On the Learnability of Hidden Markov Models

Sebastiaan A. Terwijn*

Vrije Universiteit Amsterdam, Department of Mathematics and Computer Science,
De Boelelaan 1081a, 1081 HV Amsterdam, the Netherlands,
terwijn@cs.vu.nl

Abstract. A simple result is presented that links the learning of hidden Markov models to results in complexity theory about nonlearnability of finite automata under certain cryptographic assumptions. Rather than considering all probability distributions, or even just certain specific ones, the learning of a hidden Markov model takes place under a distribution induced by the model itself.

Keywords. Pac-learning, hidden Markov models, complexity.

1 Introduction

Hidden Markov models (HMM's for short) are a widely used tool for describing processes, ranging from speech recognition to topics in computational biology. In many situations where a process can be described by a HMM, one tries to infer from data produced by the process (e.g. spoken text or products from a chemical reaction) a HMM describing that process. In this paper we address the computational complexity of such general learning tasks. We will use Valiants [13] model of pac-learning to talk about efficient learning (see [1,10] for an introduction to this model). In the early 1990's, interesting connections were proven between assumptions on which the security of public-key cryptography systems is based and hardness results for pac-learning various natural classes. E.g., Kearns and Valiant [9] proved that under the Discrete Cube Root Assumption, saying roughly that it is impossible to efficiently invert functions that occur in the RSA cryptosystem, the class of polynomial size finite automata is not efficiently pac-learnable. More precisely, let $\mathrm{ADFA}_n^{p(n)}$ denote the class of finite automata of size $p(n)$ that accept only words of length n. (ADFA stands for acyclic deterministic finite automata.) Then (under suitable cryptographic assumptions) there is a polynomial p such that $\mathrm{ADFA}_n^{p(n)}$ is not efficiently pac-learnable. Pac-learnability can be characterized in terms of a measure of complexity of the concept space, the Vapnik-Chervonenkis dimension (VC-dimension), see Blumer et al. [4]. A concept class of VC-dimension d can be pac-learned by an algorithm taking samples of size linear in d. So when for a parametrized concept

* Supported by a Marie Curie fellowship of the European Union under grant no. ERB-FMBI-CT98-3248. Most of this research was done while the author was working at the University of Munich.

P. Adriaans, H. Fernau, and M. van Zaanen (Eds.): ICGI 2002, LNAI 2484, pp. 261–268, 2002.

class $\mathcal{C} = \bigcup_n \mathcal{C}_n$ the VC-dimension of $\mathcal{C}_n$ is polynomial in n, the class can be pac-learned using polynomial size samples. If such a class $\mathcal{C}$ does not admit an efficient pac-algorithm, irrelevant of how hypotheses are represented (as long as they are polynomial time evaluatable), it is called *inherently unpredictable.* Since the class $\mathrm{ADFA}_n^{p(n)}$ has polynomial VC-dimension the result of Kearns and Valiant shows that it is inherently unpredictable.

Below we will consider in an analogous way classes of polynomial size acyclic HMM's, and link a nonlearnability result for finite automata of Kharitonov to a specific task of learning HMM's. The problem of learning HMM's is defined not in terms of arbitrary probability distributions (as is usual in the pac model, which is therefore sometimes referred to as "distribution free") but in terms of the distributions induced by the HMM's themselves. So we consider a distribution specific learning problem, where every HMM has to be identified from examples generated according to its own likelihood distribution.

We now describe the nonlearnability result for finite automata that we will use. Blum integers are integers of the form $p \cdot q$, where p and q are primes congruent to 3 modulo 4. The cryptographic assumption used in the result is that *factoring Blum integers is hard.* We refer to Kearns [8, p 105-108] for a precise statement and a discussion of this assumption.

The following result follows from Kharitonovs results in [11] using the reductions from Pitt and Warmuth [12], see also [10].

Theorem 1. (Kharitonov [11]) *Assuming that factoring Blum integers is hard, there is a polynomial p such that the class $\mathrm{ADFA}_n^{p(n)}$ is inherently unpredictable under the uniform distribution.*

2 Hidden Markov Models

In this section we discuss hidden Markov models (HMM's). HMM's can be used to describe mechanisms in a diversity of fields. A HMM can be thought of as a probabilistic automaton with a finite number of states, with fixed probabilities for going from one state to another, and in which for every state there are probabilities for outputting certain symbols. Usually, we only see these output symbols, and the states are hidden from us. Over time, the HMM thus generates an infinite sequence of output symbols, adding a symbol at every time step. In the next formal definition we use the notation of Clote and Backofen [6]. It will be convenient to work with the alphabet $\Sigma = \{0, 1\}$.

Definition 1. A hidden Markov model (HMM) is a tuple $M = (Q, \pi, a, b)$, where $Q = 1, \dots, m$ is a set of states, π is a vector of initial state probabilities, a is a matrix of transition probabilities, and b is a matrix of output probabilities:

$$\begin{aligned} \pi_i &= \Pr[q_0 = i] \\ a_{i,j} &= \Pr[q_t = j | q_{t-1} = i] \\ b_{i,k} &= \Pr[o_t = k | q_t = i], \quad k \in \{0, 1\} \end{aligned}$$

such that $\sum_{i=1}^{m} \pi_i = 1$, and for every i, $\sum_{j=1}^{m} a_{i,j} = 1$ and $b_{i,0} + b_{i,1} = 1$. Here q_0 is the initial state, and q_t is the state of the system at time t. o_t is the symbol that is output when M is in state q_t. The *likelihood* $L_w(M)$ of a binary word w is defined as the probability that M generates w, i.e. the probability that for every $t < |w|$, in state q_t the output symbol o_t is equal to the t-th bit w_t of w. The likelihood of a sample S is defined as $L_S(M) = \prod_{w \in S} L_w(M)$. The *size* of a HMM M is defined in a straightforward way as $\mathrm{size}(\pi) + \mathrm{size}(a) + \mathrm{size}(b)$. We also allow HMM's to have final states, i.e. states without outgoing transitions. For a polynomial p, denote the class of HMM's of size $p(n)$ that generate only strings of length n by $\mathrm{HMM}_n^{p(n)}$.

We prove two easy lemmas for later reference.

Lemma 1. *Let $T \in \mathbb{N}$ and let $p \in Q^T$ be a fixed path (i.e. p is a sequence of T states in Q). Then $\sum_{|w|=T} \Pr[w|p, M] = 1$.*

Proof. Note that $\sum_{|w|=T} \Pr[w|p, M] = \sum_{|w|=T} \prod_{i=0}^{T-1} b_{p(i),w_i}$. We prove the lemma by induction on T.

$T = 1$: $b_{p(0),0} + b_{p(0),1} = 1$.

$T + 1$:

$$\begin{aligned}\sum_{|w|=T+1} \Pr[w|p, M] &= \sum_{|w|=T} \Pr[w|p\restriction T, M] b_{p(T),0} + \\ &\quad \sum_{|w|=T} \Pr[w|p\restriction T, M] b_{p(T),1} \\ &= b_{p(T),0} + b_{p(T),1} = 1,\end{aligned}$$

where the second equality follows by the induction hypothesis. □

Lemma 2. $\sum_{p \in Q^T} \Pr[p|M] = 1$.

Proof. Induction on T.

$T = 1$: $\sum_{p \in Q^T} \Pr[p|M] = \sum_{p \in Q} \pi(p) = 1$.

$T + 1$:

$$\begin{aligned}\sum_{p \in Q^{T+1}} \Pr[p|M] &= \sum_{p \in Q^T} \left(\Pr[p|M] \sum_{q \in Q} a_{p(T-1),q} \right) \\ &= \sum_{p \in Q^T} \Pr[p|M],\end{aligned}$$

and this last expression equals 1 by induction hypothesis. □

Every deterministic finite automaton (DFA) G can be represented by a HMM M as follows. Split every state q in G into two states q_0 and q_1. We want to make sure that after every transition labeled with 0 with certainty an output bit 0 in M is output, so for every 0-transition to q in G we make a corresponding transition to q_0 in M, and we give q_0 output probability 1 for 0 and 0 for 1. Similarly, for every 1-transition to q in G we make a corresponding transition to q_1 in M, and we give q_1 output probability 1 for 1 and 0 for 0. If a state in M has two outgoing transitions we give both probability 1/2, and if it has one it gets probability 1. The start state of G is removed and replaced by suitable initial probabilities in M.

For M defined in this way we have $w \in L(G) \Leftrightarrow L_w(M) > 0$. (Here $L(G)$ is the language associated with G, i.e. the set of words accepted by G.) When we identify H with the set $\{w : L_w(M) > 0\}$ then we see that M represents G. We note that the likelihood $L_w(M)$ is computable in polynomial time by the forward method [5,6]. Since the class of DFA's is inherently unpredictable, irrelevant of which polynomially evaluatable hypotheses representation is used, it follows immediately from the nonlearnability results for finite automata quoted in Section 1 that the class of polynomial size HMM's is not efficiently pac-learnable:

Fact 2 *The class of polynomial size HMM's has polynomial VC-dimension. Hence (under the Discrete Cube Root Assumption) it is inherently unpredictable.*

Below we will prove that a much more specific learning task is not feasible (assuming that factoring Blum integers is hard).

3 Learning Hidden Markov Models

We can associate several learning tasks with the class of HMM's. The first well-known task is to maximize the likelihood:

HMM Likelihood Problem. Given n, the number of states in an unknown HMM M, and a sample S of observation sequences generated by M, determine a HMM H on n states such that $L_S(H)$ is maximal.

The most widely used method to attack the HMM Likelihood Problem is the Baum-Welch method [3], also called the forward-backward method. This algorithm adjusts by iteration the parameters of a given HMM so as to increase the likelihood of an observation sequence. The algorithm works in polynomial time, i.e. every iteration step can be computed efficiently, but no general analysis of its rate of convergence is known.

For a HMM M, define

$$\mathcal{D}_M(w) = L_w(M).$$

For any fixed length n, $\mathcal{D}_M$ is a probability distribution on Σ^n, as the next proposition shows. We will call $\mathcal{D}_M$ the *probability distribution induced by M*.

Proposition 1. *For any n, $\sum_{|w|=n} \mathcal{D}_M(w) = 1$.*

Proof. Using the Lemmas 1 and 2 we have

$$\begin{aligned}\sum_{|w|=T} \mathcal{D}_M(w) &= \sum_{|w|=T} L_w(M)\\ &= \sum_{|w|=T} \sum_{p \in Q^T} \Pr[w|p, M] \Pr[p|M]\\ &= \sum_{p \in Q^T} \Pr[p|M] \sum_{|w|=T} \Pr[w|p, M]\\ &= \sum_{p \in Q^T} \Pr[p|M]\\ &= 1.\end{aligned}$$

□

HMM Learning Problem. Let p be a fixed polynomial and let $\mathcal{U}$ be the uniform distribution on Σ^n. Given n and $p(n)$, the number of states in an unknown HMM M, $\varepsilon, \delta < 1/2$, and a sample $S \subseteq \Sigma^n$ of observation sequences of length n generated by M, determine a HMM H on $p(n)$ states such that with probability $1-\delta$, error$(H) < \varepsilon$. Here error$(H) := \Pr_{\mathcal{D}_M}[M-H] + \Pr_{\mathcal{U}}[H-M]$. Here $M-H$ denotes the set of strings in Σ^n that are generated by M but not by H.

Note that this is more specific than in the general pac-model, since each target concept has its own probability distribution, namely the one that it induces. This seems to reflect practical learning situations in a more natural way. After all, when we see the products of some unknown mechanism, we are most likely to see those products that the mechanism is most likely to generate. A solution to the HMM Learning Problem requires a (possibly randomized) algorithm that pac-learns any HMM M on the basis of positive examples only, which are sampled according to the induced probability distribution $\mathcal{D}_M$. The error is measured according to $\mathcal{D}_M$ on the positive examples and to the uniform distribution $\mathcal{U}$ on the negative examples. We note that, if the choice of $\mathcal{U}$ for the negative examples seems too arbitrary, it follows from the results of Kharitonov that we could have chosen any other nontrivial distribution here (see [11] for the definition of 'nontrivial'). The next theorem shows that, although the HMM Learning Problem is much more specific than the general problem of pac-learning HMM's, this problem is still not feasible. That an algorithm L for the HMM Learning Problem runs in polynomial time means that there is a polynomial p such that L runs in time $p(n, 1/\varepsilon, 1/\delta)$, where n is the size of the target concept, and ε and δ are the error and confidence parameter, respectively. Recall that $\mathrm{HMM}_n^{p(n)}$ denotes the class of HMM's of size $p(n)$ that generate only strings of length n.

Theorem 3. *Assuming that factoring Blum integers is hard, the HMM Learning Problem is not solvable in polynomial time. (That is, there is a polynomial p such that the HMM learning problem for* $\mathrm{HMM}_n^{p(n)}$ *is not solvable in polynomial time.)*

Before proving Theorem 3 we prove two auxiliary results.

Theorem 4. *Let $\mathcal{C} = \bigcup_n \mathcal{C}_n$ be a graded concept space, and suppose $\mathcal{C}_n$ is not efficiently pac-learnable under the uniform distribution $\mathcal{U}$ over Σ^n (for both the positive and the negative examples). Then $\mathcal{C}_n$ is also not efficiently pac-learnable when we use for every $c \in \mathcal{C}_n$ the distribution $\mathcal{D}_c$ defined by*

$$\mathcal{D}_c(w) = \begin{cases} 0 & \textit{if } w \notin c \\ \frac{1}{||c||} & \textit{if } w \in c. \end{cases}$$

($||c||$ is the cardinality of c), and the error of an hypothesis h is defined as $\Pr_{\mathcal{D}_c}[c-h] + \Pr_{\mathcal{U}}[h-c]$.

Proof. Suppose L is an algorithm that pac-learns every $c \in \mathcal{C}$ under the distribution $\mathcal{D}_c$, with error defined as in the theorem, and that L works in time p for some polynomial p. Then we can pac-learn $\mathcal{C}$ under the uniform distribution as follows. Let ε and δ be the given error and confidence parameters. We sample under the uniform distribution and simulate L on the positive examples. We need $p(n, 1/\delta, 1/\varepsilon)$ hits from c to do this. We show that it is enough to sample $m = (2/\varepsilon)(4\log(1/\delta) + p(n, 1/\delta, 1/\varepsilon))$ examples under the uniform distribution. There are two cases.
Case 1. The weight of c is very small: $\Pr_{\mathcal{U}}[c] < \varepsilon$, and among the m examples there are less than $p(n, 1/\delta, 1/\varepsilon)$ positive ones. Output $\emptyset$. Since $\Pr_{\mathcal{U}}[c] < \varepsilon$ the error is less than ε.
Case 2. $\Pr_{\mathcal{U}}[c] \geq \varepsilon$. When we view the sampling of the m examples as m independent Bernoulli trials, each with success of hitting c at least ε, a suitable Chernoff bound[1] gives us that

$$\begin{aligned} \Pr[\#\text{ hits in } c < p(n, 1/\delta, 1/\varepsilon)] &\leq \Pr[\#\text{ hits in } c < \frac{1}{2}\varepsilon m] \\ &\leq e^{-\varepsilon m/8} \\ &\leq e^{-\frac{\varepsilon}{8}\frac{2}{\varepsilon}(4\log\frac{1}{\delta})} \\ &= \delta. \end{aligned}$$

So with probability at least $1-\delta$, at least $p(n, 1/\delta, 1/\varepsilon)$ of the m examples are labeled positively, and we can simulate L on these examples. This gives with probability at least $1-\delta$ an hypothesis h with $\Pr_{\mathcal{D}_c}[c-h] + \Pr_{\mathcal{U}}[h-c] < \varepsilon$. Since $\Pr_{\mathcal{D}_c}[c-h] \geq \Pr_{\mathcal{U}}[c-h]$ we have in total that $\Pr_{\mathcal{U}}[(h-c) \cup (c-h)] < \varepsilon$ with probability at least $(1-\delta)^2$. This shows that $\mathcal{C}$ is pac-learnable under the uniform distribution, since we could have set the confidence parameter δ to δ^2 at the beginning of the proof, thus obtaining the good hypothesis h with confidence at least $1-\delta$. □

Theorem 5. *For every G in $\mathrm{ADFA}_n^{p(n)}$ there is an M in $\mathrm{HMM}_n^{O(p(n))}$ such that for every $w \in \Sigma^n$ it holds that $w \in L(G) \Leftrightarrow L_w(M) > 0$ and $L_w(M) > 0 \Rightarrow L_w(M) = 1/\|L(G)\|$.*

[1] If S is the number of successes in a series of m independent Bernoulli trials with probability of success p, then we use that $\Pr[S < (1-\gamma)pm] \leq e^{-mp\gamma^2/2}$ for any $0 \leq \gamma \leq 1$.

Proof. Define M from G as in the discussion preceding Fact 2, except for the transition probabilities in M which are defined as follows. Without loss of generality, all paths in G end in an accepting state. We assign to every transition in M going from node q to node q' the probability k/l, where l is the total number of *transitions* that can be followed from q (i.e. its outdegree) and k is the total number of *paths* that can be followed from q'. By induction on the length of the paths, i.e. induction on n, one can check that in this way every path in M defined from a path in G gets the same nonzero probability, and all other paths get probability zero. It is easy to check that M satisfies the statement of the theorem. □

Proof of Theorem 3. Suppose algorithm L solves the HMM Learning Problem in polynomial time. Given a finite automaton G in $\mathrm{ADFA}_n^{p(n)}$, and ε, δ, let M be as in Theorem 5. Then L outputs in polynomial time and with probability $1-\delta$ an hypothesis H with $\mathrm{Pr}_{\mathcal{D}_M}(M-H)+\mathrm{Pr}_{\mathcal{U}}(H-M)<\varepsilon$. But then, since $\mathcal{D}_M$ is uniform on M, by Theorem 4, we can pac-learn M under the uniform distribution. Now given H and $w \in \Sigma^n$, one can compute in time $O(np(n)^2)$ whether $L_w(H)>0$ using the forward method (see e.g. [5,6]). This means that, for any polynomial p, we can pac-learn $\mathrm{ADFA}_n^{p(n)}$ under the uniform distribution using the polynomially evaluatable hypothesis class $\mathrm{HMM}_n^{O(p(n))}$. But this contradicts Theorem 1. □

Note that all HMM's used in the proof are polynomial size, so that (by Fact 2) we can say that the nonlearnability is not caused by a nonpolynomial VC-dimension of the class of HMM's under consideration. Also note that we have in fact proved a result about stochastic DFA's (stochastic regular grammars).

Acknowledgment. We thank Peter Clote for introducing us to hidden Markov models and for helpful discussions.

References

1. M. Anthony and N. Biggs, *Computational learning theory: an introduction*, Cambridge University Press, 1992.
2. P. Baldi and Y. Chauvin, *Smooth on-line learning algorithms for hidden Markov models*, Neural Computation 6 (1994) 307–318.
3. L. E. Baum, *An inequality and associated maximization technique in statistical estimation for probabilistic functions of a Markov process*, Inequalities 3, 1972, 1–8.
4. A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth, *Learnability and the Vapnik-Chervonenkis dimension*, J. of the ACM 36(4), 929–965, 1989.
5. E. Charniak, *Statistical Language Learning*, MIT Press, 1993.
6. P. Clote and R. Backofen, *Computational molecular biology: an introduction*, Wiley, 2000.
7. O. Goldreich, S. Goldwasser, and S. Micali, *How to construct random functions*, J. of the ACM 33(4) (1986) 792–807.

8. M. J. Kearns, *The computational complexity of machine learning*, MIT Press, 1990.
9. M. J. Kearns and L. G. Valiant, *Cryptographic limitations on learning boolean formulae and finite automata*, J. of the ACM 41(1) (1994) 67–95.
10. M. J. Kearns, U. V. Vazirani, *An introduction to computational learning theory*, MIT Press, 1994.
11. M. Kharitonov, *Cryptographic hardness of distribution-specific learning*, Proc. 25th ACM Symp. on the Theory of Computing, 372–381, ACM Press, N.Y., 1993.
12. L. Pitt and M. K. Warmuth, *Prediction-preserving reducibility*, J. Computer and System Sci. 41(3) (1990) 430–467.
13. L. G. Valiant, *A theory of the learnable*, Communications of the ACM 27(11) (1984) 1134–1142.