) 450–466.

[21] M. Boley, C. Lucchese, D. Paurat, T. Gärtner, Direct local pattern sampling by efficient two-step random procedures, in: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'11, Association for Computing Machinery, New York, NY, USA, 2011, p. 582–590. doi:10.1145/2020408.2020500.
URL https://doi.org/10.1145/2020408.2020500