A Lower Bound for Learning Distributions Generated by Probabilistic Automata

Borja Balle, Jorge Castro, and Ricard Gavaldà

Universitat Politècnica de Catalunya, Barcelona
{bballe,castro,gavalda}@lsi.upc.edu

Abstract. Known algorithms for learning PDFA can only be shown to run in time polynomial in the so-called distinguishability μ of the target machine, besides the number of states and the usual accuracy and confidence parameters. We show that the dependence on μ is necessary for every algorithm whose structure resembles existing ones. As a technical tool, a new variant of Statistical Queries termed L_∞-queries is defined. We show how these queries can be simulated from samples and observe that known PAC algorithms for learning PDFA can be rewritten to access its target using L_∞-queries and standard Statistical Queries. Finally, we show a lower bound: every algorithm to learn PDFA using queries with a resonable tolerance needs a number of queries larger than $(1/\mu)^c$ for every $c < 1$.

1 Introduction

Probabilistic finite automata (PFA) are important as modeling formalisms as well as computation models. They are closely related to Hidden Markov Models (HMM's) in their ability to represent distributions on finite alphabets and also to POMDP's; see e.g. [8, 17, 18] for background.

One of the main associated problems is that of approximating the distribution generated by an unknown probabilistic automaton from samples. The problem is relatively simple if the structure of the automaton is somehow known and only transition probabilities have to be estimated, and much harder and poorly-solved in practice if the transition graph is unknown. Probabilistic Deterministic Finite Automata (PDFA) — in which the underlying automaton is deterministic but transitions still have probabilities — have been often considered as a restriction worth studying, even though they cannot generate all distributions generated by PFA [8].

The grammatical inference community has produced a substantial number of methods for learning (distributions generated by) PFA or PDFA, most of them using so-called "state split-merge" or "evidence-driven" strategies; see the references in [6, 17, 18, 7]. Many of these methods are only proved valid empirically, but some have proofs of learning in the limit.

The problem has also been intensely studied in variants of the PAC model adapted to distribution learning. Abe and Warmuth showed in [1] that hardness is not information-theoretic: one can learn (distributions generated by) PFA

M. Hutter et al. (Eds.): ALT 2010, LNAI 6331, pp. 179–193, 2010.

with samples of size polynomial in alphabet size, number of states in the target machine, and inverses of the accuracy and confidence parameters (ϵ and δ); but also that the problem is computationally intractable for large alphabet sizes, unless $\mathsf{RP} = \mathsf{NP}$. Kearns et al. [13] showed that learning PDFA even over 2-letter alphabets is computationally as hard as solving the *noisy parity learning problem*, of interest in coding theory and for which only super-polynomial time algorithms are known.

It was later observed that polynomial-time learnability is feasible if one allows polynomiality not only in the number of states but also in other measures of the target automaton complexity. Specifically, Ron et al. [16] showed that acyclic PDFA can be learned w.r.t the Kullback-Leibler divergence in time polynomial in alphabet size, $1/\epsilon$, $1/\delta$, number of target states, and $1/\mu$, where μ denotes the *distinguishability* of the target automaton, to be defined in Sect. 2. Clark and Thollard extended the result to general PDFA by considering also as a parameter the expected length of the strings generated by the automata [6]. Their algorithm, a state merge-split method, was in turn extended or refined in subsequent work [10, 15, 9, 4]. Furthermore, in [11] a PAC algorithm for learning PFA was given, similar in spirit to [7], whose running time is polynomial in the inverse of a condition parameter, intuitively an analog of μ for PFA.

Here we consider the dependence on the distinguishability parameter μ of known algorithms. We know that the sample complexity and running time of the Clark-Thollard and related algorithms is polynomially bounded on $1/\mu$ (as well as other parameters), but it is conceivable that one could also prove a polynomial bound in another parameter, much smaller but yet unidentified. We rule out this possibility for a large class of learning algorithms, intuitively those that proceed by applying statistical tests to subsets of the sample to distinguish distributions generated at different states of the target automaton. To this end, we define a variant of Kearns' statistical queries [12], called L_∞-queries. We observe that known algorithms for learning PDFA, such as Clark-Thollard and our variant [4], can be rewritten accessing the target distribution only through L_∞-queries (to infer the structure) plus standard statistical queries (to approximate transition probabilities). We then show that any algorithm that learns the class of PDFA with a given distinguishability μ from L_∞-queries and statistical queries with reasonably bounded tolerance will require more than $(1/\mu)^c$ queries for every $c < 1$. Our result thus indicates that, if PDFA learning algorithms of complexity substantially smaller than $1/\mu$ do exist, they must use their input sample quite differently from known algorithms.

While we introduce our L_∞-queries as a technical concept to formulate a lower bound, we believe they may deserve further study. Interestingly, the hard targets that force our lower bound are essentially the noiseless parity functions, which are learnable in time polynomial in the number of variables, but by our result not from L_∞-queries. Recalling that noisy parity functions seem computationally hard to learn, this suggests a connection to investigate between our L_∞-queries and noisy distribution learning, as there is one between SQ and noisy concept

learning. Additionally, we give several indications (not rigorous proofs) that L_∞-queries cannot be efficiently simulated by standard SQ.

2 Preliminaries

We consider several measures of divergence between distributions. Let D_1 and D_2 be probability distributions on a discrete set X. The Kullback–Leibler (KL) divergence is defined as

$$\mathsf{KL}(D_1 \| D_2) = \sum_{x \in X} D_1(x) \log \frac{D_1(x)}{D_2(x)}, \tag{1}$$

where the logarithm is taken to base 2. The KL is sometimes called relative entropy. The supremum distance is $L_\infty(D_1, D_2) = \max_{x \in X} |D_1(x) - D_2(x)|$, and the total variation distance is $L_1(D_1, D_2) = \sum_{x \in X} |D_1(x) - D_2(x)|$.

An algorithm learns a class of distributions $\mathcal{D}$ over some set X if for any $D \in \mathcal{D}$ and $\epsilon > 0$ it is given access to D through some oracle and outputs a hypothesis $\hat{D}$ that is ϵ-close to D w.r.t. the KL divergence, that is, $\mathsf{KL}(D \| \hat{D}) < \epsilon$.

A PDFA A is a tuple $\langle Q, \Sigma, \tau, \gamma, \xi, q_0 \rangle$ where Q is a finite set of states, Σ is the alphabet, $\tau : Q \times \Sigma \longrightarrow Q$ is the transition function, $\gamma : Q \times (\Sigma \cup \{\xi\}) \longrightarrow [0, 1]$ defines the probability of emitting each symbol from each state ($\gamma(q, \sigma) = 0$ when $\sigma \in \Sigma$ and $\tau(q, \sigma)$ is not defined), ξ is a special symbol not in Σ reserved to mark the end of a string, and $q_0 \in Q$ is the initial state. It is required that $\sum_{\sigma \in \Sigma \cup \{\xi\}} \gamma(q, \sigma) = 1$ for every state q. Transition function τ is extended to $Q \times \Sigma^*$ in the usual way. Also, the probability of generating a given string $x\xi$ from state q can be calculated recursively as follows: if x is the empty word λ the probability is $\gamma(q, \xi)$, otherwise x is a string $\sigma_0\sigma_1 \ldots \sigma_k$ with $k \geq 0$ and $\gamma(q, \sigma_0\sigma_1 \ldots \sigma_k\xi) = \gamma(q, \sigma_0)\gamma(\tau(q, \sigma_0), \sigma_1 \ldots \sigma_k\xi)$. Assuming every state of A has non-zero probability of generating some string, one can define for each state q a probability distribution D_q on Σ^*: for each x, probability $D_q(x)$ is $\gamma(q, x\xi)$. The one corresponding to the initial state D_{q_0} is called the distribution defined by A.

Definition 1. *We say distributions D_1 and D_2 are* μ-distinguishable *when $\mu \leq L_\infty(D_1, D_2)$. A PDFA A is μ-distinguishable when for each pair of states q_1 and q_2 their corresponding distributions D_{q_1} and D_{q_2} are μ-distinguishable.*

Given a multiset S of strings from Σ^* we denote by $S(x)$ the multiplicity of x in S, write $|S| = \sum_{x \in \Sigma^*} S(x)$. To each multiset S corresponds an empirical distribution $\hat{S}$ defined in the usual way, $\hat{S}(x) = S(x)/|S|$.

A parity on n variables is a function $h : \{0, 1\}^n \to \{0, 1\}$ of the form $h(x_1, \ldots, x_n) = \sum_i a_i x_i \mod 2$, for some $(a_1, \ldots, a_n) \in \{0, 1\}^n$.

The following is a simple consequence of Chebyshev-Cantelli inequality that will be used when proving the lower bound.

Lemma 1. *Let X be a random variable with expectation μ and variance σ^2. If $t > 2|\mu|$ then:*

$$\mathbb{P}\left[|X| \geq t\right] \leq 2\frac{\sigma^2}{t(t-2|\mu|)}. \tag{2}$$

3 L_∞-Queries

In this section we present a new kind of query, the L_∞-query, which we describe as a call to an oracle DIFF_∞. Roughly speaking, these queries can be used whenever the learning task is to approximate a probability distribution whose support is contained in a free monoid Σ^*. This query is an abstraction of a pattern of access to distributions appearing in algorithms that learn (distributions generated by) PDFA [3, 16, 6, 10, 15, 9, 4]. At some point, all algorithms described in these papers use samples from suffix distributions to test for state-distinctness w.r.t. the supremum distance.

Let D be a distribution over Σ^*, where Σ is a finite alphabet. If $A \subseteq \Sigma^*$ is prefix-free, we denote by D^A the conditional distribution under D of having a prefix in A. That is, for every $y \in \Sigma^*$ we have

$$D^A(y) = \frac{D(Ay)}{D(A\Sigma^*)} = \frac{\sum_{x\in A} D(xy)}{\sum_{x\in A} D(x\Sigma^*)}, \tag{3}$$

where $D(x\Sigma^*)$ is the probability under D of having x as a prefix. The oracle $\mathrm{DIFF}_\infty(D)$ answers queries of the form (A, B, α, β), where $A, B \subseteq \Sigma^*$ are (encodings of) disjoint and prefix-free sets, and $\alpha, \beta \in (0, 1)$ are real numbers. Let μ denote the supremum distance between distributions D^A and D^B; that is, $\mu = \mathrm{L}_\infty(D^A, D^B)$. Then oracle $\mathrm{DIFF}_\infty(D)$ must answer a query (A, B, α, β) according to the following rules:

1. If either $D(A\Sigma^*) < \beta$ or $D(B\Sigma^*) < \beta$, it answers "?".
2. If both $D(A\Sigma^*) > 3\beta$ and $D(B\Sigma^*) > 3\beta$, it answers some number $\hat{\mu}$ such that $|\mu - \hat{\mu}| < \alpha$.
3. Otherwise, the oracle may either answer "?" or give an α-good approximation $\hat{\mu}$ of μ, arbitrarily.

To be precise, the algorithm asking a query will provide A and B in the form of oracles deciding the membership problems for $A\Sigma^*$ and $B\Sigma^*$.

Similarly to oracles answering statistical queries [12], the price an algorithm has to pay for a call to $\mathrm{DIFF}_\infty(D)$ depends on the parameters of the query. As will be seen in the next section, a call to $\mathrm{DIFF}_\infty(D)$ with a query (A, B, α, β) can be simulated with $\tilde{O}(\alpha^{-2}\beta^{-2})$ samples from D. Accordingly, we make the following definition.

Definition 2. *An algorithm for learning a class of distributions $\mathcal{D}$ over Σ^* using L_∞-queries will be called* sample efficient *if there exists polynomials p, q, r such that for each $D \in \mathcal{D}$ the algorithm makes at most $r(1/\epsilon, |D|)$ queries with $\alpha > 1/p(1/\epsilon, |D|)$ and $\beta > 1/q(1/\epsilon, |D|)$ for each query, where $|D|$ is some measure of complexity, and it outputs a hypothesis $\hat{D}$ which is ϵ-close to D.*

Remark 1 (The role of β). An algorithm asking an L_∞-query does not know a priori the probability under D of having a prefix in A. It could happen that the region $A\Sigma^*$ had very low probability, and this might indicate that a good approximation of D in this region is not necessary in order to obtain a good estimate of D. Furthermore, getting this approximation would require a large number of examples. Thus, β allows a query to fail when at least one of the regions being compared has low probability. This prevents a learner from being penalized for asking queries whose answer might be irrelevant after all.

Remark 2 (Representation of A and B). From now on we will concentrate on the information-theoretic aspects of L_∞-queries. Hence, only the number of samples needed to simulate queries and the number of such queries needed to learn a specific class of distributions will be taken into account. We are not concerned with how A and B are encoded or how membership to them is tested from the code: the representation could be a finite automaton, a Turing machine, a hash table, a logical formula, etc.

3.1 Relation with Statistical Queries

Although L_∞-queries compute a value which is statistical in nature, it is not clear whether they can be simulated by statistical queries (or the other way round). Indeed, we provide some evidence suggesting that they cannot, at least efficiently.

To begin with, one has to say what would be the equivalent of statistical queries when the target of the learning process is a distribution instead of a concept. Recall that in the usual statistical query model one asks queries of the form (χ, α) where $\chi : X \times \{0,1\} \to \{0,1\}$ is a predicate and $0 < \alpha < 1$ is some tolerance. If D is a distribution over X and $f : X \to \{0,1\}$ is a concept, a query (χ, α) to the oracle $\mathrm{SQ}(f, D)$ answers with an α-good approximation $\hat{p}_\chi$ of $p_\chi = \mathbb{P}_x[\chi(x, f(x)) = 1]$, where x is drawn according to D. Kearns interprets this oracle as a proxy to the usual PAC example oracle $\mathrm{EX}(f, D)$ abstracting the fact that learners usually use samples only to obtain statistical properties about concept f under distribution D. Note that oracle $\mathrm{SQ}(f, D)$ can be simulated online using $\mathrm{EX}(f, D)$: seeing one example $(x, f(x))$ at a time, check whether $\chi(x, f(x)) = 1$ and discard it, only keeping track of the number of examples seen so far and how many of them satisfied the predicate. An obvious adaptation of statistical queries for learning distributions over X is to do the same forgetting about labels. Then $\chi : X \to \{0,1\}$ is again a predicate, and the oracle $\mathrm{SQ}(D)$ returns an α-good approximation of $\mathbb{P}_x[\chi(x) = 1]$. Since χ is the characteristic function of some subset of X, learners can ask the oracle for an approximation to the probability of any event. We assume that this is the natural translation of statistical queries for distribution learning.

As in the case of concept learning, statistical queries for distributions can be simulated online with essentially constant memory: just count elements in the sample satisfying the predicate. Now, this does not seem possible for L_∞-queries, where in order to compute the supremum distance between two empirical

distributions one needs to collect sets of examples, estimate the probabilities of elements in the sets and compare these probabilities to see which one defines the maximum difference. This indicates that a single statistical query can not be used to simulate a L_∞-query. However, this does not preclude the possibility that L_∞-queries can be simulated with a larger number of statistical queries.

An obvious such simulation is: given access to oracles $SQ(D^A)$ and $SQ(D^B)$, obtain approximations of $D^A(x)$ and $D^B(x)$ for each x in the support and then return the largest difference $|D^A(x) - D^B(x)|$. This is not feasible when the support is infinite, although for most reasonable classes of distributions with infinite support the string defining the supremum distance cannot be very long. But even for statistical queries that return exact probabilities, this approach amounts to finding a string where the supremum distance between two distributions is attained. A problem that was shown to be NP-hard for the case of distributions generated by probabilistic automata in [14]. On the other hand, when one is not asking for particular probabilities, but samples from the distributions are available instead, the empirical supremum distance is usually a good approximation of the actual distance provided enough examples are available. This is the topic of next section.

We currently have a candidate class of distributions which we believe can rule out the possibility of simulating L_∞-queries using a polynomial number of statistical queries.

3.2 Simulation

In this section we show how to simulate calls to $DIFF_\infty(D)$ using examples from D provided by the classical PAC example oracle $EX(D)$. Our fist lemma says that the supremum distance between two arbitrary distributions over Σ^* can be approximated with a moderate number of examples provided a similar number of examples from both distributions is available.

Let D^A and D^B be two distributions over Σ^*. Let S^A be a sample of size n_A from D^A and S^B a sample of size n_B from D^B. Define $\mu = L_\infty(D^A, D^B)$ and its empirical estimation $\hat{\mu} = L_\infty(\hat{S}^A, \hat{S}^B)$. Fix some error probability $0 < \delta < 1$, an approximation factor $0 < \alpha < 1$, and an arbitrary constant $0 < c < 1$. Now define

$$N_1 = \frac{6}{\alpha^2 c} \ln \frac{24}{\alpha^2 c \delta}. \tag{4}$$

Lemma 2. *If $n_A, n_B \in [cN, N]$ for some $N > N_1$, then $|\hat{\mu} - \mu| \leq \alpha$ with probability at least $1 - \delta/2$.*

The proof is based on Chernoff bounds and is omitted.

Now we describe a simulation of L_∞-queries using the usual $EX(D)$ oracle from the PAC model. For any distribution D, each call to $EX(D)$ takes unit time and returns an example drawn according to D. As it is usual in the PAC model, the simulation will have some error probability to account, among other things, for the fact that with low probability examples provided by $EX(D)$ can be unrepresentative of D.

Let D be a distribution over Σ^*. Fix some L_∞-query (A, B, α, β) and some error probability δ. Now D^A and D^B will be suffix distributions of D; that is, conditional distributions obtained when words have a prefix in A or B. Let $p_A = D(A\Sigma^*)$ (respectively, $p_B = D(B\Sigma^*)$) denote the probability that a word drawn according to D has a prefix in A (respectively, in B). As before, μ will be the supremum distance between D^A and D^B.

Given a sample S from D, a sample S^A from D^A is obtained as follows. For each word $x \in S$, check whether $x = yz$ with $y \in A$. If this is the case, add z to S^A. The multiset obtained,

$$S^A = \{z : yz \in S \text{ and } y \in A\}, \tag{5}$$

is a sample from D^A. Note that since A is prefix-free, each word in S contributes at most one word to S^A, and thus all examples in S^A are mutually independent. Similarly, a sample S^B from D^B is obtained. Let n_A and n_B denote the respective sizes of S^A and S^B.

In order to simulate a call to $\mathrm{DIFF}_\infty(D)$ with query (A, B, α, β), draw a sample S of size N from D using $\mathrm{EX}(D)$. Then, build samples S^A and S^B from S and obtain approximations $\hat{p}_A = n_A/N$ and $\hat{p}_B = n_B/N$ of p_A and p_B, respectively. If either $\hat{p}_A < 2\beta$ or $\hat{p}_B < 2\beta$, return "?". Otherwise, return $\hat{\mu} = \mathrm{L}_\infty(\hat{S}^A, \hat{S}^B)$.

The following theorem shows that $\tilde{O}(\alpha^{-2}\beta^{-2})$ samples are enough for the simulation to succeed with high probability.

Theorem 1. *For any distribution D over Σ^*, a L_∞-query (A, B, α, β) to the oracle $\mathrm{DIFF}_\infty(D)$ can be simulated with error probability smaller than δ using $N > N_0$ calls to the oracle $\mathrm{EX}(D)$, where*

$$N_0 = \max\left\{\frac{3}{\alpha^2\beta}\ln\frac{12}{\alpha^2\beta\delta}, \frac{1}{2\beta^2}\ln\frac{8}{\delta}\right\}. \tag{6}$$

Proof. It follows from Chernoff bounds that $\hat{p}_A$ and $\hat{p}_B$ will both be β-good approximations with probability at least $1 - \delta/2$ if $N > (1/2\beta^2)\ln(8/\delta)$. Thus, the simulation will answer "?" correctly with high probability. On the other side, if both $\hat{p}_A \geq 2\beta$ and $\hat{p}_B \geq 2\beta$, then by Lemma 2 with $c = 2\beta$ the estimate $\hat{\mu}$ will be α-good with probability at least $1 - \delta/2$. □

Remark 3 (Running time of the simulation). Although the number of examples required by the simulation bounds its running time from below, this number does not completely determine how long the simulation will take. In fact, the time required to check if $x \in \Sigma^*$ belongs to $A\Sigma^*$ or $B\Sigma^*$ affects the total running time. Furthermore, depending on the representation of A and B, checking whether x has a prefix in one of them may depend on its length $|x|$. Thus, if $T_A(m)$ and $T_B(m)$ represent the time needed to check if a string of length m has a prefix in A and B, respectively, the *expected* running time of the simulation using N examples is $O(N\,\mathbb{E}_x(\max\{T_A(|x|), T_B(|x|)\}))$. Note that if A and B are represented by automata, then $T_A(m), T_B(m) \leq cm$ for some constant c. In this

case, the expected running time of the simulation is $O(NL)$, where $L = \mathbb{E}_x[|x|]$ is the expected length of D. This justifies the appearance of L in running time bounds for algorithms learning PDFA in the PAC model.

4 Lower Bound

In this section we prove that no sample efficient L_∞-query algorithm satisfying some restrictions can learn a certain class of distributions $\mathcal{D}_n$. Since this class is a subclass of all PDFA with $\Theta(n)$ states, it will follow that the class of distributions generated by PDFA is not learnable sample efficiently from L_∞-queries.

Let $\mathcal{P}_n$ be the set of parities on n variables. Consider the class of distributions $\mathcal{D}_n$ over $\{0,1\}^{n+1}$ where there is a distribution D_h for each parity $h \in \mathcal{P}_n$ which for any $x \in \{0,1\}^n$ and $y \in \{0,1\}$ satisfies $D_h(xy) = 2^{-n}$ if $h(x) = y$ and $D_h(xy) = 0$ otherwise. The class $\mathcal{D}_n$ contains 2^n distributions. Note that each one of these distributions can be represented by a PDFA with at most $2n + 2$ states.

We will show that for n large enough, the class $\mathcal{D}_n$ can not be learned with a sample efficient L_∞-query algorithm. To do so, an adversary answering the queries asked by a learning algorithm is provided. Then it is shown that very little information about the underlying distribution can be gained with a sub-exponential number of such queries when answers are provided by the adversary. The argument is similar in nature to that used in [12] to prove that parities can not be learned in the statistical query model. Basically, we show that for each answer the number of distributions in $\mathcal{D}_n$ that are inconsistent with it is at most a sub-exponential number. Since there are an exponential number of distributions in $\mathcal{D}_n$, after a sub-exponential number of queries only a small fraction of the whole set of distributions has been ruled out. Thus, the adversary can always find a distribution which is consistent with every answer given to the algorithm but still has large error with respect to the hypothesis provided by the learner.

We present our lower bound for algorithms using L_∞-queries only. The argument for dealing with standard SQ queries, in case the algorithm uses both types, is exactly as in the lower bound proved by Kearns for concept learning parities, and we omit it for brevity. Let L be a sample efficient algorithm for learning $\mathcal{D}_n$ using L_∞-queries only. Fix ϵ to be some constant smaller than $1/9$. Now, let $p(n)$ and $q(n)$ be two functions such that for each query (A, B, α, β) asked by L the following holds: 1) $\alpha > 1/p(n)$, 2) $\beta > 1/q(n)$, 3) $p(n)$ and $q(n)$ are $2^{o(n)}$, and 4) there exist positive k_A and k_B such that $A \subseteq \{0,1\}^{k_A}$ and $B \subseteq \{0,1\}^{k_B}$. A query (A, B, α, β) satisfying 4 will be called *strict*. Restricting to strict queries is a technical condition which we believe can be removed in a more careful analysis. Nonetheless, this condition holds for the PDFA learning algorithms we are aware of when restricted to target PDFA representing parities. That is because a non-strict query in this setting means the algorithm is considering states generating words of different lengths, and this in turn means hypotheses having infinite KL with any $D \in \mathcal{D}_n$.

The following theorem states our lower bound formally. Its qualitative corollary is immediate.

Theorem 2. *Let functions $p(n)$ and $q(n)$ be $2^{o(n)}$. If $\epsilon \leq 1/9$ and n is large enough, an algorithm using strict L_∞-queries where $\alpha > 1/p(n)$ and $\beta > 1/q(n)$ for any query (A, B, α, β) cannot learn $\mathcal{D}_n$ with $o(2^n/\max\{p(n)^2 q(n), q(n)^2\})$ queries.*

Corollary 1. *For $\epsilon \leq 1/9$ and n large enough, the class $\mathcal{D}_n$ cannot be learned sample efficiently with L_∞-queries.*

Proof (of Theorem 2). Let (A, B, α, β) be a strict L_∞-query asked by L. Without loss of generality we assume that $k_A \geq k_B$. If $k_A \leq n$, for any $a \in \{0, 1\}$, we define the quantity $\theta_{A,a}$ as $(-1)^a/2$ if the all zero string belongs to A and as 0 otherwise. If $k_A = n+1$, the quantity θ'_A is defined as $(-1)^a/2$ if $0 \cdots 0a \in A$ and $0 \cdots 0\bar{a} \notin A$, where $a \in \{0, 1\}$ and $\bar{a}$ means negation; we let $\theta'_A = 0$ otherwise. Quantities $\theta_{B,b}$ and θ'_B are defined similarly.

The adversary is defined and analyzed in two parts. In the first part we consider the cases where it answers "?", while the situations where some $\hat{\mu}$ is answered are considered in the second part. Our analysis begins by considering the following three cases, where the adversary answers the query with "?":

1. If either $k_A, k_B > n + 1$.
2. If either $k_A \leq n$ with $|A| < 2^{k_A}\beta$ or $k_B \leq n$ with $|B| < 2^{k_B}\beta$.
3. If either $k_A = n + 1$ with $|A| < 2^{n+2}\beta - 2\theta'_A$ or $k_B = n + 1$ with $|B| < 2^{n+2}\beta - 2\theta'_B$.

Recall that an oracle answering L_∞-queries may answer "?" whenever the probability of the words with a prefix in A or B is smaller than 3β. We will only reason about A; by symmetry, the same arguments work for B. In case 1, it is obvious that $D_h(A\{0,1\}^*) = 0$ for any parity $h \in \mathcal{P}_n$ and therefore the answer "?" is consistent with all distributions in $\mathcal{D}_n$. Now, in case 2, if $k_A \leq n$ then $D_h(A\{0,1\}^*) = 2^{-k_A}|A|$ independently of h. Thus, the answer "?" is consistent with all parities if $|A| < 2^{k_A}\beta$. Lastly, for case 3 assume that $k_A = n + 1$. Now $D_h(A\{0,1\}^*) = D_h(A)$, and this probability does depend on h since it equals 2^{-n} times the number of words $xy \in A$ such that $h(x) = y$. Hence, it is not possible for the answer "?" to be consistent with all distributions, although we show that it is consistent with most of them. If parity h is chosen uniformly at random, by a routine calculation one shows that

$$\mathbb{E}_h[D_h(A)] = 2^{-n}\left(\frac{|A|}{2} + \theta'_A\right). \tag{7}$$

So, our adversary answers "?" whenever $\mathbb{E}_h[D_h(A)] < 2\beta$. The number of distributions in $\mathcal{D}_n$ inconsistent with this answer can be upper bounded using a probabilistic argument. By Chebyshev's inequality,

$$\mathbb{P}_h[D_h(A) > 3\beta] \leq \mathbb{P}_h[|D_h(A) - \mathbb{E}_h[D_h(A)]| > \beta] \leq \frac{\mathbb{V}_h[D_h(A)]}{\beta^2}. \tag{8}$$

The leftmost probability in this equation is the number of inconsistent distributions times 2^{-n}. Now, write A as the disjoint union $A = A_{01} \cup A'$, where for any $x \in \{0,1\}^n$ the words $x0$ and $x1$ belong to A_{01} if and only if $x0, x1 \in A$. This partition implies that for any parity h exactly a half of the words $xy \in A_{01}$ satisfy $h(x) = y$. It follows then that a part of $D_h(A)$ does not depend on h: $D_h(A) = 2^{-n-1}|A_{01}| + D_h(A')$. Thus only the part A' contributes to the variance of $D_h(A)$. Taking this into account, a computation with indicator variables and a standard linear algebra argument show that

$$\mathbb{V}_h[D_h(A)] = 2^{-2n}\left(\frac{|A'|}{4} - {\theta'_A}^2\right). \tag{9}$$

Applying the bounds $1/q(n) < \beta < 1$, the definition of θ'_A and recalling the assumption $|A| < 2^{n+2}\beta - 2\theta'_A$, we see that (8) and (9) imply that the number of distributions inconsistent with the answer "?" is, in this case, smaller than $q(n)^2$.

So far, we have shown that whenever the adversary answers "?", at most $q(n)^2$ distributions in $\mathcal{D}_n$ are inconsistent with this answer. Now we move ahead to the second part of the analysis. In the rest of the cases the adversary answers with some $\hat{\mu}$. In particular:

1. If $k_B < k_A < n+1$ then $\hat{\mu} = 2^{k_A - n - 1}$.
2. If $k_B < k_A = n+1$ then $\hat{\mu} = 1$.
3. If $k_B = k_A$ then $\hat{\mu} = 0$.

In what follows we show that, if n is large enough, the number of distributions inconsistent with the answer is, in each case, bounded by $\max\{p(n)^2 q(n), q(n)^2\}$.

Before proceeding, observe that in all these cases $k_A \leq n+1$ and for any parity h the conditional distribution D_h^A has support $\{0,1\}^{n+1-k_A}$ with the convention that $\{0,1\}^0 = \{\lambda\}$, the set with the empty string. Furthermore, if $k_A \leq n$ we can write any parity $h \in \mathcal{P}_n$ as $h = f + g$ where $f \in \mathcal{P}_{k_A}$ and $g \in \mathcal{P}_{n-k_A}$, with the convention that $\mathcal{P}_0$ contains only the constant 0. Then, for any $x = yz$ with $y \in \{0,1\}^{k_A}$ and $z \in \{0,1\}^{n-k_A}$ we have $h(x) = f(y) + g(z)$. Everything holds equally when replacing A by B.

We start now with case 1. Like before, we have $D_h(A\{0,1\}^*) = 2^{-k_A}|A| = p_A$ and $D_h(B\{0,1\}^*) = 2^{-k_B}|B| = p_B$ for any parity h. Now, given $y \in \{0,1\}^{n-k_A}$ and $z \in \{0,1\}$, by definition we can write

$$D_h^A(yz) = \frac{\sum_{x \in A} D_h(xyz)}{p_A}. \tag{10}$$

Writing $h = f + g$, define $A_f^a = \{x \in A : f(x) = a\}$ for $a \in \{0,1\}$. This yields the partition $A = A_f^0 \cup A_f^1$. The numerator in (10) can then be written as

$$\sum_{x \in A} D_h(xyz) = \sum_{x \in A_f^0} D_h(xyz) + \sum_{x \in A_f^1} D_h(xyz). \tag{11}$$

Recall that $D_h(xyz) = 2^{-n}$ if and only if $h(xy) = f(x) + g(y) = z$. Hence, if $g(y) = z$ then $D_h(xyz) = 2^{-n}$ for all $x \in A_f^0$. Similarly, $D_h(xyz) = 2^{-n}$ for all

$x \in A_f^1$ if and only if $g(y) \neq z$. Thus, the following expression for the conditional distribution D_h^A holds:

$$D_h^A(yz) = \frac{2^{-n}}{p_A} \cdot \begin{cases} |A_f^0| & \text{if } g(y) = z \ , \\ |A_f^1| & \text{if } g(y) \neq z \ . \end{cases} \tag{12}$$

Note that for any parity h both values can be attained for some choice of y and z. With the obvious modifications, these expressions hold for B too.

Now we compute the supremum distance between D_h^A and D_h^B for any $h \in \mathcal{P}_n$. Write $h = f + g = f' + g'$ where $f \in \mathcal{P}_{k_A}$, $f' \in \mathcal{P}_{k_B}$, $g \in \mathcal{P}_{n-k_A}$ and $g' \in \mathcal{P}_{n-k_B}$. Then $\mathrm{L}_\infty(D_h^A, D_h^B)$ equals

$$\max\left\{ \frac{2^{k_A-n}}{|A|} \max\left\{|A_f^0|, |A_f^1|\right\}, \frac{2^{k_B-n}}{|B|} \max\left\{|B_{f'}^0|, |B_{f'}^1|\right\} \right\} \tag{13}$$

because D_h^A and D_h^B are distributions over suffixes of different lengths. Since $\max\{|A_f^0|, |A_f^1|\} \geq |A|/2$ and $\max\{|B_{f'}^0|, |B_{f'}^1|\} \leq |B|$, we see that

$$\mathrm{L}_\infty(D_h^A, D_h^B) = \frac{2^{k_A-n}}{|A|} \max\left\{|A_f^0|, |A_f^1|\right\}. \tag{14}$$

Note this distance only depends on the first k_A bits of the parity h.

In order to count how many distributions in $\mathcal{D}_n$ are inconsistent with the answer $\hat{\mu} = 2^{k_A-n}/2$ given by the adversary we use another probabilistic argument. Assume that a parity $h \in \mathcal{P}_n$ is chosen uniformly at random and let $f \in \mathcal{P}_{k_A}$ be the parity obtained from the first k_A bits of h. Then it is easy to verify that for $a \in \{0, 1\}$ we have

$$\mathbb{E}_h[|A_f^a|] = \frac{|A|}{2} + \theta_{A,a}, \text{ and } \mathbb{V}_h[|A_f^a|] = \frac{|A|}{4} + \frac{\theta_{A,a}}{2}. \tag{15}$$

Using these computations and recalling that $\alpha \geq 1/p(n)$ and $p_A = |A|/2^{k_A} > \beta \geq 1/q(n)$, we apply Lemma 1 and get, after some calculations,

$$\mathbb{P}_h\left[\left|\frac{|A_f^a|}{|A|} - \frac{1}{2}\right| > \alpha 2^{n-k_A}\right] \leq \frac{p(n)^2 q(n)\left(2^{k_A} + 2q(n)\theta_{A,a}\right)}{2^{n+1}\left(2^n - 2p(n)q(n)|\theta_{A,a}|\right)}. \tag{16}$$

Since $k_A \leq n$, $|\theta_{A,a}| \leq 1/2$ and $\theta_{A,0} + \theta_{A,1} = 0$, a union bound yields

$$\mathbb{P}_h\left[\left|\mathrm{L}_\infty(D_h^A, D_h^B) - \frac{2^{k_A-n}}{2}\right| > \alpha\right] \leq \frac{p(n)^2 q(n)}{2^n - p(n)q(n)}. \tag{17}$$

Therefore, the number of distributions in $\mathcal{D}_n$ inconsistent with the answer given by our adversary in this case is asymptotically bounded from above by $p(n)^2 q(n)$.

Case 2 is next. Because the adversary has not answered "?" we know that $|A| \geq 2^{n+2}\beta - 2\theta'_A$ and $|B| \geq 2^{k_B}\beta$. Since $k_A = n + 1$ it follows that $D_h^A(\lambda) = 1$

if $D_h(A\{0,1\}^*) \neq 0$, otherwise we define $D_h^A(\lambda) = 0$. Hence, for any parity h the supremum distance between D_h^A and D_h^B can be written as

$$\mathrm{L}_\infty(D_h^A, D_h^B) = \max\left\{ D_h^A(\lambda), \frac{2^{k_B-n}}{|B|} \max\{|B_f^0|, |B_f^1|\} \right\}, \tag{18}$$

where f corresponds to the first k_B bits of h. Note that $\mathrm{L}_\infty(D_h^A, D_h^B) \neq 1$ implies that $D_h(A\{0,1\}^*) = 0$. Now there are two possibilities. If $A_{01} \neq \varnothing$ then for any parity h we have $D_h(A\{0,1\}^*) \neq 0$ and therefore the answer $\hat{\mu} = 1$ is consistent with every parity. On the other hand, $A_{01} = \varnothing$ implies that $A = A'$ and $|A| \leq 2^n$ because for each prefix $x \in \{0,1\}^n$ at most one of $x0$ and $x1$ belongs to A. In the latter situation we have $\mathbb{P}_h[|\mathrm{L}_\infty(D_h^A, D_h^B) - 1| > \alpha] \leq \mathbb{P}_h[\mathrm{L}_\infty(D_h^A, D_h^B) \neq 1] = \mathbb{P}_h[D_h(A\{0,1\}^*) = 0]$. This last probability is bounded by

$$\mathbb{P}_h\left[|D_h(A\{0,1\}^*) - \mathbb{E}_h\left[D_h(A\{0,1\}^*)\right]| \geq \mathbb{E}_h\left[D_h(A\{0,1\}^*)\right]\right], \tag{19}$$

which in turn can be bounded using Chebyshev's inequality by

$$\frac{\mathbb{V}_h\left[D_f(A\{0,1\}^*)\right]}{\mathbb{E}_h\left[D_h(A\{0,1\}^*)\right]^2}. \tag{20}$$

Therefore, by (7) and (9) and the bounds on $|A|$, θ'_A and β, we see that the at most $q(n)^2/16$ distributions in $\mathcal{D}_n$ are inconsistent with the answer $\hat{\mu} = 1$.

Now we consider case number 3, where $k = k_A = k_B$ and the adversary responds $\hat{\mu} = 0$. Two distinct situations need to be considered: $k = n + 1$ and $k \leq n$. Assume first that $k = n + 1$. An argument already used in case 2 shows that if both $A_{01} \neq \varnothing$ and $B_{01} \neq \varnothing$, then for each parity h it holds that $D_h^A(\lambda) = D_h^B(\lambda) = 1$ and therefore $\mathrm{L}_\infty(D_h^A, D_h^B) = 0$ irrespective of h. In this case the answer is consistent with every distribution. If exactly one of $A_{01} = \varnothing$ and $B_{01} = \varnothing$ holds, suppose it is $A_{01} = \varnothing$ without loss of generality, then $\mathrm{L}_\infty(D_h^A, D_h^B) \neq 0$ whenever $D_h(A\{0,1\}^*) = 0$, which, by case 2, happens for at most $q(n)^2/16$ distributions in $\mathcal{D}_n$. Now, if both $A_{01} = \varnothing$ and $B_{01} = \varnothing$, it is easy to see using a union bound that $\hat{\mu} = 0$ is inconsistent with at most $q(n)^2/8$ distributions.

Assume now that $k \leq n$. Then, from the fact that $|A| = |A_f^0| + |A_f^1|$ and $|B| = |B_f^0| + |B_f^1|$, the following expression for the L_∞ distance between D_h^A and D_h^B can be deduced:

$$\mathrm{L}_\infty(D_h^A, D_h^B) = 2^{k-n} \max_{a \in \{0,1\}} \left\{ \left| \frac{|A_f^a|}{|A|} - \frac{|B_f^a|}{|B|} \right| \right\} = 2^{k-n} \left| \frac{|A_f^0|}{|A|} - \frac{|B_f^0|}{|B|} \right|, \tag{21}$$

where $f \in \mathcal{P}_k$ is formed with the first k bits of h. We will show that in this case $\hat{\mu} = 0$ is a response consistent with most of the distributions in $\mathcal{D}_n$. Write $X_f = |A_f^0|/|A| - |B_f^0|/|B|$ and note that by (15) we have $\mathbb{E}_h[X_f] = \theta_A/|A| - \theta_B/|B|$, where, for simplicity, we write θ_A and θ_B for $\theta_{A,0}$ and $\theta_{B,0}$ respectively. Performing further computations one sees that

$$\mathbb{E}_h\left[X_f^2\right] = \frac{1}{4|A|} + \frac{1}{4|B|} + \frac{\theta_A}{|A|^2} + \frac{\theta_B}{|B|^2}. \tag{22}$$

Combining the last two expressions and observing that $\theta_A\theta_B = 0$, the following formula for the variance of X_f is obtained:

$$\mathbb{V}_h[X_f] = \frac{1}{4|A|} + \frac{1}{4|B|} + \frac{\theta_A}{2|A|^2} + \frac{\theta_B}{2|B|^2}. \tag{23}$$

Since $\beta > 1/q(n)$ implies $|A|, |B| > 2^k/q(n)$, plugging these bounds in previous formulas yields:

$$|\mathbb{E}_h[X_f]| \leq \frac{q(n)}{2^{k+1}}, \text{ and } \mathbb{V}_h[X_f] \leq \frac{q(n)}{2^{k+1}} + \frac{q(n)^2}{2^{2k+1}}. \tag{24}$$

Lemma 1 then yields the bound

$$\mathbb{P}_h[\mathrm{L}_\infty(D_h^A, D_h^B) > \alpha] = \mathbb{P}_h[|X_f| > \alpha 2^{n-k}] \leq \frac{p(n)^2 q(n)(1 + q(n)/2^n)}{2^{n+1} - 2p(n)q(n)}, \tag{25}$$

where we have used that $\alpha > 1/p(n)$ and $k \leq n$. From this bound, the number of distributions for which the answer is inconsistent is asymptotically $p(n)^2 q(n)/2$.

So far we have seen that, if n is large enough, for any strict L_∞-query issued by L, the answer given by the adversary is inconsistent with at most $\max\{p(n)^2 q(n), q(n)^2\}$ distributions in $\mathcal{D}_n$. Since there are 2^n distributions for any given n, after sub-exponentially many queries there will be still many different distributions in $\mathcal{D}_n$ consistent with all the answers provided to the learner.

Now, note that the relative entropy between any two distributions in $\mathcal{D}_n$ is infinite because they have different supports. Thus, for n big enough, if L outputs a hypothesis in $\mathcal{D}_n$, it will have infinite error with high probability with respect to the random choice of a target distribution in $\mathcal{D}_n$. Recalling that for each pair of distributions in $\mathcal{D}_n$ we have $\mathrm{L}_1(D_f, D_g) = 1$, we also get a lower bound for learning $\mathcal{D}_n$ using the variation distance as error measure. Now assume L outputs some distribution $\hat{D}$, not necessarily in $\mathcal{D}_n$, such that $\mathsf{KL}(D_f \| \hat{D}) \leq \epsilon$ for some $D_f \in \mathcal{D}_n$. Then it follows from Pinsker's inequality [5] that $\mathsf{KL}(D_g \| \hat{D}) \geq (1/2\ln 2)(1 - \sqrt{2\ln 2\epsilon})^2$ for any other distribution D_g different from D_f. Since $\epsilon \leq 1/9$, we then have $\mathsf{KL}(D_g \| \hat{D}) > 2/9$. Therefore, if a target distribution in $\mathcal{D}_n$ is chosen at random, then L will have large error with high probability. □

4.1 A Lower Bound in Terms of Distinguishability

A lower bound on the complexity of learning the class of PDFA with a given distinguishability now follows easily using a padding argument. We ignore the dependence on ϵ in the statement.

An L_∞-query algorithm is (p, q)-bounded if, for every query (A, B, α, β) it asks, $\alpha > 1/p$ and $\beta > 1/q$, where p and q may depend on inputs of the algorithm and the complexity of the target distribution.

Corollary 2. *Let p and q be functions in $n^{O(1)} \cdot (1/\mu)^{o(1)}$. For every $c < 1$, there is no (p, q)-bounded L_∞-query algorithm that, for every n and μ, learns the class of distributions generated by PDFA with n states and distinguishability μ with $(1/\mu)^c$ queries.*

Proof. Recall the class of distributions $\mathcal{D}_k$ from the proof of Theorem 2. For every m and k, define the class of distributions $\mathcal{C}_{m,k}$ as follows: for every distribution D in $\mathcal{D}_k$, there is a distribution in $\mathcal{C}_{m,k}$ that gives probability $D(x)$ to each string of the form $0^m x$, and 0 to strings not of this form. Every distribution in $\mathcal{D}_k$ is generated by a PDFA with $2k$ states and distinguishability 2^{-k}. It follows that every distribution $\mathcal{C}_{m,k}$ is generated by a PDFA with $m + 2k$ states and distinguishability also 2^{-k}.

Now let $m = m(k)$ grow as $2^{o(k)}$. Assume for contradiction the existence of an algorithm as in the statement of the theorem. This algorithm is (p, q)-bounded with p and q that grow like $(m + 2k)^{O(1)} \cdot (1/2^{-k})^{o(1)} = 2^{o(k)}$. By an immediate reduction, the algorithm can be used to learn the classes of distributions D_k with 2^{kc} queries for some $c < 1$. But since 2^{kc} is in $o(2^{k-o(k)})$, this contradicts Theorem 2. □

5 Conclusion

Let us remark that the lower bound in the previous section, as other lower bounds for learning from statistical queries, is strangely both information-theoretic and complexity-theoretic. We know, by the results in [1], that the barrier for learning PDFA is complexity-theoretic, not information-theoretic. Yet, our result says that, for algorithms that can only see their target through the lens of statistical and L_∞-queries, the problem becomes information-theoretic.

As open problems on which we are working, we shall mention possible relations between L_∞-queries and other variants of SQ proposed in the literature, and in particular those by Ben-David et al. [2] for distribution learning. Another problem is narrowing the gap between lower and upper bound: our lower bound plus the simulation we describe does not forbid the existence of algorithms that learn from $O(1/\mu)$ samples. Yet, the best bounds we can prove now for the Clark-Thollard algorithm and its variants are larger, namely $\Theta(1/\mu^2)$ at best.

Acknowledgements. This work is partially supported by the Spanish Ministry of Science and Technology contracts TIN-2008-06582-C03-01 (SESAAME) and TIN-2007-66523 (FORMALISM), by the Generalitat de Catalunya 2009-SGR-1428 (LARCA), and by the EU PASCAL2 Network of Excellence (FP7-ICT-216886). B. Balle is supported by an FPU fellowship (AP2008-02064) of the Spanish Ministry of Education.

References

[1] Abe, N., Warmuth, M.K.: On the computational complexity of approximating distributions by probabilistic automata. Mach. Learn. 9(2-3), 205–260 (1992)

[2] Ben-David, S., Lindenbaum, M.: Learning distributions by their density levels: A paradigm for learning without a teacher. J. Comput. Syst. Sci. 55(1), 171–182 (1997)

[3] Carrasco, R.C., Oncina, J.: Learning deterministic regular grammars from stochastic samples in polynomial time. RAIRO (Theoretical Informatics and Applications) 33(1), 1–20 (1999)
[4] Castro, J., Gavaldà, R.: Towards feasible PAC-learning of probabilistic deterministic finite automata. In: Clark, A., Coste, F., Miclet, L. (eds.) ICGI 2008. LNCS (LNAI), vol. 5278, pp. 163–174. Springer, Heidelberg (2008)
[5] Cesa-Bianchi, N., Lugosi, G.: Prediction, Learning, and Games. Cambridge University Press, New York (2006)
[6] Clark, A., Thollard, F.: PAC-learnability of probabilistic deterministic finite state automata. Journal of Machine Learning Research (2004)
[7] Denis, F., Esposito, Y., Habrard, A.: Learning rational stochastic languages. In: Lugosi, G., Simon, H.U. (eds.) COLT 2006. LNCS (LNAI), vol. 4005, pp. 274–288. Springer, Heidelberg (2006)
[8] Dupont, P., Denis, F., Esposito, Y.: Links between probabilistic automata and hidden markov models: probability distributions, learning models and induction algorithms. Pattern Recognition 38(9), 1349–1371 (2005)
[9] Gavaldà, R., Keller, P.W., Pineau, J., Precup, D.: PAC-learning of markov models with hidden state. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) ECML 2006. LNCS (LNAI), vol. 4212, pp. 150–161. Springer, Heidelberg (2006)
[10] Guttman, O., Vishwanathan, S.V.N., Williamson, R.C.: Learnability of probabilistic automata via oracles. In: Jain, S., Simon, H.U., Tomita, E. (eds.) ALT 2005. LNCS (LNAI), vol. 3734, pp. 171–182. Springer, Heidelberg (2005)
[11] Hsu, D., Kakade, S.M., Zhang, T.: A spectral algorithm for learning hidden markov models. CoRR abs/0811.4413 (2008)
[12] Kearns, M.: Efficient noise-tolerant learning from statistical queries. J. ACM 45(6), 983–1006 (1998)
[13] Kearns, M.J., Mansour, Y., Ron, D., Rubinfeld, R., Schapire, R.E., Sellie, L.: On the learnability of discrete distributions. In: STOC, pp. 273–282 (1994)
[14] Lyngsø, R.B., Pedersen, C.N.S.: The consensus string problem and the complexity of comparing hidden markov models. J. Comput. Syst. Sci. 65(3), 545–569 (2002)
[15] Palmer, N., Goldberg, P.W.: PAC-learnability of probabilistic deterministic finite state automata in terms of variation distance. Theor. Comput. Sci. 387(1), 18–31 (2007)
[16] Ron, D., Singer, Y., Tishby, N.: On the learnability and usage of acyclic probabilistic finite automata. J. Comput. Syst. Sci. 56(2), 133–152 (1998)
[17] Vidal, E., Thollard, F., de la Higuera, C., Casacuberta, F., Carrasco, R.C.: Probabilistic finite-state machines - part I. IEEE Trans. Pattern Anal. Mach. Intell. 27(7), 1013–1025 (2005)
[18] Vidal, E., Thollard, F., de la Higuera, C., Casacuberta, F., Carrasco, R.C.: Probabilistic finite-state machines - part II. IEEE Trans. Pattern Anal. Mach. Intell. 27(7), 1026–1039 (2005)