Hence from (4.5) and (4.10), it follows that whenever $m \geq max\{q(\frac{4}{\epsilon}, \log\frac{2}{\delta}, n, t), \log\frac{2}{\delta}\}$, with probability at least $1-\delta$,

$$E_D(\log\frac{1}{Q_B(\cdot)}) - E_D(\log\frac{1}{Opt(\cdot)}) \leq \frac{\epsilon}{2} + \frac{5(n+1)}{m-1} \tag{4.11}$$

Hence, whenever $m \geq max\{q(\frac{\epsilon}{4}, \log\frac{2}{\delta}, n, t), \frac{10(n+1)}{\epsilon} + 1, \log\frac{2}{\delta}\}$, with probability at least $1-\delta$,

$$E_D(\log\frac{1}{Q_B(\cdot)}) - E_D(\log\frac{1}{Opt(\cdot)}) \leq \epsilon$$

We have thus shown that the above algorithm trains C, which was an arbitrary constraint in $\mathcal{C}$, with sample complexity polynomial in $\frac{1}{\epsilon}, \log\frac{1}{\delta}, n, t$ and running time polynomial in the same parameters. It is easy to convert such an algorithm to one whose running time is polynomial in the total sample length and which has sample complexity still polynomial in $\frac{1}{\epsilon}, \log\frac{1}{\delta}, n, t$.

$(2 \rightarrow 1)$

This is obvious from the definitions.

Thus far, we have shown that the first three statements of the theorem are equivalent. Now we proceed to show that *3* is equivalent to *4*.

$(4 \rightarrow 3)$ Suppose that algorithm A, running in random time polynomial in $\frac{1}{\epsilon}, n, t, m$, approximates the sample MLM problem for $\mathcal{C}$ within factor $2^{p(n,t)m^\alpha}$ for some polynomial p. Let a constraint $C \in \mathcal{C}$ and a sample $S = \langle w_1, ..., w_m\rangle$ be given. Again, assume without loss of generality that $\prod_{i=1}^m OPT_S(w_i) = max\{\prod_{i=1}^m P(w_i) : P \in \mathcal{PA}(C)\}$ is positive. We then repeat the sample $r = \lceil(\frac{2}{\epsilon}(\ln 2)p(n,t)m^\alpha)^{\frac{1}{1-\alpha}}\rceil$ times to obtain a new sample $T = \langle v_1, ..., v_{rm}\rangle$ of length rm, and feed this into A to obtain a hypothesis Q. Then, by definition, Q must satisfy:

$$\frac{\prod_{i=1}^{rm} OPT_T(t_i)}{\prod_{i=1}^{rm} Q(t_i)} \leq 2^{p(n,t)(rm)^\alpha}$$

where we used OPT_T to denote the PA in $\mathcal{PA}(C)$ that assigns the maximum probability on the sample T. Since each example of S is repeated exactly r times in T,

$$\begin{aligned}\frac{\prod_{i=1}^m OPT_T(w_i)}{\prod_{i=1}^m Q(w_i)} &\leq (2^{p(n,t)(mr)^\alpha})^{\frac{1}{r}} \\ &\leq 2^{p(n,t)m^\alpha r^{\alpha-1}}\end{aligned} \tag{4.12}$$

By substituting r into the exponent in (4.12) we obtain:

$$\begin{aligned}\log\frac{\prod_{i=1}^m OPT_T(w_i)}{\prod_{i=1}^m Q(w_i)} &\leq p(n,t)m^\alpha\lceil(\frac{2}{\epsilon}(\ln 2)p(n,t)m^\alpha)^{\frac{1}{1-\alpha}}\rceil^{\alpha-1} \\ &\leq p(n,t)m^\alpha(\frac{2}{\epsilon}(\ln 2)p(n,t)m^\alpha)^{\frac{1}{1-\alpha}})^{\alpha-1}, \text{ since } \alpha - 1 < 0 \\ &= p(n,t)m^\alpha(\frac{2}{\epsilon}(\ln 2)p(n,t)m^\alpha)^{-1} \\ &= \frac{\epsilon}{2}\log e\end{aligned}$$

Hence, we have:

Now, using the trick of shifting, rounding-off and stochastic correction as done in the proof of Theorem 3.1, we obtain from P a member Q_B of $\mathcal{FPA}(C)_m$, such that

$$\forall x \in \Sigma^n \ \log \frac{1}{Q_B(x)} - \log \frac{1}{P(x)} \leq \frac{2(n+1)}{m-1} \tag{4.4}$$

From (4.3) and (4.4), (4.2) follows.

Next we can show that any algorithm having the above performance indeed trains $\mathcal{C}$. This can be shown similarly to the way the algorithm exhibited in the proof of Theorem 3.1 was proved to successfully train $\mathcal{C}$. Let us first recall the notation of that proof: We denoted by Q the output of the training algorithm which, for the input sample S, minimized $E_{\widehat{D}_S}(\log \frac{1}{F(\cdot)})$ over $F \in \mathcal{FPA}(C)_m$. The PA Opt was a member of $\mathcal{PA}$ with the minimum divergence with respect to the target distribution. F^* denoted a member of $\mathcal{FPA}(C)_m$ that is closest to Opt. In order to prove that $E_D(\log \frac{1}{Q(\cdot)})$ is close to $E_D(\log \frac{1}{Opt(\cdot)})$ with high probability (inequality (3.9)), we showed in the proof of Theorem 3.1 that:

$$E_D(\log \frac{1}{F^*(\cdot)}) - E_D(\log \frac{1}{Opt(\cdot)}) \leq \frac{2(n+1)}{m-1} \tag{4.5}$$

and whenever $m \geq q(\frac{4}{\epsilon}, \frac{1}{\delta}, n, t)$ with probability at least $1-\delta$,

$$E_D(\log \frac{1}{Q(\cdot)}) - E_D(\log \frac{1}{F^*(\cdot)}) \leq \frac{\epsilon}{2} \tag{4.6}$$

Since (4.5) always holds, we only need to show an analogue of (4.6) for any algorithm satisfying (4.2). Recall that in the earlier proof, in order to show (3.9), we applied triangle inequality to the inequalities (3.6), (3.7), and (3.8). (3.6) and (3.7) followed from the uniform convergence for $\mathcal{FPA}(C)_m$, and (3.8) followed from the fact that the training algorithm minimized $E_{\widehat{D}_S}(\log \frac{1}{Q(\cdot)})$ within $\mathcal{FPA}(C)_m$.

In this proof, the output Q_B of our training algorithm *approximately* minimizes $E_{\widehat{D}_S}(\log \frac{1}{Q(\cdot)})$ within $\mathcal{FPA}(C)_m$. More precisely, by the optimality of OPT, it follows from (4.2) that the following holds with probability at least $1-(\frac{1}{2})^m$:

$$E_{\widehat{D}_S}(\log \frac{1}{Q_B(\cdot)}) - E_{\widehat{D}_S}(\log \frac{1}{F^*(\cdot)}) \leq \frac{3(n+1)}{m-1} \tag{4.7}$$

As before, for $m \geq q(\frac{4}{\epsilon}, \log \frac{2}{\delta}, n, t)$ both of the following hold with probability at least $1-\frac{\delta}{2}$:

$$E_D(\log \frac{1}{Q_B(\cdot)}) - E_{\widehat{D}_S}(\log \frac{1}{Q_B(\cdot)}) \leq \frac{\epsilon}{4} \tag{4.8}$$

$$E_{\widehat{D}_S}(\log \frac{1}{F^*(\cdot)}) - E_D(\log \frac{1}{F^*(\cdot)}) \leq \frac{\epsilon}{4} \tag{4.9}$$

By summing up the inequalities (4.7), (4.8), and (4.9), and noting that $(\frac{1}{2})^m \leq \frac{\delta}{2}$ when $m \geq \log \frac{2}{\delta}$, we obtain that the following holds with probability at least $1-\delta$ whenever $m \geq max\{q(\frac{4}{\epsilon}, \log \frac{2}{\delta}, n, t), \log \frac{2}{\delta}\}$:

$$E_D(\log \frac{1}{Q_B(\cdot)}) - E_D(\log \frac{1}{F^*(\cdot)}) \leq \frac{\epsilon}{2} + \frac{3(n+1)}{m-1} \tag{4.10}$$

We thus obtain $\sum_{i=1}^{m} \log \frac{OPT(w_i)}{Q(w_i)} \leq \frac{\epsilon}{2}$. So for $\epsilon \in (0,1]$, we have

$$\prod_{i=1}^{m} \frac{OPT(w_i)}{Q(w_i)} \leq 2^{\frac{\epsilon}{2}} \leq 1+\epsilon$$

Since OPT is a probabilistic automaton in $\mathcal{PA}(C)$ minimizing the divergence with respect to $\widehat{D}_S$ on S, by the 'classical equivalence' given in Lemma 2.1, it is also an optimal solution in $\mathcal{PA}(C)$ of the sample MLM problem on S. Thus we have obtained a $1+\epsilon$ approximation to the sample MLM problem with probability at least a half. The running time of B is clearly bounded by a polynomial in $\frac{1}{\epsilon}, n, t, m$, since m' is polynomial in these parameters, the time spent on generating the sample S' using the algorithm U is polynomial in n, m and m', and the running time of A is polynomial in the total length of the sample S', i.e. polynomial in m' and n.

$(3 \rightarrow 2)$

We will use the hypothetical approximation algorithm for the MLM problem for a class of constraints $\mathcal{C}$ to construct B that trains $\mathcal{C}$. Let D be the target distribution over Σ^n and let a constraint $C \in \mathcal{C}$ be given as input. Assume without loss of generality that $min\{d_{KL}(D,P) : P \in \mathcal{PA}(C)\}$ is finite, since otherwise the training problem is trivial. The finiteness of $min\{d_{KL}(D,P) : P \in \mathcal{PA}(C)\}$ guarantees that for each $x \in \Sigma^n$ assigned a positive probability by D, there is a path in C labeled with x. Hence it follows that for an arbitrary sample S generated by D, $max\{\prod_{i=1}^{m} P(w_i) : P \in \mathcal{PA}(C)\}$ is positive. First we show that any approximation algorithm (in the sense of *3* in the statement of the lemma) can be modified so as to output a PA Q_B in the finite class $\mathcal{FPA}(C)_m$, defined for C as in the proof of Theorem 3.1, which with probability at least $1-(\frac{1}{2})^m$ satisfies:

$$E_{\widehat{D}_S}(\log \frac{1}{Q_B(\cdot)}) - E_{\widehat{D}_S}(\log \frac{1}{OPT(\cdot)}) \leq \frac{3(n+1)}{m-1} \tag{4.2}$$

where $OPT \in \mathcal{PA}(C)$ is an optimal solution for the sample MLM problem with input sample S satisfying the input constraint C. Second, we show that in fact such an algorithm is a training algorithm for PAs with a slightly larger sample complexity than the training algorithm exhibited in the proof of Theorem 3.1.

It is easy to verify the first of these two claims. First note that any algorithm which approximates the sample MLM problem for $\mathcal{C}$ within factor $1+\epsilon$ with probability at least a half in time polynomial in $\frac{1}{\epsilon}, n, t$ and m can be boosted to one which achieves the same approximation factor with probability at least $1-\delta$ in time polynomial in $\log \frac{1}{\delta}, \frac{1}{\epsilon}, t, n$ and m. This can be done by iteratively running the former algorithm $\lceil \log \frac{1}{\delta} \rceil$ times and then selecting, from among its outputs, one that assigns the maximum likelihood on the input sample. (Note that using dynamic programming it is easy to compute for a given sample and PA the likelihood of that sample in time polynomial in the total length of the sample and t.) Now set $\epsilon = n$ and $\delta = (\frac{1}{2})^m$ and run this algorithm on the input sample S and obtain a PA P, which with probability at least $1-(\frac{1}{2})^m$ approximates the sample MLM problem for $\mathcal{C}$ within factor $1+n$:

$$\prod_{i=1}^{m} \frac{OPT(w_i)}{P(w_i)} \leq 1+n \leq e^{1+n}$$

Hence we have:

$$E_{\widehat{D}_S}(\log \frac{1}{P(\cdot)}) - E_{\widehat{D}_S}(\log \frac{1}{OPT(\cdot)}) = \frac{1}{m}\sum_{i=1}^{m} \log \frac{OPT(w_i)}{P(w_i)} \leq \frac{n+1}{m} < \frac{n+1}{m-1} \tag{4.3}$$

Let $\epsilon > 0$, a finite sample $S = \langle w_1, ..., w_m \rangle$ with $|w_i| = n$ for each w_i, and a constraint $C \in \mathcal{C}$ be given. We use the hypothetical training algorithm A on the empirical distribution $\widehat{D}_S$ of S, as defined in Definition 2.1. We can assume without loss of generality that $min\{d_{KL}(\widehat{D}_S, P) : P \in \mathcal{PA}(C)\}$ is finite, since otherwise $max\{\prod_{i=1}^{m} P(w_i) : P \in \mathcal{PA}(C)\} = 0$ and the MLM problem is trivial. B then computes a sample size m' for A large enough for accuracy $\frac{\epsilon}{2m}$ and confidence $1 - \frac{1}{4}$. For example, $m' = \lceil q(\frac{2m}{\epsilon}, 4, n, t) \rceil$ suffices where $q(...)$ is an upper bound on the sample complexity of the hypothetical training algorithm A, which by assumption is polynomially bounded. Now, B gives A a sample S' of size m' obtained by sampling from S according to $\widehat{D}_S$, the empirical distribution of S, or the uniform distribution over the m elements of the sequence S. Here note that while it is not always possible to simulate the empirical distribution of a finite sample with a fair coin, there exists an algorithm which generates a new sample of an arbitrary size according to the empirical distribution of the original sample with high probability. We make this precise below.

Lemma 4.1: *There exists a randomized algorithm, U, which, given a finite sample $S = \{w_1, ..., w_m\}$ of size m, an integer m', and a confidence parameter $\nu > 0$, always terminates in time polynomial in m, m' and $\frac{1}{\nu}$ and outputs a sample S' of size m' which with probability at least $1 - \nu$ is drawn according to the empirical distribution $\widehat{D}_S$ of S.*

Proof of Lemma 4.1 The algorithm U first calculates $i = \lceil \log m \rceil$ and iterates the following: U flips a fair coin i times to obtain a bit string x of length i. If $x \leq m$, then U appends w_x to the end of the sequence S', and does nothing otherwise. It is clear that each example of S' is drawn independently at random according to $\widehat{D}_S$. Notice that at each iteration the length of S' increases by one with probability at least a half. It is easy to see, by applying Chernoff's bound (c.f. [Val84]) that in $p(m', \frac{1}{\nu})$ many iterations, the length of S' becomes m' with probability at least $1 - \nu$, where p is a certain polynomial. If this fails to occur, i.e. the length of S' is shorter than m' after $p(m', \frac{1}{\nu})$ many iterations, U pads S' with arbitrary examples to make its length m'.

End of proof of Lemma 4.1

We run U on S, m', and $\frac{1}{3}$ to obtain a sample S', and let Q be the output obtained by running A on S'. Thus, A is run on a sample of size m' generated with respect to $\widehat{D}_S$ with probability at least $\frac{2}{3}$. Furthermore, when this is the case, by the performance guarantee on A, the divergence of Q with respect to $\widehat{D}_S$ is $\frac{\epsilon}{2m}$ close to the divergence of the best PA satisfying C, with probability at least $\frac{3}{4}$. Hence, with probability at least $\frac{3}{4} \cdot \frac{2}{3} = \frac{1}{2}$, the following holds:

$$d_{KL}(\widehat{D}_S, P) - d_{KL}(\widehat{D}_S, OPT) = \sum_{x \in \Sigma^n} \widehat{D}_S(x) \log \frac{OPT(x)}{P(x)} \leq \frac{\epsilon}{2m} \tag{4.1}$$

where OPT is a PA in $\mathcal{PA}(C)$ satisfying:

$$d_{KL}(\widehat{D}_S, OPT) = \min\{d_{KL}(\widehat{D}_S, P) : P \in \mathcal{PA}(C)\}$$

Substituting $\widehat{D}_S(x) = \frac{\sharp(x,S)}{m}$ in (4.1) above,

$$d_{KL}(\widehat{D}_S, Q) - d_{KL}(\widehat{D}_S, OPT) = \frac{1}{m} \sum_{i=1}^{m} \log \frac{OPT(w_i)}{Q(w_i)} \leq \frac{\epsilon}{2m}$$

4 The Equivalence Between the Training and Maximum Likelihood Model Problems

The sample complexity bounds of Theorem 3.1 can be used to establish the equivalence between the efficient trainability of a class of constraints and the efficient approximability of the sample MLM problem for the same class. We first define what it means for a randomized algorithm to *approximate* the sample MLM problem within a given factor.

Definition 4.1 (Approximate Sample MLM problem): *A randomized algorithm A is said to approximate the sample MLM problem for a class of constraints $\mathcal{C}$ within factor K, possibly a function of various parameters of the problem, in random $T(...)$ time, if given a constraint $C \in \mathcal{C}$, any $\epsilon > 0$, and an input sample $S = \langle w_1, ..., w_m \rangle$ of Σ^n for some $n > 0$ and some finite alphabet Σ, A terminates in $T(...)$ many steps and outputs a PA $Q \in \mathcal{PA}(C)$, which with probability at least a half satisfies:*

$$\frac{\prod_{i=1}^{m} OPT(w_i)}{\prod_{i=1}^{m} Q(w_i)} \leq K$$

where OPT is a member of $\mathcal{PA}(C)$ which maximizes the likelihood of S, i.e.

$$\prod_{i=1}^{m} OPT(w_i) = max\{\prod_{i=1}^{m} P(w_i) : P \in \mathcal{PA}(C)\}$$

By convention, we let $\frac{0}{0} = 1$

As before, OPT is guaranteed to exist because of the compactness of $\mathcal{PA}(C)$ and the continuity of the likelihood function on a finite sample.

Theorem 4.1: *For an arbitrary class of PA constraints $\mathcal{C}$, the following three statements are equivalent. Below, we let t denote the size of the input constraint $C \in \mathcal{C}$ to be trained, m the sample size, and n the length of each example.*

1. *There exists a training algorithm for $\mathcal{C}$ with sample complexity polynomial in $\frac{1}{\epsilon}, \frac{1}{\delta}, t, n$ and m, running in time polynomial in the total sample length.*
2. *There exists a training algorithm for $\mathcal{C}$ with sample complexity polynomial in $\frac{1}{\epsilon}, \log \frac{1}{\delta}, t, n$ and m, running in time polynomial in the total sample length.*
3. *The sample MLM problem for $\mathcal{C}$ is approximable within a factor $1 + \epsilon$ in random time polynomial in $\frac{1}{\epsilon}, t, n$ and m.*
4. *The sample MLM problem for $\mathcal{C}$ is approximable within factor $2^{p(n,t)m^{\alpha}}$ in random time polynomial in t, n and m, for some $\alpha < 1$.*

Proof of Theorem 4.1
$(1 \rightarrow 3)$ The idea of the proof is as follows. We use the hypothetical training algorithm A for a class of constraints $\mathcal{C}$ to construct a randomized approximation algorithm B for the sample MLM problem for the same class.

In order to do this, we take advantage of the *robustness* of the training algorithm. In particular, the algorithm must meet its performance guarantee when it is fed examples generated by the *empirical distribution* observed in the input sample S for the sample MLM problem. We then appeal to the classical equivalence between minimizing the divergence with respect to the empirical distribution and maximizing the likelihood of a given sample (see Lemma 2.1).

initial state with probability at least $\frac{1}{2tm}$ and each transition in the path has probability at least $\frac{1}{2tm}$. Hence $\log \frac{1}{F(x)}$ is bounded above by $(n+1)\log 2tm$. Since $\mathcal{FPA}(C)_m$ equals $\mathcal{G}'(\frac{1}{m}, \frac{1}{2tm})$, and the cardinality of $\mathcal{G}'(\gamma, \theta)$ is at most $(\frac{1}{\gamma}\ln\frac{1}{\theta})^t$, the cardinality of $\mathcal{FPA}(C)_m$ is at most $(m \ln 2tm)^t$. Plugging in $M = (n+1)\log 2tm$ and $\mid \mathcal{FPA}(C)_m \mid \leq (m \ln 2tm)^t$ into the inequality in Lemma 3.2 gives the following inequality:

$$m \geq \frac{(n+1)^2 \log^2 2tm}{\epsilon^2}(t \ln(m \ln 2tm) + \ln \frac{1}{\delta}) \tag{3.15}$$

To show that if m is at least the bound in Lemma 3.1 then (3.15) holds, we first note that (3.15) is implied by the following two inequalities:

$$\frac{m}{2} \geq \frac{(n+1)^2 \log^2 2tm}{\epsilon^2} t \ln(m \ln 2tm) \tag{3.16}$$

$$\frac{m}{2} \geq \frac{(n+1)^2 \log^2 2tm}{\epsilon^2} \ln \frac{1}{\delta} \tag{3.17}$$

To get a simple argument for (3.16) we first show that

$$\log^2 2tm \ln(m \ln 2tm) \leq \frac{\ln^2 2tm}{\ln^2 2} \ln(tm \ln 2tm) \leq 2 \ln^3 2tm \tag{3.18}$$

For $tm \geq 1$ the latter inequality is equivalent to $\ln \ln 2tm \leq (2 \ln^2 2 - 1) \ln 2tm + \ln 2$ which holds for $tm = 1$ and only improves for larger m. We now return to the proof of (3.16): Since (3.18) holds it suffices to show

$$m \geq \alpha \ln^3 \beta m, \text{ for } \alpha = \frac{4(n+1)^2 t}{\epsilon^2} \text{ and } \beta = 2t. \tag{3.19}$$

If $m \geq \alpha \ln^3 \beta m$ holds for some choice of m then it also does for all larger m. We set m to $8\alpha \ln^3 8\alpha\beta$ which is the first bound in the maximization clause of $q(\frac{1}{\epsilon}, \log \frac{1}{\delta}, n, t)$. Then $m \geq \alpha \ln^3 \beta m$ is equivalent to $8\alpha\beta \geq \ln^3 8\alpha\beta$. Since $n \geq 1$ and $t \geq 2$, $\alpha\beta = \frac{4(n+1)^2 t}{\epsilon^2} \cdot (2t)$ is always at least 64. Now observe that the last inequality holds for $\alpha\beta = 64$ because $(\ln 512)^3 \approx 242.77 < 512$, and thus for all larger $\alpha\beta$. We conclude that (3.19) and (3.16) hold.

The proof of (3.17) is simpler. We need to show

$$m \geq \alpha \log^2 \beta m, \text{ for } \alpha = \frac{2(n+1)^2}{\epsilon^2} \ln \frac{1}{\delta} \text{ and } \beta = 2t.$$

One can show that if $m \geq 4\alpha \log^2 2\alpha\beta$ (which is the second bound in the maximization clause of $q(\frac{1}{\epsilon}, \log \frac{1}{\delta}, n, t)$) then $m \geq \alpha \ln^2 \beta m$.

To complete the proof of Theorem 3.1 observe that $q(\frac{4}{\epsilon}, \log \frac{1}{\delta}, n, t)$ of Lemma 3.1 is at least $\frac{4(n+1)}{\epsilon} + 1$ and fulfills the order bound promised in the statement of the theorem. □

As an immediate corollary to the proof of Theorem 3.1, we have the following positive result for training PAs of a fixed number of parameters.

Corollary 3.2: *Any finite class of PA constraints is polynomially trainable.*

Proof of Corollary 3.2

Since the sample size m required in Theorem 3.1 is polynomial in $\frac{1}{\epsilon}, \frac{1}{\delta}, n$ and t, this immediately follows from the observation that $\mid \mathcal{FPA}(C)_m \mid \leq (2m \log 2tm)^t$ is polynomial in m when t is bounded from above by a constant. □

This completes the proof of (3.11). The second inequality (3.12) is straightforward to verify. Let B be an arbitrary member of $\mathcal{BPA}(C)_m$. We round off each initial probability and each transition probability in B to a power of $(1-\frac{1}{m})$. Denote the obtained PA by R and its stochastic correction by F. Each probability in F is at least $(1-\frac{1}{m})$ times the corresponding probability in B and this leads to two consequences. First, since $B \in \mathcal{BPA}(C)_m$ each initial and transition probability in F is at least $\frac{1}{tm}(1-\frac{1}{m}) \geq \frac{1}{2tm}$, since the final choice of m will be larger than 2. This implies that the 'nearest grid point' R next to B lies in $\mathcal{C}(\frac{1}{m}, \frac{1}{2tm})$ and the stochastic correction F lies in $\mathcal{FPA}(C)_m = \mathcal{G}'(\frac{1}{m}, \frac{1}{2tm})$. Second, each probability in B is at most $(1+\frac{1}{m-1})$ times the corresponding probability in F. Therefore, for any $x \in \Sigma^n$ we must have:

$$\frac{B(x)}{F(x)} \leq (1+\frac{1}{m-1})^{n+1} \leq e^{\frac{n+1}{m-1}}$$

and hence:

$$\log\frac{1}{F(x)} - \log\frac{1}{B(x)} \leq \frac{n+1}{m-1}$$

This proves (3.12) and completes the proof of (3.2).

The next lemma shows that whenever $m \geq q(\frac{4}{\epsilon}, \log\frac{1}{\delta}, n, t)$ the inequality (3.5) holds with probability at least $1-\delta$.

Lemma 3.1: *Let $\mathcal{F}(\mathcal{FPA}(C)_m) = \{\log\frac{1}{F(\cdot)} : F \in \mathcal{FPA}(C)_m\}$. Let D be an arbitrary distribution over Σ^n for some $n \in N$, and D^m denote the product distribution induced by D over $(\Sigma^n)^m$. If $S = \langle w_1, ..., w_m\rangle \in (\Sigma^n)^m$ then we let $E_{\widehat{D}_S}(f)$ denote the empirical estimate of the random variable f by the sample S, i.e. $E_{\widehat{D}_S}(f) = \frac{\sum_{i=1}^m f(w_i)}{m}$, and let $E_D(f)$ denote the expectation (according to D) of f. Then we have, for all $\epsilon \leq 1$:*

$$\forall n, t \in N \quad \textit{If } m \geq q(\frac{1}{\epsilon}, \log\frac{1}{\delta}, n, t),$$

$$\textit{then } D^m\{S \in (\Sigma^n)^m : \exists f \in \mathcal{F}(\mathcal{FPA}(C)_m) \textit{ such that } \mid E_{\widehat{D}_S}(f) - E_D(f) \mid > \epsilon\} < \delta,$$

$$\textit{where } q(\frac{1}{\epsilon}, \log\frac{1}{\delta}, n, t) \textit{ is defined as}$$

$$max\{\frac{32(n+1)^2t}{\epsilon^2}\ln^3\frac{64t(n+1)^2t}{\epsilon^2}, \frac{8(n+1)^2\ln\frac{1}{\delta}}{\epsilon^2}\log^2\frac{8t(n+1)^2\ln\frac{1}{\delta}}{\epsilon^2}\}$$

Proof of Lemma 3.1
We use the following lemma due to Hoeffding. (See for example [Pol84].)

Lemma 3.2 (Hoeffding): *Let $\mathcal{F}$ be a finite class of bounded random variables on a set X, that is for each $f \in \mathcal{F}$, $f : X \to [0, M]$ for some real $M \in R$. Let D be an arbitrary distribution over X. Then we have:*

$$\textit{If } m \geq \frac{M^2}{\epsilon^2}(\ln \mid \mathcal{F} \mid + \ln\frac{1}{\delta}) \textit{ then } D^m\{S \in X^m : \exists f \in \mathcal{F} \mid E_{\widehat{D}_S}(f) - E_D(f) \mid > \epsilon\} < \delta$$

To apply Lemma 3.2, we compute an upper bound on the random variables in $\mathcal{F}(\mathcal{FPA}(C)_m)$, and the cardinality of $\mathcal{FPA}(C)_m$. Since by our assumption at the beginning of the proof $min\{d_{KL}(D, P) : P \in \mathcal{PA}(C)\}$ is finite, for any string x in Σ^n assigned a positive probability by the target distribution D, there is a path for x in the input constraint C. Now any path in C is produced with probability at least $(\frac{1}{2tm})^{n+1}$, since any state is chosen as the

We next verify (3.2) in two steps (inequalities (3.11) and (3.12) below). We define $\mathcal{BPA}(C)_m$, a subclass of $\mathcal{PA}(C)$ in which all transition and initial probabilities are bounded from below by[6] $\frac{1}{tm}$.

$$\mathcal{BPA}(C)_m = \{B \in \mathcal{PA}(C) : \forall i,j \in S, z \in \Sigma \; M_B(i,j,z) \geq \frac{1}{tm} \text{ and } \pi_B(i) \geq \frac{1}{tm}\}$$

We then show that for arbitrary P in $\mathcal{PA}(C)$, there exists $B \in \mathcal{BPA}(C)_m$ which is close to P in the following sense:

$$\forall P \in \mathcal{PA}(C) \; \exists B \in \mathcal{BPA}(C)_m \; \forall x \in \Sigma^n \; \log \frac{1}{B(x)} - \log \frac{1}{P(x)} \leq \frac{n+1}{m-1} \tag{3.11}$$

In addition, we show that for each member of $\mathcal{BPA}(C)_m$ there is one in $\mathcal{FPA}(C)_m$ that is close to it:

$$\forall B \in \mathcal{BPA}(C)_m \; \exists F \in \mathcal{FPA}(C)_m \; \forall x \in \Sigma^n \; \log \frac{1}{F(x)} - \log \frac{1}{B(x)} \leq \frac{n+1}{m-1} \tag{3.12}$$

(3.2) clearly follows from (3.11) and (3.12). To verify the first inequality (3.11), let an arbitrary $P \in \mathcal{PA}(C)$ be given. We obtain P_m from P by shifting each $M_P(i,j,z)$ towards the '*uniform* stochastic matrix' to obtain P_m – the matrix in which each transition in G out of any state i receives the same probability $\frac{1}{t(i)}$, where we let $t(i)$ denote the number of transitions out of state i. Formally:

$$\forall i,j \in S \; \forall z \in \Sigma \; M_{P_m}(i,j,z) = \frac{(m-1) \cdot M_P(i,j,z) + \frac{1}{t(i)}}{m} \tag{3.13}$$

Similarly we shift each $\pi_P(i)$ towards the uniform distribution over I:

$$\forall i \in S \; \pi_{P_m}(i) = \frac{(m-1) \cdot \pi_P(i) + \frac{1}{|I|}}{m} \tag{3.14}$$

Note that for all i,j,z, $M_{P_m}(i,j,z) \geq \frac{1}{mt(i)} \geq \frac{1}{|G|m} \geq \frac{1}{tm}$ and $\pi_{P_m}(i) \geq \frac{1}{|I|m} \geq \frac{1}{tm}$, and hence $P_m \in \mathcal{BPA}(C)_m$. Now since each of the initial and transition probabilities in P_m is at least as much as $\frac{m-1}{m}$ times the corresponding probability in P, the probability assigned by P_m on any path (and hence on any string) is at least $(\frac{m-1}{m})^{n+1}$ times that assigned by P. Hence we have for any string $x \in \Sigma^n$:

$$\frac{P(x)}{P_m(x)} \leq (1 + \frac{1}{m-1})^{n+1} \leq e^{\frac{n+1}{m-1}}$$

Hence, for every $x \in \Sigma^n$, we have

$$\log \frac{1}{P_m(x)} - \log \frac{1}{P(x)} \leq \frac{n+1}{m-1}$$

[6] Note that these bounds can be made slightly smaller, which would result in a slight improvement on the sample complexity. In particular, the bound on the transition probabilities can be $\frac{1}{t^* m}$ where t^* denotes the maximum number of transitions in G out of any state. The bound on the initial probabilities can be $\frac{1}{|I|m}$.

Suppose that m is at least $\frac{4(n+1)}{\epsilon}+1$ and thus $\frac{2(n+1)}{m-1} \leq \frac{\epsilon}{2}$. Then if we let F^* be a member of $\mathcal{FPA}(C)_m$ such that for all $F \in \mathcal{FPA}(C)_m$, $d_{KL}(D,F^*) \leq d_{KL}(D,F)$, then we have:

$$d_{KL}(D,F^*) - d_{KL}(D,Opt) \leq \frac{\epsilon}{2} \tag{3.4}$$

Now by the uniform convergence for the random variables $\mathcal{F}(\mathcal{FPA}(C)_m)$ as stated in Lemma 3.1, for moderately large m ($m \geq q(\frac{4}{\epsilon}, \log\frac{1}{\delta}, n, t)$ where q is polynomially bounded and will be specified in Lemma 3.1), we have with probability at least $1-\delta$:

$$\forall F \in \mathcal{FPA}(C)_m \mid E_{\widehat{D}_S}(\log\frac{1}{F(\cdot)}) - E_D(\log\frac{1}{F(\cdot)}) \mid \leq \frac{\epsilon}{4} \tag{3.5}$$

Now let $Q \in \mathcal{FPA}(C)_m$ be the output of our training algorithm, minimizing the empirical estimate of $\log\frac{1}{Q(\cdot)}$ over $Q \in \mathcal{FPA}(C)_m$. Then, by (3.5), we have that with probability at least $1-\delta$, both of the following hold:

$$E_D(\log\frac{1}{Q(\cdot)}) - E_{\widehat{D}_S}(\log\frac{1}{Q(\cdot)}) \leq \frac{\epsilon}{4} \text{ and} \tag{3.6}$$

$$E_{\widehat{D}_S}(\log\frac{1}{F^*(\cdot)}) - E_D(\log\frac{1}{F^*(\cdot)}) \leq \frac{\epsilon}{4} \tag{3.7}$$

Also by the definition of our training algorithm

$$E_{\widehat{D}_S}(\log\frac{1}{Q(\cdot)}) - E_{\widehat{D}_S}(\log\frac{1}{F^*(\cdot)}) \leq 0 \tag{3.8}$$

By summing the inequalities, (3.6), (3.7) and (3.8), we have the following with probability at least $1-\delta$:

$$E_D(\log\frac{1}{Q(\cdot)}) - E_D(\log\frac{1}{F^*(\cdot)}) \leq \frac{\epsilon}{2} \tag{3.9}$$

This implies:

$$d_{KL}(D,Q) - d_{KL}(D,F^*) \leq \frac{\epsilon}{2} \tag{3.10}$$

Then from (3.4) and (3.10) we have with probability at least $1-\delta$:

$$d_{KL}(D,Q) - d_{KL}(D,Opt) \leq \epsilon$$

Thus any training algorithm that minimizes the empirical divergence over $\mathcal{FPA}(C)_m$ also approximately minimizes the actual divergence over $\mathcal{PA}(C)$ within accuracy ϵ with probability $1-\delta$, whenever $m \geq max\{\frac{4(n+1)}{\epsilon}+1, q(\frac{4}{\epsilon}, \log\frac{1}{\delta}, n, t)\}$.

Proof of Theorem 3.1

We need to show that (3.2) holds and that whenever $m \geq q(\frac{4}{\epsilon}, \log\frac{1}{\delta}, n, t)$ the inequality (3.5) holds with probability at least $1-\delta$, for some q for which $q(\frac{4}{\epsilon}, \log\frac{1}{\delta}, n, t)$ is at least $\frac{4(n+1)}{\epsilon}+1$ and which fulfills the order bound promised in statement of the theorem.

We begin by defining $\mathcal{FPA}(C)_m$. We define a finite subset (grid points), denoted by $\mathcal{G}(\gamma,\theta)$, of $\mathcal{PA}(C)$ as the set of all members P in $\mathcal{PA}(C)$, such that each of $\pi_P(i)$ and $M_P(i,j,z)$ is some power of $1-\gamma$, and is at least θ. Let $\mathcal{G}'(\gamma,\theta)$ be the set of stochastic corrections of the members of $\mathcal{G}(\gamma,\theta)$. We then define $\mathcal{FPA}(C)_m$ as $\mathcal{G}'(\frac{1}{m}, \frac{1}{2tm})$.

and show that the empirical estimates converge to their true means *uniformly* for the class of random variables $\mathcal{F}(\mathcal{PA}(C)) = \{\log \frac{1}{P(\cdot)} : P \in \mathcal{PA}(C)\}$ for moderate sample size. The difficulty here is the fact that $\mathcal{F}(\mathcal{PA}(C))$ is unbounded in the sense that $\log \frac{1}{P(x)}$ diverges to infinity when $P(x)$ goes to zero. This fact prohibits the direct application of certain lemmas on the convergence of random variables, such as Hoeffding's inequality.

It turns out, however, that we do not need to show uniform convergence for the entire class. We let our training algorithm output a probabilistic automaton Q for which $E_{\widehat{D}_S}(\log \frac{1}{Q(\cdot)})$ is minimized when Q is restricted to be in a finite subclass $\mathcal{FPA}(C)_m$ of $\mathcal{PA}(C)$. (If there is more than one member in the finite class that achieves the minimum we make no further assumption about how our algorithm picks its hypothesis.) To describe $\mathcal{FPA}(C)_m$, first note that $\mathcal{PA}(C)$ is nothing but the subset of the 'parameter space' $[0,1]^I \times [0,1]^G$ satisfying the stochastic condition. For convenience we expand $\mathcal{PA}(C)$ by relaxing the equalities in the stochastic condition to inequalities:

$$\forall i \in S \sum_{j \in S, z \in \Sigma} M_P(i,j,z) \leq 1 \text{ and } \sum_{j \in I} \pi_P(j) \leq 1 \tag{3.1}$$

Note that any member P of the parameter space satisfying this weaker condition can easily be converted to a probabilistic automaton P' satisfying the strict stochastic condition which assigns at least as large a probability on every word:

$$\forall i,j \in S, z \in \Sigma \; M_{P'}(i,j,z) = \frac{M_P(i,j,z)}{\sum_{k \in S, z' \in \Sigma} M_P(i,k,z')} \text{ and } \pi_{P'}(i) = \frac{\pi_P(i)}{\sum_{j \in I} \pi_P(j)}$$

Let us say that P' is the *stochastic correction* of P. Note that P' has less divergence than P with respect to *any* distribution[5].

$\mathcal{FPA}(C)_m$ is defined for each sample size m and is the set of stochastic corrections of all 'bounded grid points' of $\mathcal{PA}(C)$, that is, those members of $\mathcal{PA}(C)$ in which all transition and initial probabilities are bounded from below by some decreasing function of m and are powers of $(1-\gamma)$ where γ also is a decreasing function of m.

Now, since $\mathcal{F}(\mathcal{FPA}(C)_m) = \{\log \frac{1}{F(\cdot)} : F \in \mathcal{FPA}(C)_m\}$ is a finite class of bounded random variables of moderate cardinality, we can show *fast* uniform convergence (of empirical estimates to true means) for them. We then show that for sufficiently large m, for each P in $\mathcal{PA}(C)$, there exists $F \in \mathcal{FPA}(C)_m$ which is 'close to' P everywhere in the domain:

$$\forall P \in \mathcal{PA}(C) \, \exists F \in \mathcal{FPA}(C)_m \, \forall x \in \Sigma^n \; \log \frac{1}{F(x)} - \log \frac{1}{P(x)} \leq \frac{2(n+1)}{m-1} \tag{3.2}$$

This immediately implies that for an arbitrary source distribution D over Σ^n, we have:

$$\exists F \in \mathcal{FPA}(C)_m \; d_{KL}(D,F) - d_{KL}(D,P) \leq \frac{2(n+1)}{m-1} \tag{3.3}$$

So if we let Opt be a member of $\mathcal{PA}(C)$ satisfying $d_{KL}(D,Opt) = min\{d_{KL}(D,P) : P \in \mathcal{PA}(C)\}$, then we have:

$$\exists F \in \mathcal{FPA}(C)_m \; d_{KL}(D,F) - d_{KL}(D,Opt) \leq \frac{2(n+1)}{m-1}$$

[5]We have also implicitly extended the notion of divergence for the generalized notion of probability distributions in which the total sum of probabilities over the domain may be less than one.

probability zero. For a stochastic matrix M and a word w, let $M(w)$ be the maximum generation probability assignable on w by M with the *best* initial state. Note that with this definition, $M(uv) \leq M(u) \cdot M(v)$, in general. The single string MLM problem is defined as the problem of finding a *stochastic matrix* M satisfying the input constraint, which maximizes $M(w)$ on the input string w.

Definition 2.6: Single-String MLM Problem for $\mathcal{C}$
Input: A constraint $C \in \mathcal{C}$ and a string w in Σ^*.
Output: A stochastic matrix Q satisfying C which assigns the maximum generation probability on w among all such stochastic matrices, i.e.

$$Q(w) = max\{M(w) : M \in \mathcal{M}(G)\}$$

As usual let **P** denote the class of decision problems decidable in polynomial time and **NP** the class of decision problems acceptable in non-deterministic polynomial time. **RP** denotes the class of decision problems that are acceptable in random polynomial time [Gil77]: A decision problem L is said to be accepted in random polynomial time if and only if there exists a randomized algorithm A, that is, A has access to a fair coin, such that A halts in polynomial time on all inputs, and A always outputs 'no' on a negative instance and outputs 'yes' with probability at least a half on a positive instance. It is widely conjectured that **P** is strictly contained in **NP**, and also that **RP** is strictly contained in **NP**. All hardness results of this paper only hold modulo one of the above conjectures.

3 Sample Complexity Bounds for Training PAs

Our main positive result on the training problem is the following bound on the sample complexity of the PA training problem.

Theorem 3.1: *An arbitrary class of PA constraints $\mathcal{C}$ is trainable with sample complexity $O((\frac{n}{\epsilon})^2 t \cdot \log^3 \frac{nt}{\epsilon} \cdot \log \frac{1}{\delta} \cdot \log^2 \log \frac{1}{\delta})$, where t is the size of the input constraint.*

Note that the above sample complexity bound is essentially linear in the size of the input constraint t, and a low-order polynomial in n, $\frac{1}{\epsilon}$, and $\log \frac{1}{\delta}$. As an easy corollary, the following bound on the sample complexity of the training problem for the complete constraints follows.

Corollary 3.1: *The class of complete PA constraints is trainable with sample size: $O((\frac{n}{\epsilon})^2 s^2 a \cdot \log^3 \frac{nsa}{\epsilon} \cdot \log \frac{1}{\delta} \cdot \log^2 \log \frac{1}{\delta})$, where s is the number of states and a is the alphabet size of the complete constraint to be trained.*

Outline of the proof of Theorem 3.1
Let an input constraint $C = \langle I, G \rangle \in \mathcal{C}$ be given, and let t be its size, i.e. $t = \mid I \mid + \mid G \mid$. Assume that $min\{d_{KL}(D, P) : P \in \mathcal{PA}(C)\}$ is finite with respect to the target probability distribution D, since if the minimum is infinite then by the definition of trainability any sample complexity suffices. Our objective is to show that there exists a training algorithm such that for any sufficiently large sample S, its output $Q \in \mathcal{PA}(C)$ is likely to approximately minimize $d_{KL}(D, Q)$, where D is the source distribution. Recall that $d_{KL}(D; Q) = E_D(\log \frac{1}{Q(\cdot)}) - H(D)$. Since the second term (the entropy of D) is independent of Q, in order to find a Q that minimizes $d_{KL}(D; Q)$, it suffices to find a Q that minimizes $E_D(\log \frac{1}{Q(\cdot)})$. Thus a natural attempt would be to find a Q that minimizes $E_{\widehat{D}_S}(\log \frac{1}{Q(\cdot)})$

when $inf\{d_{KL}(D,P) : P \in \mathcal{PA}(C)\}$ is finite. Define a function $\xi_D : \mathcal{PA}(C) \rightarrow [0,1]$ for an arbitrary target distribution D by :

$$\xi_D(P) = \prod_{x \in \Sigma^n} P(x)^{D(x)}$$

Then since ξ_D is a continuous function, for an arbitrary D, mapping a compact subset of the parameter space $[0,1]^G$, it attains a maximum at a particular PA, say Opt':

$$\xi_D(Opt') = max\{\xi_D(P) : P \in \mathcal{PA}(C)\}$$

Hence,

$$\begin{aligned} E_D(\log \frac{1}{Opt'(\cdot)}) &= \sum_{x \in \Sigma^n} D(x) \log \frac{1}{Opt'(x)} \\ &= \log \frac{1}{\xi_D(Opt')} \\ &= \log \frac{1}{max\{\xi_D(P) : P \in \mathcal{PA}(C)\}} \\ &= min\{\log \frac{1}{\xi_D(P)} : P \in \mathcal{PA}(C)\} \\ &= min\{E_D(\log \frac{1}{P(\cdot)}) : P \in \mathcal{PA}(C)\} \end{aligned} \tag{2.5}$$

Noting that $d_{KL}(D,P) = E_D(\log \frac{1}{P(\cdot)}) - H(D)$, the inequality (2.5) implies that $d_{KL}(D,Opt')$ equals $min\{d_{KL}(D,P) : P \in \mathcal{PA}(C)\}$. Now let $Opt = Opt'$ and we see that Opt is well-defined.

We define two versions of the maximum likelihood model problem for PAs considered in this paper.

Definition 2.5: Sample MLM Problem for $\mathcal{C}$
Input: A constraint $C \in \mathcal{C}$, a string length n and a finite sample $S = \langle w_1, ..., w_m \rangle$ of strings from Σ^n.
Output: A probabilistic automaton Q satisfying C which assigns the maximum generation probability (or the maximum likelihood) on S among all such probabilistic automata, i.e.

$$\prod_{i=1}^{m} Q(w_i) = max\{\prod_{i=1}^{m} P(w_i) : P \in \mathcal{PA}(C)\}$$

Again note that $max\{\prod_{i=1}^{m} P(w_i) : P \in \mathcal{PA}(C)\}$ is well-defined because $\zeta : \mathcal{PA}(C) \rightarrow [0,1]$ defined by

$$\zeta(P) = \prod_{i=1}^{m} P(w_i)$$

is a continuous function mapping a compact domain $\mathcal{PA}(C)$ into the range $[0,1]$.

The following definition is a special case of the sample maximum likelihood model problem in which the input sample consists of a single string. Note that for a single string, the initial probability distribution plays no significant role, because among probabilistic automata assigning the maximum probability on a given string there is always a probabilistic automaton in which exactly one state has initial probability one and all other states have

the Kullback-Leibler divergence with respect to the source distribution also minimizes the expected code length of the data. It is also well-known that minimizing the Kullback-Leibler divergence with respect to the *empirical* distribution $\widehat{D}_S$ observed in a sample $S = \langle w_1, ..., w_m \rangle$ (Definition 2.1) corresponds to maximizing the likelihood of the sample, as demonstrated below. Minimizing $d_{KL}(\widehat{D}_S; Q)$ corresponds to minimizing $E_{\widehat{D}_S}(\log \frac{1}{Q(\cdot)})$, and the following always holds:

$$E_{\widehat{D}_S}(\log \frac{1}{Q(\cdot)}) = \frac{1}{m} \cdot \sum_{i=1}^{m} \log \frac{1}{Q(w_i)} = \frac{1}{m} \log \prod_{i=1}^{m} \frac{1}{Q(w_i)} \tag{2.4}$$

Thus, $d_{KL}(\widehat{D}_S; Q)$ is minimized when $\prod_{i=1}^{m} Q(w_i)$ is maximized, i.e. when Q maximizes the probability of having generated the sample. We summarize this as a lemma.

Lemma 2.1: *Let $\mathcal{P}$ be an arbitrary class of distributions over Σ^n, $S = \langle w_1, ..., w_m \rangle$ an arbitrary sample of Σ^n, and $\widehat{D}_S$ the empirical distribution of S. Then,*

$$E_{\widehat{D}_S}(\log \frac{1}{Q(\cdot)}) = inf\{E_{\widehat{D}_S}(\log \frac{1}{P(\cdot)}) : P \in \mathcal{P}\}$$

if and only if

$$\prod_{i=1}^{m} Q(w_i) = sup\{\prod_{i=1}^{m} P(w_i) : P \in \mathcal{P}\}$$

Below we give the definition of a training algorithm which is the central definition in this paper. Here we assume that a *randomized algorithm* has access to a fair coin and can flip it in a single time unit.

Definition 2.4 (Training PA Constraints): *A training algorithm takes as input a constraint C, a string length n, and a finite sample S of Σ^n for some alphabet Σ, and outputs a probabilistic automaton which satisfies C. We say that a (possibly randomized) training algorithm A trains a class of constraints $\mathcal{C}$ with sample size $q(\frac{1}{\epsilon}, \frac{1}{\delta}, n, t)$, if A, when given as input an arbitrary constraint $C \in \mathcal{C}$ of size t, a string length n, and a finite sample S drawn independently at random from an arbitrary unknown distribution D over Σ^n, is such that whenever the sample size m exceeds $q(\frac{1}{\epsilon}, \frac{1}{\delta}, n, t)$, then provided that $min\{d_{KL}(D, P) : P \in \mathcal{PA}(C)\}$ is finite, A's output Q satisfies the following with probability at least $1 - \delta$:*

$$d_{KL}(D, Q) - d_{KL}(D, Opt) \leq \epsilon$$

where Opt is a member of $\mathcal{PA}(C)$ satisfying:

$$d_{KL}(D, Opt) = min\{d_{KL}(D, P) : P \in \mathcal{PA}(C)\}$$

Here the probability is taken over the product distribution of D producing the sample and the random coin flips of A, if A is probabilistic. If there exists such an algorithm then we say that $\mathcal{C}$ is trainable with sample complexity $q(\frac{1}{\epsilon}, \frac{1}{\delta}, n, t)$. If there exists such an algorithm running in time polynomial in the total sample length, then we say that $\mathcal{C}$ is polynomially trainable.

Note that $inf\{d_{KL}(D, P) : P \in \mathcal{PA}(C)\}$ is infinite if and only if there is a word $x \in \Sigma^n$ such that $D(x) > 0$ and there is no path in C for x. In this case, $d_{KL}(D, P) = \infty$ for any P in $\mathcal{PA}(C)$. We still need to verify that Opt in the above definition is well-defined,

is a pair $C = \langle I, G \rangle$ where I is the *initial state set* and G is the transition graph of C. I is a subset of the set S of all states. G is a subset of the set $S \times S \times \Sigma$ of all *transitions*. Note that a transition graph is a labeled directed graph with the state set S as the vertices, and the alphabet Σ as the set of labels. We write $\mid I \mid$ for the number of states in I, $\mid G \mid$ for the number of transitions in G. We then define the *size* of a constraint C, written $\mid C \mid$, as $\mid C \mid = \mid I \mid + \mid G \mid$. Note that the size of C corresponds to the number of probability parameters specified by C.

We say that there is a path in C for a string $w_1 ... w_n \in \Sigma^n$, if there is a sequence of states $\sigma = \langle \sigma_0, ..., \sigma_{n+1} \rangle$ from S^{n+1} such that $\sigma_0 \in I$ and for all i, $0 \leq i < n$, $(\sigma_i, \sigma_{i+1}, w_{i+1}) \in G$. We say that a probabilistic automaton P *satisfies* a constraint $C = \langle I, G \rangle$, if and only if $S_P = S$ and $\Sigma_P = \Sigma$ and

$$\forall i \notin I, \;\; \pi_P(i) = 0 \text{ and}$$

$$\forall (i, j, z) \notin G, \;\; M_P(i, j, z) = 0 \tag{2.3}$$

Note that if a probabilistic automaton P satisfies a constraint C then for every word on which P assigns a positive probability, there is a path for it in C. If P satisfies C, one can think of π_P as a function from I into $[0, 1]$, and M_P as a function from G into $[0, 1]$. We let $\mathcal{PA}(C)$ denote the class of probabilistic automata satisfying the constraint C. Note that $\mathcal{PA}(C) \subset [0, 1]^I \times [0, 1]^G$, where we let $[0, 1]^I$ denote the class of all functions from I into $[0, 1]$, and $[0, 1]^G$ from G into $[0, 1]$. (In general, we use Y^X to denote the class of functions from domain X into range Y.) We also let $\mathcal{M}(G)$ denote the class of stochastic matrices M_P satisfying (2.3).

The notion of 'distance' among distributions we employ in this paper is a well-known measure in information theory called 'Kullback-Leibler divergence,' also known as the 'relative entropy.'

Definition 2.3 (Kullback-Leibler Divergence): *Let D and Q be probability distributions over countable domain X. The 'Kullback-Leibler divergence' of Q with respect to D, $d_{KL}(D; Q)$ is defined as follows.*

$$d_{KL}(D; Q) = \sum_{x \in X} D(x) \log \frac{D(x)}{Q(x)}$$

(Normally we think of D as the actual distribution against which a 'candidate' distribution Q is being compared. By convention, we let $0 \log 0 = 0$, and $\frac{0}{0} = 1$.)

Note in the above definition that the base of logarithm is 2, following the conventions in coding theory. (Throughout this paper we let ln denote the natural logarithm and log the logarithm base 2.) The Kullback-Leibler divergence enjoys many natural properties: We can rewrite $d_{KL}(D; Q)$ as $E_D(\log \frac{1}{Q(\cdot)}) - H(D)$ where $E_D(\log \frac{1}{Q(\cdot)})$ is the expectation according to D of the random variable $\log \frac{1}{Q(\cdot)}$, and $H(D)$ is the 'entropy' of D, defined as $\sum_{x \in X} D(x) \log \frac{1}{D(x)}$. Recall that $\log \frac{1}{D(x)}$ is the code length for x with respect to the ideal code for D, and $H(D)$ is the expected code length[4] of that code [Ham86] for the source distribution D. In other words, for the source distribution D, the divergence $d_{KL}(D; Q)$ measures the expected additional code length required when using the ideal code for Q instead of the ideal code for D. Thus, the ideal code of the distribution which minimizes

[4] In Physics, it is customary to use the natural logarithm for the definition of entropy. We use the binary logarithm for the entropy and the Kullback-Leibler divergence as is done in information and coding theory.

Note that for any y not in S, we have $\widehat{D}_S(y) = 0$.

The probabilistic automaton is formalized as a stochastic matrix M together with an 'initial distribution' π over the set of states. Intuitively, the probabilistic automaton is much like a non-deterministic finite state automaton except that the transitions take place with probabilities prescribed by M. (See Figure C.1.) To start the process, the machine chooses the initial state according to the initial distribution π, and then at any given point after that, the machine is in some state i, and at the next time step moves to another state j outputting some letter z, with probability specified by $M(i, j, z)$. If one stops the machine at time step n the machine ends in some state having generated a string of length n. In this way, a probabilistic automaton naturally defines a probability distribution over the set of strings of length n, for any particular n.

Definition 2.2 (Probabilistic Automata (PA)): *A probabilistic automaton P is a quadruple $\langle S_P,\ \Sigma_P,\ \pi_P,\ M_P\rangle$ where S_P is a finite set of states, Σ_P is a finite alphabet, $\pi_P : S_P \to [0, 1]$ is a probability distribution over S_P, and $M_P : S_P \times S_P \times \Sigma_P \to [0, 1]$ is a stochastic matrix[3], i.e.*

$$\sum_{i \in S_P} \pi_P(i) = 1 \text{ and } \forall i \in S_P \sum_{j \in S_P, z \in \Sigma_P} M_P(i, j, z) = 1 \tag{2.1}$$

Each $\pi_P(i)$ is called an initial probability, and each $M_P(i, j, z)$ is called a transition probability. For any string $w = w_1 ... w_n \in \Sigma_P^*$, the generation probability assigned on it by $P = \langle S_P, \Sigma_P, \pi_P, M_P\rangle$ is computed as follows.

$$P(w_1 ... w_n) = \sum_{\langle i_0,...,i_n\rangle \in S_P^{n+1}} \pi_P(i_0) \cdot \prod_{j=0}^{n-1} M_P(i_j, i_{j+1}, w_{j+1}) \tag{2.2}$$

Thus, for any given example length n, P defines a probability distribution over Σ_P^n.

For example, the probability assigned by the probabilistic automaton P shown in Figure C.1 on the string $w = aab$ is calculated as follows:

$$\begin{aligned}
P(w) &= \pi_P(0) \cdot M_P(0, 0, a) \cdot M_P(0, 1, a) \cdot M_P(1, 1, b) \\
&\quad + \pi_P(0) \cdot M_P(0, 1, a) \cdot M_P(1, 1, a) \cdot M_P(1, 1, b) \\
&\quad + \pi_P(0) \cdot M_P(0, 1, a) \cdot M_P(1, 2, a) \cdot M_P(2, 0, b) \\
&\quad + \pi_P(1) \cdot M_P(1, 1, a) \cdot M_P(1, 1, a) \cdot M_P(1, 1, b) \\
&\quad + \pi_P(1) \cdot M_P(1, 1, a) \cdot M_P(1, 2, a) \cdot M_P(2, 0, b) \\
&= 0.5 \cdot 0.4 \cdot 0.6 \cdot 0.1 + 0.5 \cdot 0.6 \cdot 0.1 \cdot 0.1 + 0.5 \cdot 0.6 \cdot 0.8 \cdot 0.6 + \\
&\quad 0.5 \cdot 0.1 \cdot 0.1 \cdot 0.1 + 0.5 \cdot 0.1 \cdot 0.8 \cdot 0.6 \\
&= 0.012 + 0.003 + 0.072 + 0.0005 + 0.024 = 0.1115
\end{aligned}$$

Note that for a probabilistic automaton P we use the same letter P to denote the probability distribution defined by P on Σ_P^n, where n will be clear from the context. A *PA constraint*

[3]The probabilistic automaton is often formulated as a probabilistic acceptor in the literature. Here we view PAs as generators. Thus the stochastic condition in this definition states that, for each state, the total probability of transitions out of that state sums to one, rather than the total probability for each state-letter pair as is the case for PAs as acceptors. Tzeng considers the incomparable problem of learning PAs as acceptors from queries [Tze89].

approximate (unless $\mathbf{P} = \mathbf{NP}$), and hence the class of 2-state complete constraints is not polynomially trainable (unless $\mathbf{RP} = \mathbf{NP}$).

The hardness result for the MLM problem for 2-state complete constraints is shown via the following non-approximability result for the *single string* MLM problem for the same class – the special case of MLM in which the input sample consists of a single string. We show that it is hard to approximate the single string MLM problem for the 2-state complete constraints within a factor of $2^{|w|^{1-\alpha}}$ for any positive constant α, where w is the input word, in time polynomial in the word length and alphabet size, unless $\mathbf{P} = \mathbf{NP}$. Note that it is a very strong non-approximability result, since there is a trivial training algorithm, using only 1-state probabilistic automata, that can guarantee approximation within a factor of $2^{|w|}$ of the best 2-state PA. The proof of the hardness result uses as a starting point the type of technique commonly used in the learning theory literature for showing the hardness of a 'sample consistency' or 'minimum consistent concept' problem in discrete domains such as automata and boolean formulas [Gol78,Ang78,PW89]. In particular, our proof makes use of notions used in Angluin's proof of the NP-completeness of the sample consistency problem for 2-state DFA [Ang89]. The proof given here is, however, significantly more complex than the proof of the discrete case, since corresponding to 'consistency' we have 'probability,' which is continuous and is thus much harder to get a hold of. For example, in our reduction of the satisfiability problem to the MLM problem for the 2-state complete constraints, it is already non-trivial to formalize how a truth assignment is to be simulated by a PA. We let each truth assignment correspond *conceptually* to one of 2^n many *deterministic* PAs of a *particular* kind, and for all the other (infinitely many) PAs, we quantify how 'far' they are from those corresponding to truth assignments. We then show that any PA that assigns the input word w a probability at least $\frac{1}{2^{|w|^{1-\alpha}}}$ times the probability assigned on w by an optimum PA must be 'close' to a deterministic PA corresponding to a satisfying assignment.

This paper is outlined as follows. We begin in Section 2 with some preliminary definitions and give the proof of the sample size bounds for training PAs in Section 3. In Section 4 we show the equivalence between the training problem and the approximate MLM problem for any class of constraints. In Section 5 we give the hardness result for the single string MLM problem for 2-state complete constraints. Parts of this lengthy proof are given in Appendices A and B. In Section 6 we discuss briefly how the results of this paper apply to HHMs. We conclude by discussing a number of open problems inspired by this research in Section 7.

2 Preliminaries

This paper deals with approximating a probability distribution over words over some finite alphabet, Σ. For simplicity, we assume that all words with positive probability have the same length, n, i.e. the distribution is over the domain Σ^n. We call an element of Σ^n an *example*. A *sample* S of Σ^n is a finite sequence of examples of Σ^n, $S = \langle w_1, ..., w_m \rangle$, where m is the sample size. We abuse notation and write $x \in S$ to mean that x appears in the sequence S. We let $\sharp(x, S)$ denote the number of occurrences of example x in sample S. Using the above notation, we define the notion of the empirical distribution of a sample.

Definition 2.1: *Given a sample S of size m of Σ^n, the empirical distribution of S over Σ^n, written $\widehat{D}_S$, is defined by:*

$$\forall x \in \Sigma^n \; \widehat{D}_S(x) = \frac{\sharp(x, S)}{m}$$

Using this model, we show a number of results: We show that an arbitrary class of constraints is trainable by exhibiting a training algorithm whose sample complexity is essentially linear in the size of the constraint being trained and a low-degree polynomial in the example length and parameters quantifying the accuracy and confidence. In addition, the running time of our training algorithm is polynomial in the total sample length if the size of the input constraint is bounded by a constant, thus showing that finite classes are polynomially trainable. In particular, an arbitrary *fixed* constraint is trainable in time polynomial in the accuracy and confidence parameters and the example length. If the alphabet size is variable, however, no polynomial time training algorithm exists for the class of 2-state complete constraints[2], unless **RP** = **NP**.

To the best of our knowledge, our upper bound is the first rigorous result on the sample complexity of training PAs and HMMs, with respect to the classical measure of Kullback-Leibler divergence. Our proof is also interesting in the sense that we manage to get around the problem caused by the fact that the Kullback-Leibler divergence is unbounded. This property prohibits the direct use of certain useful techniques such as Hoeffding's inequality for showing uniform convergence of empirical estimates of random variables to their true means. We get around this problem roughly as follows: We bound the smallest transition probabilities in the training algorithm's hypotheses from below by a decreasing function of the sample size m. We show uniform convergence for these successive classes of 'bounded' probabilistic automata using Hoeffding's inequality. We then show that for sufficiently large m, an optimum automaton in the m-th bounded class is close to an optimum one in the entire class with high probability. Interestingly, the trick of bounding probabilities away from zero is often used in practice in an attempt to solve what is known as the 'finite sample problem' [LRS83]. Our result provides a rigorous justification for a particular way of setting those probability bounds, from the point of view of proving bounds on the sample complexity.

The sample complexity bound we obtain allows us to extend the classical equivalence between the minimization of the Kullback-Leibler divergence with respect to the *empirical* distribution and the maximization of the likelihood of the given data: We show that the polynomial time trainability of a class of constraints $\mathcal{C}$ is equivalent to the polynomial time approximability of the 'maximum likelihood model' problem (MLM) for the same class $\mathcal{C}$ – the problem of setting the initial and transition probabilities in a given constraint in $\mathcal{C}$ so that the probability assigned on a given finite sample is maximized. More precisely, we show that the polynomial time trainability of a class of constraints $\mathcal{C}$ is equivalent to the approximability of the MLM problem for $\mathcal{C}$ within a factor $1+\epsilon$ in random time polynomial in $\frac{1}{\epsilon}$ and the size t of the input constraint. Furthermore, we show that this latter notion of approximability of the MLM problem for $\mathcal{C}$ is also equivalent to a seemingly much weaker notion of approximability: Approximability within factor $2^{p(n,t)m^{\alpha}}$ in random polynomial time, where m is the sample size, α is an arbitrary constant less than 1, and $p(n,t)$ is a polynomial in the example length n and the size of the input constraint t. We use the above equivalence between the training and MLM problems to show our hardness result: For *variable* alphabet size, the MLM problem for 2-state *complete* constraints, or the problem of finding a 2-state PA assigning the maximum likelihood on the input sample, is hard to

[2]The existence of an algorithm for training hidden Markov models which always outputs a *near local optimum* on a given sample is well-known ('Baum-Welch' algorithm [Bau72]) and is used extensively in practice. Note that in applications to speech recognition, the alphabet size is determined by how precise the acoustic signals are quantized. The alphabet size is often in the hundreds.

1 Introduction

We address the problem of approximating an arbitrary, unknown source distribution by distributions generated by probabilistic automata. Probabilistic automata (PAs), and hidden Markov models[1] (HMMs) which are closely related to PAs, are used extensively as models for probabilistic generation of speech signals for the purpose of speech recognition (see for example [LRS83]). The problem addressed in the present paper corresponds to that of training a parameterized hidden Markov model for a particular spoken word with a set of actual speech signals for that word. In particular, we are interested in the question of whether there exists an algorithm that, when given a sample generated from an arbitrary unknown target distribution, outputs a probabilistic automaton that approximates the unknown distribution 'as closely as possible,' that is, with high probability the distribution induced by the output PA is sufficiently close to an 'optimal' one among all possible probabilistic automata satisfying a certain *prescribed constraint*. Here a *constraint* is given to the algorithm in the form of a subset of the state set specifying the *legal initial states* and a labeled directed graph specifying the set of *legal transitions*. The training problem, therefore, is the problem of finding a near optimal setting of the initial and transition probabilities on the legal initial states and transitions in the input constraint. A *class* of constraints is said to be trainable with sample complexity $q(...)$ if there exists an algorithm which trains every constraint in the class, and the sample size required for a given accuracy, confidence, length of the example strings, and the size of the input constraint is bounded above by the function q of these parameters. Here the size of the input constraint translates to the number of probability parameters being trained. A class of constraints is said to be *polynomially trainable* if there is a training algorithm with polynomial sample complexity whose running time is polynomial in the total sample length. Of particular interest to us is the special case of this problem in which the input constraint is complete, namely all initial states and transitions are legal. This special case translates to the problem of finding a near optimal probabilistic automaton with a *given number of states*.

Our model is a natural adaptation of the PAC-learning paradigm of Valiant [Val84, BEHW89] and is inspired by the model of efficient unsupervised learning of Laird [Lai88]. It is also related to the models for learning languages from stochastic data in the limit proposed and studied by Angluin [Ang88]. Our formulation requires the algorithm to be particularly *robust* in the sense that we do not assume anything about the target distribution – a formulation which is closely related to the 'robust' generalization of the PAC paradigm proposed by Haussler in [Hau89]. The distance measure between the distributions used in this paper to evaluate the accuracy of a hypothesis with respect to the target distribution is the well-known 'Kullback-Leibler divergence.' Other commonly used measures of distance between probability distributions are, for example, the χ^2 distance, the variation distance, the quadratic distance [KS90], and the Hellinger distance. The Kullback-Leibler divergence is a standard notion of distance, which enjoys many desirable properties (see Section 2). Furthermore, the Kullback-Leibler divergence is known to bound from above the Hellinger distance as well as the square of the variation distance and of the quadratic distance. These relationships for the more general case of *conditional distributions* are surveyed by Yamanishi in [Yam91].

[1] HMMs are similar to probabilistic automata, except that outputs in an HMM are associated with the states rather than the transitions, and thus the transitions are unlabeled state to state pairs.

ABSTRACT

We introduce a rigorous performance criterion for training algorithms for probabilistic automata (PAs) and hidden Markov models (HMMs), used extensively for speech recognition, and analyze the complexity of the training problem as a computational problem. The PA training problem is the problem of approximating an arbitrary, unknown source distribution by distributions generated by a PA. We investigate the following question about this important, well-studied problem: Does there exist an *efficient* training algorithm such that the trained PAs *provably converge* to a model close to an optimum one with high confidence, after only a feasibly small set of training data? We model this problem in the framework of computational learning theory and analyze the sample as well as computational complexity. We show that the number of examples required for training PAs is moderate – essentially linear in the number of transition probabilities to be trained and a low-degree polynomial in the example length and parameters quantifying the accuracy and confidence. Computationally, however, training PAs is quite demanding: Fixed state size PAs are trainable in time polynomial in the accuracy and confidence parameters and example length, but *not* in the alphabet size, unless **RP** = **NP**. The latter result is shown via a strong non-approximability result for the single string maximum likelihood model problem for 2-state PAs, which is of independent interest.

On the Computational Complexity of Approximating Distributions by Probabilistic Automata*

Naoki Abe
Manfred K. Warmuth

UCSC-CRL-90-63

December 28, 1990

Baskin Center for
Computer Engineering & Information Sciences
University of California, Santa Cruz
Santa Cruz, CA 95064 USA

*Supported by the Office of Naval Research, contract number N0014-86-K-0454. Part of this work was done after the first author began employment by C & C Information Technology Research Labs, NEC, 4-1-1 Miyazaki Miyamae-ku, Kawasaki, 216 Japan, and the second author was employed by IIAS-SIS Fujitsu Limited, Numazu, Japan. Email addresses: abe@ibl.cl.nec.co.jp and manfred@cis.ucsc.edu.