2023 59th Annual Allerton Conference on Communication, Control, and Computing (Allerton) | 979-8-3503-2814-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/Allerton58177.2023.10313360

Learning Source Coding for General Alphabets and Finite State Machines

Mohammad Zaeri Amirani, Anders Høst-Madsen
Department of Electrical & Computer Engineering
University of Hawaii, Manoa
Honolulu, HI, 96822, Email: {zaeri,ahm}@hawaii.edu

***Abstract*—This paper develops bounds for learning lossless source coding under the PAC (probably approximately correct) framework. The paper considers both iid sources and sources generated by finite state machines (FSM).**

I. Introduction

Traditionally, there have been two types of source coders: fixed, optimum coders such as Huffman coders; and universal source coders, such as Lempel-Ziv [1], [2], [3]. We will consider a third type of source coders: learned coders. These are coders that are trained on data of a particular type, and then used to encode new data of that type. Examples could be source coders for English texts, DNA data, or protein data represented as graphs.

In both machine learning and information theory literatures, there has been some work on learned coding. From a machine learning perspective, the paper [4] stated the problem precisely and developed and evaluated some algorithms. A few follow up papers, e.g., [5], [6], [7], [8], [9], [10] have introduced new machine learning algorithms. For lossy coding, in particular of images and video, there has been much more activity recently, initiated by the paper [11] from Google, see for example [12], [13], [14]. Our aim is to find theoretical bounds for how well it is possible to learn coding. In the current paper we will limit ourselves to lossless coding.

Stating the problem more precisely, we consider the following problem of learned coding [16]. We are given a training sequence x^m; based on the training we develop coders $C(x^l;x^m)$ with length function $L(x^l;x^m)$ for encoding *test sequences* x^l. The codelength is $\frac{1}{l}E_\theta[L(X^l;x^m)|x^m]$ (the expectation here is only over x^l), and the redundancy is

$$R_l(L,x^m,\boldsymbol{\theta})=\frac{1}{l}E_{\boldsymbol{\theta}}[L(X^l;x^m)|x^m]-H_{\boldsymbol{\theta}}(X). \quad (1)$$

The redundancy depends on the training sequence x^m. One way to remove this dependency is to average also over x^m,

$$R_l(L,m,\boldsymbol{\theta})=\frac{1}{l}E_{\boldsymbol{\theta}}[L(X^l;X^m)]-H_{\boldsymbol{\theta}}(X) \quad (2)$$

$$R_l^+(m)=\min_L\sup_\theta R_l(L,m,\boldsymbol{\theta}). \quad (3)$$

The research was funded in part by the NSF grant CCF-1908957.

The paper [17] considers (3), and proves

$$\frac{1}{2m\ln 2}+o\left(\frac{1}{m}\right)\le R_l^+(m)\le\frac{\alpha_0}{m\ln 2}+o\left(\frac{1}{m}\right) \quad (4)$$

$$\alpha_0\approx 0.50922. \quad (5)$$

for the IID case. The papers [18], [19] consider some generalizations to the Markov case.

However, in machine learning performance usually is not measured by average over test sequences, see [20], [21]. One way performance is measured is in the PAC (probability approximately correct) learning framework [21]. Rather than usual error probability in classification, we use the redundancy (1) as risk measure. We can then say that coding in a class or sources is PAC-learnable if for any $a>0, P_e>0$ and for any sample size $m>\text{poly}(1/a,1/P_e)$

$$\inf_{\boldsymbol{\theta}}P\left(R_l(L,X^m,\boldsymbol{\theta})\le a\right)\ge 1-P_e$$

where the probability is over X^m. Alternatively, we can state this by defining

$$E(m,a)=\sup_{\boldsymbol{\theta}}P\left(R_l(L,X^m,\boldsymbol{\theta})>a\right), \quad (6)$$

For some given a and small P_e the goal is then to ensure $E(m,a)\le P_e$. Thus, we require the redundancy of the learned codelength to be smaller than a, except with a small error probability P_e.

The idea of learning to code is to obtain information about the distribution of the source from the training x^m and then apply this to code the test sequence x^l. One can take two approaches to the application phase. First, the the coder can be *frozen* in the sense that it does not further update from the test sequence (in that case $E(m,a)$ in (6) does not depend on l). There are both practical and theoretical reasons for freezing the coder. Machine learning algorithms usually have a distinct learning phase, and once the algorithm is trained, it is not updated with test samples; the reason for this is both that training is much more computational intensive than application, often run on specialized hardware, and that there are few good algorithms for updating for example neural networks with new data. As a case in point, the LSTM in [4] was not updated after the training phase, and the theoretical work in [15] also considered frozen coders. We have some result for non-frozen coders (online learning), and they show that only for the uninteresting scenario $l\gg m$ does it make

any difference. We will therefore limit ourselves to the frozen scenario here.

In [16] we analyzed this problem for the IID with a binary alphabet. The result is

Theorem 1. *For estimators that are functions of the sufficient statistic and P_e sufficiently small,*

$$a(m,P_e) \geq \frac{Q^{-1}(P_e/2)^2}{2m\ln 2} + o\left(\frac{1}{m}\right). \tag{7}$$

For the estimator $\hat{p} = \frac{k+\alpha}{m+2\alpha}$. The optimum value of α that satisfies $\frac{1}{6}Q^{-1}(P_e/2)^2 - 1 \leq \alpha \leq \frac{1}{6}Q^{-1}(P_e/2)^2 + 1$ which gives an achievable $a(m,P_e)$;

$$a(m,P_e) = b(P_e)\frac{Q^{-1}(P_e/2)^2}{2m\ln 2} + o\left(\frac{1}{m}\right), \tag{8}$$

where $\lim_{P_e\to 0} b(P_e) = 1$.

In the current paper we will generalize this to a general K-alphabet IID source, and to (binary) sources generated by finite state machines.

II. Learning for general IID Sources

We consider an alphabet with $K+1$ symbols. The average case has already been solved in (4), so we we only need to consider PAC performance. In this case

$$E(m,a) = \sup_P P(D(P||\hat{P}) > a) \tag{9}$$

where $\hat{P}$ is an estimate of the K parameter probability distribution. We use the add-α estimator [22], [17]

$$\hat{P}_k = \frac{n_k+\alpha}{m+(K+1)\alpha} = \frac{\check{P}_k + \alpha/m}{1+(K+1)\alpha/m} \tag{10}$$

where $\check{P}_k = \frac{n_k}{m}$ with n_k the number of observations of symbol k. We let $P_{K+1} = 1 - \sum_{k=1}^K P_k$ and $\hat{P}_{K+1} = 1 - \sum_{k=1}^K \hat{P}_k$. We can consider sequences of probabilities $P_k(m)$ and then take supremum of the limits. For each component there are two possibilities

$$\lim_{m\to\infty} mP_k(m) = \gamma_k < \infty \quad \text{and} \quad \lim_{m\to\infty} P_k(m) = \bar{P}_k > 0$$

We allow $\gamma_k = 0$. We can assume it is the first $K_p < K+1$ components that have finite limit. We first have

Lemma 2. *Let*

$$\mathbf{P}(m) = \big[m\check{P}_1(m),\ldots,m\check{P}_{K_p}(m),$$
$$\sqrt{m}(\check{P}_{K_p+1}(m) - P_{K_p+1}(m)),\ldots,\sqrt{m}(\check{P}_K(m) - P_K(m))\big]$$

then $\mathbf{P}(m) \xrightarrow{D} \mathbf{P}$, where $\mathbf{P}$ is a random vector with

- *$P_k = \psi_k \sim$ Pois(γ_k), independent of other components*
- *The $[P_{K_p+1},\ldots,P_K]$ is multivariate Gaussian.*

Proof. We define:

$$\mathbf{A}(m) = \begin{bmatrix} m\boldsymbol{I}_{K_p} & \mathbf{0}_{K_p,K+1-K_p} \\ \mathbf{0}_{K+1-K_p,K_p} & \sqrt{m}\boldsymbol{I}_{K+1-K_p} \end{bmatrix}$$
$$\mathbf{b}(m) = \sqrt{m}\left[\mathbf{0}_{1,K_p}, \bar{P}_{K_p+1},\ldots,\bar{P}_K,\bar{P}_{K+1}\right]^T$$

Then we have $\mathbf{P}(m) = \mathbf{A}(m)\check{\mathbf{P}}(m) - \mathbf{b}(m)$.

As the n_k are multinomial, the characteristic function for $\mathbf{P}(m)$ then is $\varphi_{\mathbf{P}(m)}(\mathbf{t}) = e^{-i\mathbf{t}^T\mathbf{b}(m)}\left(\mathbf{P}(m)^T e^{i\frac{1}{m}\mathbf{A}(m)\mathbf{t}}\right)^m$.

Now we have:

$$\begin{aligned}
&\lim_{m\to\infty} \ln\varphi_{\mathbf{P}(m)}(\mathbf{t}) \\
&\quad= \lim_{m\to\infty}\left(-i\mathbf{t}^T\mathbf{b}(m) + m\ln\left(\mathbf{P}(m)^T e^{i\frac{1}{m}\mathbf{A}(m)\mathbf{t}}\right)\right) \\
&\quad= \lim_{m\to\infty}\left(-i\sqrt{m}\sum_{k=K_p+1}^{K+1}\bar{P}_k t_k + m\ln\left(\sum_{k=1}^{K_p}P_k(m)e^{it_k}\right.\right. \\
&\qquad\left.\left.+\sum_{k=K_p+1}^{K+1}P_k(m)\left(1+\frac{it_k}{\sqrt{m}}-\frac{t_k^2}{m}+o\left(\frac{1}{m\sqrt{m}}\right)\right)\right)\right)
\end{aligned} \tag{11}$$

Note that $\sum_{k=1}^{K+1}P_k(m) = 1$ and $\lim_{m\to\infty} mP_k(m) = \gamma_k < \infty$ for $k = 1,\ldots,K_p$, so by applying the Taylor expansion of $\ln(1+x)$ we have:

$$\begin{aligned}
&\lim_{m\to\infty}\ln\varphi_{\mathbf{P}(m)}(\mathbf{t}) = \sum_{k=1}^{K_p}\gamma_k(e^{it_k}-1) \\
&+\lim_{m\to\infty}\left(-\sum_{k=K_p+1}^{K+1} i\sqrt{m}(\bar{P}_k - P_k(m))t_k + \frac{1}{2}P_k^2(m)t_k^2\right) \\
&+\lim_{m\to\infty}\sum_{k\neq l\geq K_p+1}P_k(m)P_l(m)t_kt_l + o(\frac{1}{\sqrt{m}}) \\
&=\sum_{k=1}^{K_p}\gamma_k(e^{it_k}-1) - \frac{1}{2}\sum_{k=K_p+1}^{K+1}\bar{P}_k^2t_k^2 + \sum_{k\neq l\geq K_p+1}\bar{P}_k\bar{P}_l t_k t_l
\end{aligned} \tag{12}$$

This is the characteristic function of independent Poisson random variables and a vector Gaussian random vector. Since convergence of characteristic functions imply convergence in distribution [23], we get the lemma. □

Theorem 3. *We have:*

$$mD(P(m)\|\hat{P}) \xrightarrow{D} Y$$
$$Y = \sum_{k=1}^{K_p}X_k + \sum_{k=K_p+1}^{K}Y_k^2 \tag{13}$$

where

- *The X_k and Y_k are all independent and*

$$X_k = \gamma_k\log\left(\frac{\gamma_k}{\psi_k+\alpha}\right) + \frac{1}{\ln 2}(\psi_k+\alpha-\gamma_k) \tag{14}$$

  with $\psi_k \sim$ Pois(γ_k).
- *The $Y_k \sim \mathcal{N}\left(0, \frac{1}{2\ln 2}\right)$.*

Proof. We can expand relative entropy as follows

$$\begin{aligned}
&D(P||\hat{P}) \\
&= \sum_{i=1}^{K_p} P_i \log\left(\frac{P_i}{\hat{P}_i}\right) + \sum_{i=K_p+1}^{K} P_i \log\left(\frac{P_i}{\hat{P}_i}\right) \\
&+ \left(1 - \sum_{i=1}^{K_p} P_i - \sum_{i=K_p+1}^{K} P_i\right) \\
&\times \log\left(\frac{1 - \sum_{i=1}^{K_p} P_i - \sum_{i=K_p+1}^{K} P_i}{1 - \sum_{i=1}^{K_p} \hat{P}_i - \sum_{i=K_p+1}^{K} \hat{P}_i}\right) \\
&= \sum_{i=1}^{K_p} P_i \log\left(\frac{P_i}{\hat{P}_i}\right) + \frac{1}{\ln 2}(\hat{P}_i - P_i) + o\left(\frac{1}{m}\right) \\
&+ \sum_{i=K_p+1}^{K} P_i \log\left(\frac{P_i}{\hat{P}_i}\right) \\
&+ \left(1 - \sum_{i=K_p+1}^{K} P_i\right) \log\left(\frac{1 - \sum_{i=K_p+1}^{K} P_i}{1 - \sum_{i=K_p+1}^{K} \hat{P}_i}\right) \\
&= \sum_{i=1}^{K_p} P_i \log\left(\frac{P_i}{\hat{P}_i}\right) + \frac{1}{\ln 2}(\hat{P}_i - P_i) + o\left(\frac{1}{m}\right) \\
&+ \frac{1}{2\ln 2}(\hat{\mathbf{P}} - \mathbf{P})^T \mathbf{C}^{-1} (\hat{\mathbf{P}} - \mathbf{P}) + o\left(\frac{1}{m}\right) \qquad (15)
\end{aligned}$$

where Let $\mathbf{P} = [P_{K_p+1}, \ldots, P_K]^T$, $\hat{\mathbf{P}} = [\hat{P}_{K_p+1}, \ldots, \hat{P}_K]^T$ and

$$\mathbf{C} = \text{diag}(\mathbf{P}) - \mathbf{P}\mathbf{P}^T)$$

is the covariance matrix of the multinomial distribution. The first terms of (15) converges towards (14) in distribution while the second term converges towards a (scaled) χ^2 distribution. Using the Taylor series expansion for convergence in distribution can be done as in [24, Theorem 3.3.A]

□

As a consequence of the lemma we have

$$\lim_{m\to\infty} P\left(D(P(m)\|\hat{P}) > \frac{b}{m}\right) = P(Y > b)$$

Theorem 4. *A lower bound for a $K+1$ alphabet source is*

$$a(m, P_e, K) \geq \frac{F^{-1}_{\chi^2_K}(1 - P_e)}{2m \ln 2} + o\left(\frac{1}{m}\right) \qquad (16)$$

where $F_{\chi^2_K}$ is the CDF for a χ^2-distribution with K degrees of freedom.

Proof. Any estimator of the coding probability can be written as $\hat{P} = f_m(\check{P})$, where $\check{P}$ is the maximum likelihood estimator. Equivalently, $\hat{P} = \check{P} + g_m(\check{P})$. Then

$$E(m, a) = \sup_P P(D(P||\check{P} + g_m(\check{P})) > a) \qquad (17)$$

For the converse we may assume a restricted class of distributions. We consider distributions P with $P_k > \varepsilon$ for some small $\varepsilon > 0$. Then in Lemma 2 none of the components can converge towards a Poisson distribution, and we achieve a purely Gaussian limit. It is clear that we must have $g_m(\check{P}) \xrightarrow{D} 0$ as $m \to \infty$, as otherwise we cannot get $a(m, P_e) \to 0$. Now in (15) we then get

$$\begin{aligned}
&mD(P||\hat{P}) \\
&= \frac{1}{2\ln 2}(\sqrt{m}(\check{\mathbf{P}} - \mathbf{P}) + \sqrt{m} g_m(\check{\mathbf{P}}))^T \mathbf{C}^{-1} \\
&\times (\sqrt{m}(\check{\mathbf{P}} - \mathbf{P}) + \sqrt{m} g_m(\check{\mathbf{P}})) + \epsilon\left(\frac{1}{m}\right) \qquad (18)
\end{aligned}$$

We will argue that $\sqrt{m} g_m(\check{P}) \xrightarrow{D} 0$ as $m \to \infty$. First, we must have $\limsup_{m\to\infty} \sqrt{m} g_m(P)$ bounded, as otherwise we cannot get $a(m, P_e) \to 0$. From this follows that $\sqrt{m} g_m(\check{P}) \xrightarrow{D} g(P)$ for some function $g(P)$. Thus, (18) converges to a possible non-central χ^2_K distribution. But among those, the central χ^2_K distribution has the smallest tail probability, i.e., $g(P) \equiv 0$, which gives the theorem.

□

What remains is how to choose K_p in (13). All our numerical experiments show that the maximum is obtained for $K_p = K$, but at the moment we do not have a proof of this. So, a numerical computations in principle requires trying all values of K_p between 0 and K, which is feasible (but eventually unnecessary). Fig. 1 numerically compares upper and lower bounds. One can see that one can use the lower bouund as a good approximation of performance, and the lower bound is straightforward to calculate.

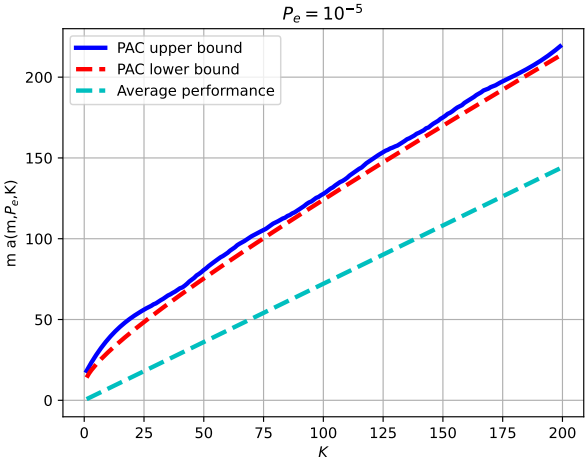

Fig. 1. Upper and Lower bounds for $K+1$ alphabet IID. Average performance is (4)

III. Learning for FSM

We consider binary sequences generated by FSM [25]. We will first discuss universal source coding. Universal coding

for FSM was first considered in [25]. The coding is done as follows (slightly changed from [25]). The coder first transmits the order K of the FSM, with Rissanen's coder for the integers [26], which can be done in $\log^* K + c$ bits. It then transmits which FSM of order K it uses, which can be done in $\log K^3$ bits. Finally, it encodes the sequence with the coder in [25]. We then define the minimax redundancy as

$$R_l^+(K) = \min_L \max_{f \in \text{FSM}(K), \boldsymbol{\theta}} \frac{1}{l} E[L(X^l)] - H_{f,\boldsymbol{\theta}}(X) \quad (19)$$

While [25] did not directly consider this criterion, we can conclude that

$$R_l^+(K) = \frac{K}{2l} \log l + O\left(\frac{1}{l}\right)$$

We now turn to learned coding of FSM. We adopt a strategy inspired by universal coding. Given the training data, the learning algorithm trains for all possible FSMs, in principle for $K = 1, 2, \ldots \infty$; however, it does not decide on a model. Only when it is presented with a test sequence is the model decided: given a test sequence, the encoder finds the FSM giving the shortest codelength, and then informs the decoder which FSM it used. As for universal coding this can be done with $\log^* K + c + \log K^3$ bits. We now define

$$R_l^+(m, K) = \min_L \max_{f \in \text{FSM}(K), \boldsymbol{\theta}} \frac{1}{l} E[L(X^l; X^m)] - H_{f,\boldsymbol{\theta}}(X) \quad (20)$$

and

$$a(m, l, P_e, K) = \min_L \max_{f \in \text{FSM}(K), \boldsymbol{\theta}} \max_a P(\frac{1}{l} E[L(X^l; X^m)] - H_{f,\boldsymbol{\theta}}(X) > a) \le P_e \quad (21)$$

which is the PAC criterion.

The "trick" of deciding the model based on each individual test sequence is something specific to coding that cannot be done for usual ML problems like classification. If K is large, the overhead is quite modest. It avoids a complicated problem of model selection, which is an unsolved problem (i.e., active research problem), with solutions like structural risk minimization [20].

Notice that $a(K, m, l, P_e)$ depends on K, as does the performance for universal source coding. The supremum over K is infinite in either case, and therefore does not make much sense. However, in PAC learning one would also like to get a universal bound on m, and (21) is therefore most useful to give insight into performance. For practical implementation, non-uniform PAC might be more useful.

The training consists of n sequences of length s, $ns = m$. We assume the FSM starts in a specific starting state. The implication is that all training sequences and test sequences start in the same state. As a consequence, for analysis purposes only $l = s$ makes sense, i.e., each training sequence is the same length as the test sequences. If $s < l$, some states might not be seen in the training. On the other hand, any steps $s > l$ might not tell anything about the first l steps. The latter is because the performance measure is a minimax criterion. For many FSM having $s > l$ does help training. But not in the worst case. We will therefore assume $l = s$.

Let π_t denote the expected proportion of time the state spends in state t (which is not necessarily a stationary distribution). The redundancy for coding is

$$\sum_{t=1}^{K} \pi_t D(p_t \| \hat{p}_t)$$

Whether we consider R_l^+ or a we have to consider the worst case over $\boldsymbol{\pi}$ and $\mathbf{p}$. As $m \to \infty$ the total number of state visits becomes unlimited. However, the average number of visits to specific states could still stay finite, namely if $\limsup_{m\to\infty} m\pi_t(m) < \infty$.

We will first consider the case when all states are visited infinitely many times, and later the case when some states are visited finitely many times. The former case can be specified as follows: there exists some small $\varepsilon > 0$ so that $\boldsymbol{\pi} > \varepsilon$.

Let m_t denote the number of visits to state t over all training sequences. We then have

$$\frac{m_t}{m} \xrightarrow{P} \pi_t > 0$$

Let $\bar{m}_t = \lfloor (\pi_t - \epsilon)m \rfloor$ for some small $\epsilon < \varepsilon$. We use the following genie inhibited training scheme for the achievable rate

- if $m_t > \bar{m}_t$, only the first $\bar{m}_t$ visits to state t is used for estimation of p_t.
- if for any t, $m_t < \bar{m}_t$ the genie adds $\bar{m}_t - m_t$ visits to state t. But at the same time the whole training is declared a failure.

We use (10),

$$\hat{p}_t = \frac{k_t + \alpha}{m_t + 2\alpha} = \frac{k_t + \alpha}{\bar{m}_t + 2\alpha} \quad (22)$$

Here k_t is the number of ones in state t. Because of the genie, the k_t are independent and binomially distributed, $B(p_t, \bar{m}_t)$.

Let E_2 be a training failure due to not enough visits to a state, that is

$$P(E_2) = P(\exists t : m_t < \bar{m}_t). \quad (23)$$

We then have

Lemma 5. *With $P(E_2)$ given by (23) we can bound*

$$P(E_2) \le K \exp(-3n\epsilon^2)$$

for any FSM with K states.

Proof. Let $m_{t,i}$ be the number of visits to state t in the i-th training sequence; the total number of visits then is $m_t = \sum_{i=1}^n m_{t,i}$, with the $m_{t,i}$ independent for fixed t and $1 \le m_{0,i} \le s$, $0 \le m_{t,i} \le s-1$, $t > 0$. We can write

$$P(E_2) = P(\exists t : m_t < \bar{m}_t) \le \sum_{t=1}^{K} P(m_t < \bar{m}_t). \quad (24)$$

We use Hoefding's inequality [23],

$$P(m_t < (\pi_t - \epsilon)m) = P\left(m_t - \pi_t m < -\epsilon m\right) \leq \exp\left(-2\frac{\epsilon^2 m^2}{n(s-1)^2}\right).$$

for $t > 0$ – similar for $t = 0$. □

For FSM the measures of performance become

$$R_l^+(m,K) = \sup_{\text{FSM}(K)} E\left[\sum_{t=1}^{K} \pi_t D(p_t\|\hat{p}_t)\right] \quad (25)$$

$$E(m,a,K) = \sup_{\text{FSM}(K)} P\left(\sum_{t=1}^{K} \pi_t D(p_t\|\hat{p}_t) \geq a\right). \quad (26)$$

Theorem 6. *Consider an FSM(K) model with $\boldsymbol{\pi} > \varepsilon$. For the estimator (22), with $\alpha = \alpha_0$ (5) we get*

$$R_l^+(m,K) = \frac{K\alpha_0}{m\ln 2} + o\left(\frac{1}{m}\right) \quad (27)$$

for any $\varepsilon > 0$, while a lower bound is

$$R_l^+(m,K) \geq \frac{K}{2m\ln 2} + o\left(\frac{1}{m}\right). \quad (28)$$

Proof. For the achievable rate, we use the bad genie. Whenever E_2 happens, for reasons of symmetry the learning algorithm still uses (22). The encoder and decoder, however, are told that the whole training failed, and they therefore transmit the data uncoded in l bits; this gives an upper bound,

$$\begin{aligned}\lim_{m\to\infty} mR_l^+(m,K) &\leq \lim_{m\to\infty} m \sup_{\mathbf{p},\boldsymbol{\pi}>\varepsilon} E\left[\sum_{t=1}^{K} \pi_t D(p_t\|\hat{p}_t)\right] \\ &\leq \sum_{t=1}^{K} \lim_{m\to\infty} m \sup_{\pi_t>\varepsilon} \pi_t \sup_{p_t} E\left[D(p_t\|\hat{p}_t)\right] \\ &\leq \frac{K}{\ln 2} \sup_{\pi_t>\varepsilon} \pi_t \frac{\alpha_0}{(\pi_t - \epsilon)} + lP(E_2) \\ &\leq \frac{K}{\ln 2}\alpha_0 \frac{\varepsilon}{\varepsilon - \epsilon} + lK\exp(-3n\epsilon^2) \quad (29)\end{aligned}$$

The condition for the first two terms in (29) to converge to α_0 is just that $\epsilon \to 0$. The condition for the last term to converge to zero is just that $n \to \infty$ and that ϵ does not converge to zero too fast. We can always choose a suitable ϵ to satisfy this.

We now turn to the converse. In (25) the maximization is over FSM(K). In the converse we therefore need to choose $\mathbf{p}, \boldsymbol{\pi}$ corresponding to a realizable FSM(K). We use a ring shaped FSM where both arrows from state t go to state $t+1$ (and from state K to 1). Then $\pi_t = \frac{1}{K}$ while $\mathbf{p}$ can be arbitrary. The number of visits to each state is $\lfloor \frac{m}{K} \rfloor$, and it is clear that the $\hat{p}_t$ are independent. Then

$$\begin{aligned}\lim_{m\to\infty} mR(m) &= \lim_{m\to\infty} \inf_{\hat{\boldsymbol{\pi}}} m \sup_{\mathbf{p}} E\left[\sum_{t=1}^{K} \frac{1}{K} D(p_t\|\hat{p}_t)\right] \\ &= \lim_{m\to\infty} \sum_{t=1}^{K} E\left[D(p_t\|\hat{p}_t)\right] \\ &\geq \frac{K}{2\ln 2} \quad (30)\end{aligned}$$

The last inequality is due to [17, Theorem 2]. □

For PAC performance we have the following result

Theorem 7. *For sufficiently large b and for any $\varepsilon > 0$, the estimator (22) satisfies*

$$\lim_{m\to\infty} \sup_{\mathbf{p},\boldsymbol{\pi}>\varepsilon} P\left(\sum_{t=1}^{K} \pi_t D(p_t\|\hat{p}_t) \geq \frac{b}{m}\right) = \sup_{\gamma>0} P(Y > b)$$

for both $n, l \to \infty$, with

$$Y = \sum_{k=1}^{K_p} X_k + \sum_{k=K_p+1}^{K} Y_k^2 \quad (31)$$

where

- *The X_k and Y_k are all independent and*

$$X_k = \gamma_k \log\left(\frac{\gamma_k}{\psi_k + \alpha}\right) + \frac{1}{\ln 2}(\psi_k + \alpha - \gamma_k) \quad (32)$$

  with $\psi_k \sim$ Pois(γ_k).
- *The $Y_k \sim \mathcal{N}\left(0, \frac{1}{2\ln 2}\right)$.*

Proof. Define the event

$$E_1 = \left\{\sum_{t=1}^{K} \pi_t D(p_t\|\hat{p}_t) \geq a\right\}$$

while E_2 is still the event that the genie flags a sequence as invalid. These events are not independent, but we can upper bound the probability of training error by

$$E(m,a,K) \leq \sup_{p_0,p_1} (P(E_1) + P(E_2)).$$

Of course, the addition of extra artificial data can decrease $P(E_1)$, but whenever artificial data is added E_2 occurs, and the total error probability is not decreased. Here

$$P(E_1) = P\left(\sum_{t=1}^{K} \pi_t D(p_t\|\hat{p}_t) \geq a\right)$$

Notice that the $\hat{p}_t$ are independent with the way the (augmented) data set is generated. From Theorem 3 with $K = 1$ we have

$$m\pi_t D(p_t\|\hat{p}_t) \xrightarrow{D} \frac{\pi_t}{\pi_t - \epsilon} X_t$$

where X_t is given by (32) in the Poisson case or otherwise χ_1^2. □

Theorem 8. *For any FSM(K) a lower bound is*

$$a(m,P_e,K) \geq \frac{F_{\chi_K^2}^{-1}(1-P_e)}{2m\ln 2} + o\left(\frac{1}{m}\right) \quad (33)$$

where $F_{\chi_K^2}$ is the CDF for a χ^2-distribution with K degrees of freedom.

Proof. We allow any estimator of the state probability $f_m(\hat{p}_t)$. We consider the same ring FSM as for the proof of Theorem 6. Since we are proving the converse we can assume that $\mathbf{p} > \epsilon$

for some $\epsilon > 0$. In that case we are in the CLT regime. We can now apply Theorem 4) to the case $K = 1$ and we get

$$mD(p_t \| f_m(\hat{p}_t)) \xrightarrow{D} \frac{K}{\ln 2}\chi_1^2$$

Since the $\hat{p}_t$ are independent for the ring FSM, we have

$$\sum_{t=1}^{K} \pi_t D(p_t \| f_m(\hat{p}_t)) \xrightarrow{D} \frac{K}{\ln 2}\chi_K^2$$

which proves the theorem. □

We notice that the results are the same as for the K-alphabet iid case, although the proofs are quite different. Fig. 1 therefore also applies to the FSM case.

We now consider the case when some states are only visited finitely many times (on average), that is $\lim_{m\to\infty} m\pi_t(m) = c < \infty$. In that case $\hat{p}_t$ is learned poorly and $D(p_t\|\hat{p}_t)$ does not converge to zero. Since π_t is also the ratio state t is visited in training sequences and $\pi_t(m) \to 0$, training is still possible. The question is if the states with finitely many visits or those with infinitely many visits dominate performance. For simplicity we will only analyze average performance.

Let $0 \leq m_{t,i} \leq s$ be the the number of visits to state t in the i-the training sequence with $m_t = \sum_{i=1}^{n} m_t, i$ and the $m_{t,i}$ iid with respect to i. Since $E[m_t] = c$, $E[m_{t,i}] = \frac{c}{n}$. Explicitly

$$cE[D(p\|\hat{p})] = c\sum_{j=0}^{\infty} P(m_t = j)E[D(p\|\hat{p}(j))]$$

Here $E[D(p\|\hat{p}(j))]$ is a decreasing function of j for any p. To maximize $E[D(p\|\hat{p})]$ as much as possible mass should be at $j = 0$, subject to the constraint $E[m_t] = c$. This is achieved if $m_{t,i}$ is binomial with $P(m_{t,i} = 1) = \frac{c}{n}$. Then $m_t \xrightarrow{D} \mathbb{P}_c$, the Poisson distribution. We can now maximize $cE[D(p\|\hat{p})] = c\sum_{j=0}^{\infty} \mathbb{P}_c(j)E[D(p\|\hat{p}(j))]$ over c and p, which much be done numerically, to get

$$p = 0, c = 3.38$$
$$\sup_{c,p} cE[D(p\|\hat{p})] = 0.79$$

We then get the bound (since at least one state must have infinitely many visits)

$$R_l^+(m, K) = \frac{0.79(K-1)}{m \ln 2} + \frac{\alpha_0}{m \ln 2} + o\left(\frac{1}{m}\right) \qquad (34)$$

which can be compared with (27): the performance is still $O\left(\frac{1}{m}\right)$ but with the proportionality constant increased from 0.509 to 0.79.

IV. Concluding remarks

While we have treated the K-alphabet case and the binary FSM case separately, one can see from the similarity of the proofs that these can readily be combined (for the case of infinite visits to FSM states). The performance just depends on the total degrees of freedom, and for example Fig. 1 can be seen as a plot vs. degrees of freedom.

References

[1] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *Information Theory, IEEE Transactions on*, vol. 23, no. 3, pp. 337 – 343, may 1977.

[2] ——, "Compression of individual sequences via variable-rate coding," *Information Theory, IEEE Transactions on*, vol. 24, no. 5, pp. 530 – 536, sep 1978.

[3] T. Cover and J. Thomas, *Information Theory, 2nd Edition*. John Wiley, 2006.

[4] J. Schmidhuber and S. Heil, "Sequential neural text compression," *IEEE Transactions on Neural Networks*, vol. 7, no. 1, pp. 142–146, 1996.

[5] J.-L. Zhou and Y. Fu, "Scientific data lossless compression using fast neural network," in *ISNN 2006*, 2006, pp. 1293–1298.

[6] A. Kattan, "Universal intelligent data compression systems: A review," in *2010 2nd Computer Science and Electronic Engineering Conference (CEEC)*, Sept 2010, pp. 1–10.

[7] M. V. Mahoney, "Fast text compression with neural networks." in *FLAIRS Conference*, 2000, pp. 230–234.

[8] ——, "Adaptive weighing of context models for lossless data compression," Texas A&M University, Tech. Rep., 2005.

[9] D. Cox, "Syntactically informed text compression with recurrent neural networks," *CoRR*, vol. abs/1608.02893, 2016. [Online]. Available: http://arxiv.org/abs/1608.02893

[10] K. Tatwawadi, "Deepzip: Lossless compression using recurrent networks," Stanford University, Tech. Rep.

[11] G. Toderici, D. Vincent, N. Johnston, S. J. Hwang, D. Minnen, J. Shor, and M. Covell, "Full resolution image compression with recurrent neural networks," *CoRR*, vol. abs/1608.05148, 2016. [Online]. Available: http://arxiv.org/abs/1608.05148

[12] Q. Li and Y. Chen, "Lossy source coding via deep learning," in *2019 Data Compression Conference (DCC)*. IEEE, 2019, pp. 13–22.

[13] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Deep convolutional autoencoder-based lossy image compression," in *2018 Picture Coding Symposium (PCS)*. IEEE, 2018, pp. 253–257.

[14] D. Liu, Y. Li, J. Lin, H. Li, and F. Wu, "Deep learning-based video coding: A review and a case study," *ACM Comput. Surv.*, vol. 53, no. 1, Feb. 2020. [Online]. Available: https://doi.org/10.1145/3368405

[15] Y. Hershkovits and J. Ziv, "On fixed-database universal data compression with limited memory," *IEEE Transactions on Information Theory*, vol. 43, no. 6, pp. 1966–1976, Nov 1997.

[16] A. Høst-Madsen, "Bounds for learning lossless source coding," in *ISIT'2021, Melbourne, Australia, July 12-20, 2021*, 2021.

[17] R. E. Krichevskiy, "Laplace's law of succession and universal encoding," *IEEE Transactions on Information Theory*, vol. 44, no. 1, pp. 296–303, Jan 1998.

[18] M. Falahatgar, A. Orlitsky, V. Pichapati, and A. T. Suresh, "Learning markov distributions: Does estimation trump compression?" in *2016 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2016, pp. 2689–2693.

[19] Y. Hao, A. Orlitsky, and V. Pichapati, "On learning markov chains," in *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 648–657. [Online]. Available: http://papers.nips.cc/paper/7345-on-learning-markov-chains.pdf

[20] V. N. Vapnik, *Statistical Learning Theory*. John Wiley, 1998.

[21] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*. MIT Press, 2018.

[22] R. Krichevsky and V. Trofimov, "The performance of universal encoding," *IEEE Transactions on Information Theory*, vol. 27, no. 2, pp. 199–207, Mar 1981.

[23] G. R. Grimmett and D. R. Stirzaker, *Probability and Random Processes, Third Edition*. Oxford University Press, 2001.

[24] R. J. Serfling, *Approximation Theorems of Mathematical Statistics*. Wiley-Interscience, 2001.

[25] J. Rissanen, "Complexity of strings in the class of markov sources," *Information Theory, IEEE Transactions on*, vol. 32, no. 4, pp. 526 – 532, jul 1986.

[26] ——, "A universal prior for integers and estimation by minimum description length," *The Annals of Statistics*, no. 2, pp. 416–431, 1983.