

Universal Variable-to-Fixed Length Source Codes

Karthik Visweswariah, *Member, IEEE*, Sanjeev R. Kulkarni, *Senior Member, IEEE*, and Sergio Verdú, *Fellow, IEEE*

Abstract—A universal variable-to-fixed length algorithm for binary memoryless sources which converges to the entropy of the source at the optimal rate is known. We study the problem of universal variable-to-fixed length coding for the class of Markov sources with finite alphabets. We give an upper bound on the performance of the code for large dictionary sizes and show that the code is optimal in the sense that no codes exist that have better asymptotic performance. The optimal redundancy is shown to be $H \log \log M / \log M$ where H is the entropy rate of the source and M is the code size. This result is analogous to Rissanen's result for fixed-to-variable length codes. We investigate the performance of a variable-to-fixed coding method which does not need to store the dictionaries, either at the coder or the decoder. We also consider the performance of both these source codes on individual sequences. For individual sequences we bound the performance in terms of the best code length achievable by a class of coders. All the codes that we consider are prefix-free and complete.

Index Terms—Data compression, entropy, minimum description length, Tunstall algorithm, universal coding, variable-fixed length codes.

I. INTRODUCTION

CONSIDER a source X which takes values in a finite set A . A variable-to-fixed length code is a finite dictionary $\mathcal{D} \in A^\infty$ which is used to parse an infinite sequence into variable-length phrases. Each phrase is then coded into a fixed number of bits. The number of output bits per phrase depends on the size of the dictionary. It is assumed that the dictionary is complete and prefix-free. A complete dictionary is one such that any infinite sequence will have a prefix in the dictionary. As the source outputs a string we wait until the string matches one in \mathcal{D} , we then code the string with a fixed number of bits, $\lceil \log_2 |\mathcal{D}| \rceil$. The dictionary being prefix-free ensures that there is a unique way to parse every source string. We note that unlike the completeness assumption, the prefix-free assumption is not necessary to get a well-defined variable-to-fixed length code. It is also possible in certain cases to improve performance using a code which is not prefix-free [5]. For the dictionary \mathcal{D} to be a good source code it is required that the expected input phrase length be as high as possible for a fixed output size (i.e., a fixed dictionary size).

Manuscript received July 7, 1999; revised August 31, 2000. This work was supported in part by the National Science Foundation under Grants NYI Award IRI-9457645 and NCR 9523805.

K. Visweswariah was with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544 USA. He is now with IBM T. J. Watson Research Center, Yorktown Heights, NY 10598 USA (e-mail: karthik@ee.princeton.edu).

S. R. Kulkarni and S. Verdú are with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544 USA (e-mail: kulkarni@ee.princeton.edu; verdu@ee.princeton.edu).

Communicated by N. Merhav, Associate Editor for Source Coding.
Publisher Item Identifier S 0018-9448(01)02716-X.

The problem of variable-to-fixed length coding was first looked at by Tunstall in [9], where an algorithm to find the dictionary of a given size that maximizes the expected phrase length for a given memoryless source is found. The method requires the source distribution to be known. The problem of finding good dictionaries for coding Markov sources was investigated in [7] and [6], where codes which achieve the entropy rate for sources with memory were given. Though the coding rate achievable is still the entropy it is known that variable-to-fixed length codes perform better than fixed-to-variable length codes in certain ways [3]. Tjalkens and Willems in [8] solved the universal coding problem for the class of binary-valued and independent and identically distributed (i.i.d.) sources. They show that their method converges to the entropy at an optimal rate. Here a more general universal coding problem is investigated in which the class of sources consists of Markov sources that take values in a finite set of size possibly greater than two.

In Section II, the problem looked at in this paper is formally defined. In Section III, a variable-to-fixed source code is found which is shown to achieve a redundancy of $Hk(a-1) \log \log M / 2 \log M$, where H is the entropy rate of the source, M is the size of the dictionary, a is the alphabet size, and k is the number of states in the Markov source. Throughout, we use a logarithm to an arbitrary base, and we specifically mention the base when we need to use a particular base. In Section IV, a converse for the above achievability result is given by showing that the set of parameters for which a code can achieve a better redundancy than the code in Section III has vanishingly small volume. These results are a generalization of [8] where the optimal redundancy was given as $H \log \log M / 2 \log M$ for binary, memoryless sources (i.e., $k = 1$ and $a = 2$). They are also analogous to the results in [4] for fixed-to-variable length codes, the difference in the two being the entropy multiplying the redundancy term. In Section V, we consider the performance of this coding scheme on individual sequences. In Section VI, we investigate the performance of a variable-to-fixed length coding method given in [1], which could be used to implement a practical coding method since it does not need the dictionary to be stored at the encoder or the decoder. The redundancy that we show for this method is not optimal but is close to being optimal. This method has the added advantage that it can be adapted efficiently and can thus give smaller redundancies than predicted by our lower bound for coding long strings of sequences. The lower bound that we give holds for fixed coders and decoders.

II. PROBLEM FORMULATION AND PRELIMINARIES

Let the source output alphabet be $A = \{1, 2, \dots, a\}$ and the set of states of the Markov source be $S = \{1, 2, \dots, k\}$.

The Markov source is fully defined by the next-state function $T: A \times S \mapsto S$ and the letter probabilities $P(\cdot|\cdot): A \times S \mapsto [0, 1]$. At each time the source outputs a letter from A and moves to another state. The next state depends on the present state and the output and is given by the next state function T . The probability of a particular output $i \in A$ occurring when we are in state $s \in S$ is $P(i|s)$. We can consider the P to be a matrix with the entry in row i and column j being $P(i|j)$. Let X_i, S_i be the random variables denoting the output and state, respectively, at time i . The distribution of the Markov source is defined by

$$\Pr(X_n = x | S_n = s, X_{n-1}, S_{n-1}, \dots) = P(x|s).$$

We will use A^* to denote the set of strings of finite length with letters from the alphabet A . We will use $P(x^*|s)$ for the probability of a string $x^* \in A^*$ occurring starting in the state s . Note that this probability depends on P and T . We assume throughout this paper that the underlying source is irreducible and aperiodic. Also note that Markov chains of a finite order are a special case of the Markov source as we have defined it. The most general case that we consider is when we do not know T and P . In most cases, we will discuss the situation when we know T and then indicate how the results change if T is not known. Some of the methods we give are computationally efficient only if we assume that the class of sources is the class of tree sources. Though most of the results can be extended to the case when T is not known, we state results for less general classes if the restriction makes the algorithm more efficient. We indicate specifically when the results do not hold for more general classes of sources.

We consider the case when the encoder and the decoder can keep track of the state of the source (T and initial state s_0 are known). This allows the encoder and the decoder to choose the dictionary to be used depending on what state the source is in. We cannot do this if we do not know T since the coder and decoder will not be able to keep track of the current state of the source. When the methods described can be generalized beyond this scenario we indicate how to do so. When the state can be tracked, the encoder \mathcal{D} is a collection of $|S|$ dictionaries. A dictionary \mathcal{D}_s is used to parse the source output if the source is in state s . When we do not know the state that the source is in, the encoder will consist of just one dictionary \mathcal{D} .

We now describe the performance of a collection of dictionaries \mathcal{D}_s with a maximum of M segments in a dictionary. The performance of this collection of dictionaries will be determined by the expected output length relative to $\log M$. The case when we only have one dictionary is easily dealt with by taking \mathcal{D}_s to be independent of s . To define the compression ratio precisely we first define a segment source as in [7]. The starting state of a segment and the segment determine the state at the start of the next segment. Also, the probability of a segment occurring given the present state and the past states and segments depends only on the present state. Thus, the segment source is a Markov source induced by the original Markov source and \mathcal{D} . Let X_1^*, X_2^*, \dots denote the output of the induced segment source. The state set S of the segment source need not be irreducible even if it is irreducible for the original source. Let $\mathcal{G} = \{G_r: r = 1, 2, \dots, R\}$ be the set of all the irreducible

subsets of S . The performance of the code is governed by the expected average phrase length given by

$$E(L) = \lim_{n \rightarrow \infty} E \frac{1}{n} \sum_{i=1}^n l(x_i^*).$$

This is also given by (see [7])

$$E(L) = \sum_{i=1}^R \Pr(G_i) \sum_{s \in G_i} q(s|G_i) \sum_{x^* \in \mathcal{D}_s} l(x^*) P(x^*|s)$$

where $\Pr(G_i)$ is the probability of the source entering the irreducible set G_i , and $q(s|G_i)$ is the steady-state probability of $s \in G_i$ given that the source has entered G_i . Our goal is to find a collection of dictionaries \mathcal{D}_s independent of the letter probabilities P such that the compression $\max_s \log |\mathcal{D}_s|/E(L)$ converges quickly (with increasing dictionary size) to the entropy of the source. This definition of compression is as in [7]. It differs from the definition in [11]. The two definitions are equivalent if the segment source is ergodic. For a given code, we can increase the number of codewords by a constant number to make the segment source ergodic (if the underlying source is irreducible). Since our results are asymptotic they also hold for the stricter definition. The argument that we can indeed make the segment source ergodic is as follows. Suppose that the number of states is k . Consider the code tree for a given state. We can add a tree with k leaves to one of the leaves of the code tree. Call these $l_1 \dots l_k$. Assume that in the underlying source we can go from one state to any other in at most m steps. We can find such an m if the underlying source is irreducible. We extend leaf l_i by the path needed to go from state that we are in at leaf l_i to state i . We can do this with a path of length no more than m . We can complete the tree with an addition of at most m leaves. So we can add at most $(m+1)k$ leaves to each tree and ensure that we can go to any state from any state in one step of the segment source.

III. ACHIEVABILITY

In this section, we will give a method to achieve optimal coding for a Markov source. As mentioned earlier, we assume that the encoder and the decoder know the next-state function though the letter probability function is unknown.

We first describe a method to form a complete, prefix-free dictionary given a distribution P . We assume that P satisfies

$$\sum_{x \in A} P(y^*x) = P(y^*)$$

and

$$\sum_{x \in A} P(x) = 1.$$

Define the dictionary \mathcal{D} by $x^* \in \mathcal{D}$ if $P(x^*) \leq c$ and no prefix of x^* satisfies this property. Clearly, \mathcal{D} is prefix-free, it is also complete if the distribution is such that $P(x_1^n)$ goes to zero as n goes to infinity for any infinite sequence x_1^∞ , where x_1^n denotes the prefix of length n of the infinite sequence x_1^∞ .

We will create dictionaries using the method described above for a finite but growing set of possible letter probability matrices and combine all these dictionaries to form a dictionary

which will work well for any possible letter probability matrix. The method we use for combining dictionaries was given in [6]. Suppose that we want to build a dictionary of size at most M . m will be an integer parameter in our scheme and will need to be chosen appropriately. Consider k column vectors of integers $\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_k$, each vector of size a . The vectors are chosen so that each entry of a vector is between 1 and m and the entries in a particular vector sum to $m+1$. Thus, $\mathbf{I}_j/(m+1)$ defines a distribution on A for each j . Let

$$Q_{\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_k} = [\mathbf{I}_1 \mathbf{I}_2 \dots \mathbf{I}_k] / (m+1).$$

Then Q is a valid letter probability matrix for each set of vectors chosen according to the restrictions above. Each valid Q along with an initial state s defines a distribution $Q_{\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_k}(\cdot|s)$. We will form $\mathcal{D}_s(\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_k)$ by using the procedure described above with

$$c = \frac{m^{k(a-1)}(m+1)}{M \min(i_1, i_2, \dots, i_k)}$$

where i_j is the minimum entry in vector \mathbf{I}_j .

We note that because of the way the dictionary is constructed we have

$$Q_{\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_k}(x^*|s) \geq c \frac{\min(i_1, i_2, \dots, i_k)}{m+1}$$

for $x^* \in \mathcal{D}_s(\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_k)$. Thus, since

$$\sum_{x^* \in \mathcal{D}_s(\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_k)} Q_{\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_k}(x^*|s) = 1$$

we have

$$|\mathcal{D}_s(\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_k)| \leq \frac{m+1}{c \min(i_1, i_2, \dots, i_k)}.$$

Substituting in the value of c we have used to form the dictionary we have

$$|\mathcal{D}_s(\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_k)| \leq \frac{M}{m^{k(a-1)}}. \quad (1)$$

We have so far described how to form the dictionary for a particular letter probability matrix Q depending on $\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_k$. We can thus obtain dictionaries for each of the different possible valid vectors $\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_k$. We will combine these dictionaries by taking a union of all the dictionaries and then removing elements so as to make the union prefix-free. Let the dictionary so obtained be \mathcal{D}_s . The idea is that if we take m large enough then, for any parameter value, we will have a dictionary tuned to some parameter close to the true parameter and so we do not lose much in performance. On the other hand, taking large m imposes a larger penalty when combining the dictionaries to form one dictionary. We will now show an upper bound to the compression achieved by this scheme with $m = \sqrt{\log M}$.

Let us denote by $H(P)$ the entropy rate of the source, which is given by

$$H(P) = \sum_{s \in S} q(s) H(P(\cdot|s))$$

where q is the steady-state distribution of the Markov source.

Theorem 1: For any $\delta \geq 0$ and for any letter probability function P with $0 < P(x|s) < 1$ for each $x \in A, s \in S$, there exists $M(P, \delta) < \infty$ such that

$$\begin{aligned} R &= \frac{\max_s \log |\mathcal{D}_s|}{E(L)} \\ &\leq H(P) \left(1 + (1 + \delta) \frac{k(a-1) \log \log M}{2 \log M} \right) \end{aligned}$$

for all $M \geq M(P, \delta)$.

Proof: First let us upper-bound $|\mathcal{D}_s|$. Now

$$\mathcal{D}_s \subseteq \bigcup \mathcal{D}_s(\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_k)$$

where the union is over all valid vectors $\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_k$. This implies that

$$|\mathcal{D}_s| \leq \left| \bigcup \mathcal{D}_s(\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_k) \right|.$$

Because of the constraint that the components of each integer vector should add up to $m+1$ we have that the number of valid sets of vectors is bounded by $m^{k(a-1)}$. Thus, we have, using (1), that $|\mathcal{D}_s| \leq M$ for each $s \in S$ and so $R \leq \log M/E(L)$. We will now lower-bound $E(L)$.

Let

$$D(P, Q) = \sum_{s \in S} q(s) D(P(\cdot|s) \| Q(\cdot|s))$$

where q is the steady-state distribution of the Markov source.

We have from [7]

$$\begin{aligned} E(L)(H(P) + D(P, Q)) \\ &= \sum_{r=1}^R \Pr(G_r) \sum_{s \in G_r} q(s|G_r) \sum_{x^* \in \mathcal{D}_s} P(x^*|s) \log \frac{1}{Q(x^*|s)} \end{aligned} \quad (2)$$

for any $P(\cdot|s)$ and $Q(\cdot|s)$. In the above equation, $\Pr(G_r)$ is the probability of the source entering the irreducible set G_r , and $q(s|G_r)$ is the steady-state probability of $s \in G_r$ given that the source has entered G_r . Let $P(\cdot|s)$ be the true letter probability matrix and $Q(\cdot|s)$ be the letter probability matrix which is component-wise closest to $P(\cdot|s)$. Let the vectors of integers which result in this Q be $\mathbf{I}_1^*, \mathbf{I}_2^*, \dots, \mathbf{I}_k^*$. For any x^* in \mathcal{D}_s we have (by the construction of \mathcal{D}_s) that

$$\log \frac{1}{Q(x^*|s)} \geq \log \frac{M \min(i_1^*, \dots, i_k^*)}{m^{k(a-1)}(m+1)}.$$

Thus, using (2) we have

$$E(L) \geq \frac{\log M - k(a-1) \log m + \log(p_{\min} - 1/m)}{H(P) + D(P, Q)} \quad (3)$$

for $m > 1/p_{\min}$ and where $p_{\min} = \min_{x, s} P(x|s)$.

Now we will upper-bound $D(P, Q)$. To do this, we will upper-bound $D(P(\cdot|s) \| Q(\cdot|s))$ for each $s \in S$, using the two lemmas below. Note that in Lemma 1, on the right-hand side (RHS) the argument of the divergence is a real number and not a distribution. By $D(p \| q)$ for $p, q \in [0, 1]$ we mean the divergence between the distributions $p, 1-p$ and $q, 1-q$.

Lemma 1: Consider two distributions P_1 and P_2 on a finite alphabet $A = \{1, 2, \dots, a\}$. Then

$$D(P_1(\cdot) \| P_2(\cdot)) = \sum_{j=1}^{a-1} D \left(\frac{P_1(j)}{\sum_{i=j}^a P_1(i)} \middle\| \middle\| \frac{P_2(j)}{\sum_{i=j}^a P_2(i)} \right) \sum_{i=j}^a P_1(i).$$

Proof of Lemma 1: The RHS equals

$$\sum_{j=1}^{a-1} \left[P_1(j) \log \frac{P_1(j) \sum_{i=j}^a P_2(i)}{P_2(j) \sum_{i=j}^a P_1(i)} + \sum_{i=j+1}^a P_1(i) \log \frac{\sum_{i=j+1}^a P_1(i) \sum_{i=j}^a P_2(i)}{\sum_{i=j+1}^a P_2(i) \sum_{i=j}^a P_1(i)} \right] \quad (4)$$

$$= \sum_{j=1}^{a-1} \left[P_1(j) \log \frac{P_1(j)}{P_2(j)} + \sum_{i=j}^a P_1(i) \log \frac{\sum_{i=j}^a P_2(i)}{\sum_{i=j}^a P_1(i)} - \sum_{i=j+1}^a P_1(i) \log \frac{\sum_{i=j+1}^a P_2(i)}{\sum_{i=j+1}^a P_1(i)} \right] \quad (5)$$

$$= \sum_{j=1}^a P_1(j) \log \frac{P_1(j)}{P_2(j)}. \quad (6)$$

Equation (4) follows by the definition of $D(p \| q)$. Equation (5) follows by rearranging the terms in (4). Equation (6), by noting

that if the second term in (5) is denoted by f_j then the third term is $-f_{j+1}$, and hence all but two of the terms cancel when we do the outer sum over j . \square

The use of this lemma is to write a divergence between distributions on an arbitrary finite alphabet in terms of divergences between distributions on an alphabet of size two.

Each divergence on the RHS can be bounded using the following lemma.

Lemma 2: For $\theta_1, \theta_2 \in (0, 1)$

$$D(\theta_1 \| \theta_2) \leq \frac{(\theta_1 - \theta_2)^2}{\theta_2(1 - \theta_2)} \log e.$$

Proof of Theorem 1 (Cont.): Using Lemmas 1 and 2 to bound the divergence we get the equation shown at the bottom of the page. The last inequality holds for $m > 1/p_{\min}$. Thus we have for sufficiently large m

$$D(P, Q) \leq \frac{a(a/m)^2}{p_{\min}(p_{\min} - 1/m)^2} \log e.$$

Taking $m = \sqrt{\log M}$ and using the upper bound on $D(P, Q)$ in (3) we have the equation shown at the bottom of the following page. The last inequality holds for sufficiently large M , i.e., for $M \geq M(P, \delta)$.

We have shown a scheme to code over a class of Markov sources when the coder and decoder can keep track of the state of the source. We can use the same idea to code when the next state function is not known. Given a finite set of states and a finite alphabet, the number of possible next-state functions is finite. For each next-state function T , we can build a collection of dictionaries $\mathcal{D}_s(T)$. We can then combine the dictionaries for

$$\begin{aligned} D(P(\cdot|s) \| Q(\cdot|s)) &\leq \sum_{j=1}^{a-1} \frac{\left[\frac{P(j|s)}{\sum_{i=j}^a P(i|s)} - \frac{Q(j|s)}{\sum_{i=j}^a Q(i|s)} \right]^2}{\left(\frac{Q(j|s)}{\sum_{i=j}^a Q(i|s)} \right) \left(1 - \frac{Q(j|s)}{\sum_{i=j}^a Q(i|s)} \right)} \sum_{i=j}^a P(i|s) \log e \\ &= \sum_{j=1}^{a-1} \frac{\left(P(j|s) \sum_{i=j+1}^a Q(i|s) - Q(j|s) \sum_{i=j+1}^a P(i|s) \right)^2}{\left(\sum_{i=j}^a P(i|s) \right) Q(j|s) \left(\sum_{i=j+1}^a Q(i|s) \right)} \log e \\ &\leq \sum_{j=1}^{a-1} \frac{\left(P(j|s) \sum_{i=j+1}^a (P(i|s) + 1/m) - (P(j|s) - 1/m) \sum_{i=j+1}^a P(i|s) \right)^2}{\left(\sum_{i=j}^a P(i|s) \right) (P(j|s) - 1/m) \left(\sum_{i=j+1}^a (P(i|s) - 1/m) \right)} \log e \\ &\leq \frac{a(a/m)^2}{p_{\min}(p_{\min} - 1/m)^2} \log e. \end{aligned}$$

the various states using the same scheme as before to get a dictionary $\mathcal{D}(T)$ with at most $M \times |S|$ words. We now combine these dictionaries for each next-state function to form one dictionary \mathcal{D} with at most $M \times |S| \times f(|S|, |A|)$ words. $f(|S|, |A|)$ denotes the number of next-state functions T possible when the set of states has size $|S|$ and the alphabet has size $|A|$. Now, as M gets large, the rate loss from combining these dictionaries becomes insignificant since $|S| \times f(|S|, |A|)$ is finite. Thus, the performance of this scheme is asymptotically equivalent to the scheme when the next state function is known. For practical dictionary sizes not knowing T could cause significant penalties since $f(|S|, |A|)$ grows rapidly with $|S|$ and $|A|$.

IV. CONVERSE

In this section, we will show a lower bound on the performance achievable by any code. We will assume that the next-state function is such that the states are irreducible. We will also represent the letter probability matrix P as a vector \mathbf{p} with $k(a-1)$ components with $p_{(i-1)(a-1)+j} = P(j|s=i)$, $i = 1, 2, \dots, k$ and $j = 1, 2, \dots, a-1$. Note that we do not need ka components because of the constraint that the rows of P must sum to 1. The converse cannot be for every parameter value \mathbf{p} but rather for “most” parameter values. This is because for any given code there might be some parameter values for which the rate is close to the entropy. What we show is that for any code the set of parameter values for which the code performs “well” is small. The converse result is stated precisely in the theorem below. The proof is a modification of the converse for i.i.d. sources given in [8] which in turn is analogous to Rissanen’s proof of the converse result fixed-to-variable length codes [4].

Theorem 2: For all $\delta > 0$ and any prefix-free variable-to-fixed length code with M large enough

$$R_{\mathbf{p}} \geq H(P) \left(1 + (1 - \delta) \frac{k(a-1) \log \log M}{2 \log M} \right)$$

for all valid $\mathbf{p} \in [0, 1]^{k(a-1)}$ except for those in a set whose volume goes to zero as M increases.

Proof: Fix $\delta > 0$ and $0 < \gamma < 1$. Let $l_{\min} = \lceil \gamma \log_a M \rceil$. Consider a code which consists of dictionaries \mathcal{D}_s for each $s \in S$ with $|\mathcal{D}_s| \leq M$ for each $s \in S$.

Let

$$A_{\mathbf{p}}^s \triangleq \Pr(\{x^* \in \mathcal{D}_s : l(x^*) < l_{\min}\} | s)$$

and

$$A_{\mathbf{p}} \triangleq \sum_r \Pr(G_r) \sum_{s \in G_r} q(s|G_r) A_{\mathbf{p}}^s$$

where as before $\Pr(G_r)$ is the probability of the source entering the irreducible set G_r , and $q(s|G_r)$ is the steady-state probability of $s \in G_r$ given that the source has entered G_r .

Also let

$$H_{\text{seg}}^s(P) \triangleq \sum_{x^* \in \mathcal{D}_s} P(x^*|s) \log \frac{1}{P(x^*|s)}$$

and

$$H_{\text{seg}}(P) \triangleq \sum_r \Pr(G_r) \sum_{s \in G_r} q(s|G_r) H_{\text{seg}}^s(P).$$

Now

$$\begin{aligned} H_{\text{seg}}^s(P) &= \sum_{x^*: l(x^*) < l_{\min}} P(x^*|s) \log \frac{1}{P(x^*|s)} \\ &\quad + \sum_{x^*: l(x^*) \geq l_{\min}} P(x^*|s) \log \frac{1}{P(x^*|s)} \\ &= h(A_{\mathbf{p}}^s) + A_{\mathbf{p}}^s H_s(X^* | l(X^*) < l_{\min}) \\ &\quad + (1 - A_{\mathbf{p}}^s) H_s(X^* | l(X^*) \geq l_{\min}) \\ &\leq 1 + A_{\mathbf{p}}^s \log a^{(l_{\min}-1)} + (1 - A_{\mathbf{p}}^s) \log M \\ &\leq 1 + (1 - A_{\mathbf{p}}^s(1 - \gamma)) \log M. \end{aligned}$$

Thus, we have $H_{\text{seg}}(P) \leq 1 + (1 - A_{\mathbf{p}}(1 - \gamma)) \log M$.

Now from [7, proof of Theorem 3] we have $E(L)H(P) = H_{\text{seg}}(P)$ so we have

$$\begin{aligned} R_{\mathbf{p}} &= \frac{\log M}{E(L)} \\ &\geq H(P) \frac{\log M}{1 + (1 - A_{\mathbf{p}}(1 - \gamma)) \log M} \\ &= \frac{H(P)}{1 - A_{\mathbf{p}}(1 - \gamma) + 1/\log M} \\ &\geq H(P)(1 + A_{\mathbf{p}}(1 - \gamma) - 1/\log M). \end{aligned}$$

Thus,

$$\begin{aligned} A_{\mathbf{p}}(1 - \gamma) &\geq \frac{k(a-1) \log \log M}{2 \log M} \Rightarrow R_{\mathbf{p}} \\ &\geq H(P) \left(1 + \frac{k(a-1) \log \log M}{2 \log M} - \frac{1}{\log M} \right). \end{aligned} \quad (7)$$

We will make a few more definitions. Let $N_j(x_1^{l_{\min}}, s)$ denote the number of times state $j \in S$ is visited when the source emits

$$\begin{aligned} \frac{\log M}{E(L)} &\leq \left(H(P) + \frac{a^3 \log e}{\log M p_{\min} (p_{\min} - 1/\sqrt{\log M})^2} \right) \\ &\quad \times \left(1 + \frac{(k(a-1)/2) \log \log M + \log(1/(p_{\min} - 1/\sqrt{\log M}))}{\log M - (k(a-1)/2) \log \log M - \log(1/(p_{\min} - 1/\sqrt{\log M}))} \right) \\ &\leq H(P) \left(1 + (1 + \delta) \frac{k(a-1) \log \log M}{2 \log M} \right). \end{aligned}$$

$x_1^{l_{\min}}$ and the initial state is s . Also, let $N_{j,i}(x_1^{l_{\min}}, s)$ be the number of times the output $i \in A$ occurs when the source is in state j . Let

$$\hat{P}_{ij}(x_1^{l_{\min}}, s) = \frac{N_{j,i}(x_1^{l_{\min}}, s)}{N_j(x_1^{l_{\min}}, s)}$$

if $N_j(x_1^{l_{\min}}, s) > 0$ else let $\hat{P}_{ij}(x_1^{l_{\min}}, s) = 1/a$. \hat{P} is a matrix which is an estimate of the letter probability matrix P . Again we will convert \hat{P} to a vector $\hat{\mathbf{p}}$ with $\hat{p}_{(j-1)(a-1)+i}(x^*, s) = \hat{P}_{ij}(x^*, s)$ for $j = 1, 2, \dots, k$ and $i = 1, 2, \dots, a-1$. Now let

$$\mathcal{X}_{\mathbf{p}}^s \triangleq \left\{ x^* \in \mathcal{D}_s : l(x^*) \geq l_{\min}, \rho(\hat{\mathbf{p}}(x_1^{l_{\min}}, s), \mathbf{p}) \leq c/\sqrt{l_{\min}} \right\}$$

where $\rho(\hat{\mathbf{p}}, \mathbf{p})$ denotes the L_2 distance between the two vectors and $c > 0$ is a constant to be specified later. Let $|\mathcal{X}_{\mathbf{p}}^s| \triangleq M_{\mathbf{p}}^s$. $P(\mathcal{X}_{\mathbf{p}}^s|s)$ is the probability that the event $\mathcal{X}_{\mathbf{p}}^s$ occurs given the starting state is s . Also let

$$T_{\mathbf{p}}^s \triangleq \sum_{x^* \in \mathcal{X}_{\mathbf{p}}^s} P(x^*|s) \log \frac{P(x^*|s)}{1/M}.$$

Since $E(L)H(P) = H_{\text{seg}}(P)$ we have

$$\begin{aligned} \log M &= E(L)H(P) + \sum_r \Pr(G_r) \sum_{s \in G_r} q(s|G_r) \\ &\quad \cdot \sum_{x^* \in \mathcal{D}_s} P(x^*|s) \log MP(x^*|s) \\ &\geq E(L)H(P) + \sum_r \Pr(G_r) \sum_{s \in G_r} q(s|G_r) \\ &\quad \cdot \left(T_{\mathbf{p}}^s + (1 - P(\mathcal{X}_{\mathbf{p}}^s|s)) \log \frac{(1 - P(\mathcal{X}_{\mathbf{p}}^s|s))}{(1 - M_{\mathbf{p}}^s)} \right) \\ &\geq E(L)H(P) + T_{\mathbf{p}} - \log e \end{aligned} \quad (8)$$

where the first inequality follows from the log-sum inequality, the second inequality follows from

$$T_{\mathbf{p}} \triangleq \sum_r \Pr(G_r) \sum_{s \in G_r} q(s|G_r) T_{\mathbf{p}}^s$$

and the fact that $\log x \geq (1 - 1/x) \log e$ for $x < 0$.

Equation (8) can also be written as

$$\frac{\log M}{E(L)} \geq H(P) + \frac{T_{\mathbf{p}}}{E(L)} - \frac{\log e}{E(L)}.$$

Since $E(L)H(P) \leq \log M$ we have

$$\begin{aligned} T_{\mathbf{p}} &\geq (1-\delta) \frac{k(a-1)}{2} \log \log M \Rightarrow R_{\mathbf{p}} \\ &\geq H(P) \left(1 + (1-\delta) \frac{k(a-1) \log \log M}{2 \log M} - \frac{\log e}{\log M} \right). \end{aligned} \quad (9)$$

Let

$$B_{\delta} \triangleq \left\{ \mathbf{p} : A_{\mathbf{p}}(1-\gamma) < \frac{k(a-1) \log \log M}{2 \log M}, \right. \\ \left. T_{\mathbf{p}} < (1-\delta) \frac{k(a-1)}{2} \log \log M \right\}.$$

Due to (7) and (9) we have that

$$\mathbf{p} \notin B_{\delta} \Rightarrow R_{\mathbf{p}} \geq H(P) \left(1 + (1-2\delta) \frac{k(a-1) \log \log M}{2 \log M} \right)$$

for sufficiently large M . Thus, if we can show that the volume of B_{δ} is vanishing as M increases we will be done. Let $B'_{\delta} = B_{\delta} \cap [\epsilon, 1]^{k(a-1)}$. Then we have $|B_{\delta}| \leq |B'_{\delta}| + k(a-1)\epsilon$, where $|\cdot|$ denotes the volume of a set.

Let $\mathcal{B}_{\mathbf{p}}$ be a ball of radius $c/\sqrt{l_{\min}}$ centered at \mathbf{p} . Let N be the maximal number of nonintersecting balls of radius $c/\sqrt{l_{\min}}$ that can be packed into B'_{δ} and let \mathcal{C} be the corresponding set of centers of these balls. Then the volume of B'_{δ} is bounded by

$$V \leq NV_0 \left(\frac{2c}{\sqrt{l_{\min}}} \right)^{k(a-1)}$$

where V_0 is a constant. Thus, to bound the volume of B'_{δ} we need to find a bound for N . To bound N we will need a lower bound on $T_{\mathbf{p}}$. We can lower-bound $T_{\mathbf{p}}$ as follows:

$$\begin{aligned} T_{\mathbf{p}} &= \sum_r \Pr(G_r) \sum_{s \in G_r} q(s|G_r) \sum_{x^* \in \mathcal{X}_{\mathbf{p}}^s} P(x^*|s) \log \frac{P(x^*|s)}{1/M} \\ &\geq \sum_r \Pr(G_r) \sum_{s \in G_r} q(s|G_r) P(\mathcal{X}_{\mathbf{p}}^s|s) \log \frac{P(\mathcal{X}_{\mathbf{p}}^s|s)}{M_{\mathbf{p}}^s/M} \\ &\geq P_{\text{avg}}(\mathcal{X}_{\mathbf{p}}^s) \log \frac{P_{\text{avg}}(\mathcal{X}_{\mathbf{p}}^s)}{M_{\mathbf{p}}/M} \end{aligned} \quad (10)$$

where both inequalities follow from the log-sum inequality and

$$P_{\text{avg}}(\mathcal{X}_{\mathbf{p}}^s) \triangleq \sum_r \Pr(G_r) \sum_{s \in G_r} q(s|G_r) P(\mathcal{X}_{\mathbf{p}}^s|s).$$

$M_{\mathbf{p}}$ is defined similarly from $M_{\mathbf{p}}^s$.

From (10), for $\mathbf{p} \in B_{\delta}$, we have

$$P_{\text{avg}}(\mathcal{X}_{\mathbf{p}}^s) \log \frac{P_{\text{avg}}(\mathcal{X}_{\mathbf{p}}^s)}{M_{\mathbf{p}}/M} \leq (1-\delta) \frac{k(a-1)}{2} \log \log M.$$

This implies that

$$-\log \frac{M_{\mathbf{p}}}{M} \leq -\log P_{\text{avg}}(\mathcal{X}_{\mathbf{p}}^s) + \frac{(1-\delta)k(a-1)}{2P_{\text{avg}}(\mathcal{X}_{\mathbf{p}}^s)} \log \log M \quad (11)$$

$$= \left(\frac{-2 \log P_{\text{avg}}(\mathcal{X}_{\mathbf{p}}^s)}{\log \log M} + \frac{(1-\delta)k(a-1)}{P_{\text{avg}}(\mathcal{X}_{\mathbf{p}}^s)} \right) \cdot \frac{\log \log M}{2}. \quad (12)$$

We will need the following lemma which essentially says that $P_{\text{avg}}(\mathcal{X}_{\mathbf{p}}^s)$ is close to one.

Lemma 3: For every $\delta > 0$ there exist c and M such that $P_{\text{avg}}(\mathcal{X}_{\mathbf{p}}^s) \geq 1 - \delta/2$ for all $\mathbf{p} \in B'_{\delta}$.

We give the proof of this lemma in the Appendix. Lemma 3 implies that for sufficiently large M we have for some $\alpha < 1$

$$M_{\mathbf{p}} \geq M(\log M)^{-k(a-1)\alpha/2}.$$

Now, since the set of balls of radius $c/\sqrt{l_{\min}}$ are nonintersecting, we have

$$\sum_{\mathbf{p} \in \mathcal{C}} M_{\mathbf{p}}^s \leq M.$$

By the definition of $M_{\mathbf{p}}$ we have $M_{\mathbf{p}} \leq \sum_{s \in \mathcal{S}} M_{\mathbf{p}}^s$ and hence $\sum_{\mathbf{p} \in \mathcal{C}} M_{\mathbf{p}} \leq kM$. Using the fact that

$$M_{\mathbf{p}} > M(\log M)^{-k(a-1)\alpha/2}$$

we have $N \leq k(\log M)^{k(a-1)\alpha/2}$. Thus,

$$V \leq \frac{4V_0 c^{k(a-1)}}{\gamma^{k(a-1)/2}} (\log M)^{-k(a-1)(1-\alpha)/2}$$

for some constant V_0 independent of M .

Thus, we have $|B_\delta| \leq \epsilon(1 + k(a-1))$ for sufficiently large M depending on ϵ and δ . \square

V. PERFORMANCE ON INDIVIDUAL SEQUENCES

So far we have considered coding of the output of a Markov source using a universal variable-to-fixed length code. Our measure of performance was $\max_s \log |D_s|/E(L)$, so we tried to find codes of a given size for which $E(L)$ is maximized. We now analyze the performance of the coding method described in Section III on an individual sequence $x^n \in A^n$. Here we do not assume x^n to be generated by a probabilistic source. Given a variable-to-fixed length code of size M the output length is $\log M$ per phrase. If a dictionary \mathcal{D} parses x^n into $r(x^n, \mathcal{D})$ phrases (we denote the i th phrase by $x^{(i)}$) the output length is $r(x^n, \mathcal{D}) \log M$. The coding rate for this sequence is, thus, $r(x^n, \mathcal{D}) \log M/n$. We will compare the code length achieved by the code we constructed in Section III to the best performance achievable on the sequence x^n by codes in a particular class. This measure of performance easily incorporates adaptive coding methods, as opposed to our earlier measure.

To construct the code in Section III, we needed to assume that we know the underlying structure of the source (i.e., the set of states, initial state, and the next-state function). The class of codes we consider corresponds to the set of Markov sources which have the given structure. This set is parameterized by P , the transition probability matrix. For each valid matrix P we have a corresponding code which assigns a code length $\lceil \log 1/P(x^n|s_0) \rceil$ to the string x^n . For a particular string x^n there will be a transition matrix P which minimizes the code length. The next theorem gives a bound on the performance of code \mathcal{D} (described in Section III) in terms of this minimum code length. The proof is similar to the proof of Theorem 1.

Theorem 3: Fix $\delta > 0$. For sufficiently large M we have

$$\begin{aligned} \frac{r(x^n, \mathcal{D}) \log M}{n} &\leq \frac{1}{n} \min_P \log \frac{1}{P(x^n|s_0)} + \frac{5a \log e}{2\sqrt{\log M}} \\ &\quad + \frac{a(k(a-1) + 1)(1 + \delta) \log \log M}{\log M} \end{aligned}$$

for all $x^n \in A^n$ with $n > M$.

Proof: Given a next state function and an initial state s_0 the probability of a string x^n can be written as

$$P(x^n|s_0) = \prod_{s \in \mathcal{S}} \prod_{a \in A} P(a|s)^{N_{s,a}(x^n, s_0)}$$

where, as before, $N_{s,a}(x^n, s_0)$ denotes the number of times a occurs in state s when x^n is the output of a source starting in state s_0 . It is clear that $P(x^n|s_0)$ is maximized by

$$P^*(a|s) = N_{s,a}(x^n, s_0)/N_s(x^n, s_0)$$

where

$$N_s(x^n, s_0) = \sum_{a \in A} N_{s,a}(x^n, s_0).$$

Let Q be the transition matrix on the grid (see Section III) which is component-wise closest to P^* . By the construction of the dictionary for any $x^* \in D_s$ we have

$$Q(x^*|s) \leq \frac{m^{k(a-1)}}{MQ_{\min}}$$

where Q_{\min} is the minimal entry in the matrix Q . We assume that $n \geq M$ so that there is at least one completed phrase. Now we have

$$Q(x^n|s_0) = \prod_{i=1}^{r(x^n, \mathcal{D})} Q(x^{(i)}|s_{i-1}) \leq \left(\frac{m^{k(a-1)}}{MQ_{\min}} \right)^{r(x^n, \mathcal{D})}.$$

Since $Q_{\min} \geq 1/(m+1)$ so we have

$$\begin{aligned} r(x^n, \mathcal{D}) \log \frac{M}{m^{k(a-1)}(m+1)} &\leq \log \frac{1}{Q(x^n|s_0)} \\ &= \log \frac{1}{P^*(x^n|s_0)} + \log \frac{P^*(x^n|s_0)}{Q(x^n|s_0)} \\ &= \log \frac{1}{P^*(x^n|s_0)} + \sum_{s \in \mathcal{S}} N_s(x^n, s_0) D(P^*(\cdot|s) \| Q(\cdot|s)) \\ &\leq \log \frac{1}{P^*(x^n|s_0)} + \sum_{s \in \mathcal{S}} N_s(x^n, s_0) \log \sum_{i=1}^a \left(\frac{P^*(i|s)^2}{Q(i|s)} \right) \\ &\leq \log \frac{1}{P^*(x^n|s_0)} + (\log e) \sum_{s \in \mathcal{S}} N_s(x^n, s_0) \\ &\quad \times \sum_{i=1}^a \frac{P^*(i|s)^2 - Q(i|s)^2}{Q(i|s)} \\ &\leq \log \frac{1}{P^*(x^n|s_0)} + (\log e) \sum_{s \in \mathcal{S}} N_s(x^n, s_0) \\ &\quad \times \sum_{i=1}^a \frac{P^*(i|s) + Q(i|s)}{mQ(i|s)} \\ &\leq \log \frac{1}{P^*(x^n|s_0)} + (n \log e) \frac{5a}{2m}. \end{aligned}$$

The last inequality follows due to the following argument: $Q(i|s)$ is the closest point to $P^*(i|s)$ on a grid of uniform spacing $1/(m+1)$ on the segment $(0, 1)$. Point i on the grid is

at $i/(m+1)$. Thus, the largest that $P * (i|s)/Q(i|s)$ can be is $3/2$. Hence, we have

$$\frac{r(x^n, \mathcal{D}) \log M}{n} \leq \left(\frac{1}{n} \log \frac{1}{P^*(x^n|s_0)} + (\log e) \frac{5a}{2m} \right) \cdot \left(\frac{\log M}{\log M - (k(a-1)+1) \log m+1} \right).$$

Taking $m = \sqrt{\log M}$ (as in Section III) and for sufficiently large $M \geq M(\delta)$, we have

$$\frac{r(x^n, \mathcal{D}) \log M}{n} \leq \frac{1}{n} \min_P \log \frac{1}{P(x^n|s_0)} + \frac{5a \log e}{2\sqrt{\log M}} + \frac{a(k(a-1)+1)(1+\delta) \log \log M}{\log M}. \quad \square$$

Again, although we have stated this result for the case when the state can be tracked by the encoder and decoder, we can use a modification (described at the end of Section III) to remove this restriction. The minimum on the RHS of our result would then be over P and T (the next state function), and the constants on the RHS would be larger.

VI. ANOTHER CODING METHOD

The method described in Section III is optimal but requires the dictionary for each state to be precomputed and stored unlike the coding method given by Nissenbaum and Feder [1]. In [1], the performance of the coding method is investigated for i.i.d. sources. We describe and investigate the performance of this coding method described for Markov sources. The method gives a variable-to-fixed length source-coding method which can be used in conjunction with a modeler for the source statistics which provides estimates of probabilities based on the past data. This scheme can be used to obtain a variable-to-fixed length coder for Markov sources and, in general, for stationary ergodic sources. For Markov sources we will find the rate at which coding rate converges to the entropy. The coding method is closely related to arithmetic coding.

We will first describe the method for variable-to-fixed length coding in detail. As before, we assume that our input is from a source x that takes values on an alphabet $A = \{1, 2, \dots, a\}$ of size a . Let $\hat{P}: A^* \rightarrow [0, 1]$ be a function which satisfies

$$\sum_{a \in A} \hat{P}(a) = 1 \quad \text{and} \quad \sum_{a \in A} \hat{P}(xa) = \hat{P}(x)$$

for any $x \in A^*$. We will use the probabilities given by \hat{P} to code the source. Suppose that we want to build a coding tree with M leaves. Initially, the range for our output index is $[1, M]$. Suppose that the source output is a string $x = x_1, x_2, x_3, \dots$. As each letter of the string is processed we will narrow the range for the index. When the range is strictly smaller than a we stop. Suppose that after stage m our index range is $[i_l(x_1^m), i_h(x_1^m)]$. Let the size of this set be $r(x_1^m)$. At each stage we reduce the range of indexes roughly proportional to the probability of the symbol that occurs. Suppose that $x_{m+1} = i$. We make the size of the new index set

$$r(x_1^{m+1}) = \lfloor (r(x_1^m) - a) * \hat{P}(i|x_1^m) \rfloor + 1$$

if $i < a$. If $i = a$ we make the index set size

$$r(x_1^{m+1}) = r(x_1^m) - \sum_{j=1}^{a-1} r(x_1^m j).$$

By the construction above we have that

$$r(x_1^{m+1}) \geq r(x_1^m) \hat{P}(x_{m+1}|x_1^m) - a. \quad (13)$$

Once we have found the sizes of the index range for various possible output letters we can then determine the lower and upper limits of the range when $x_{m+1} = i$ by

$$i_l(x_1^{m+1}) = i_l(x_1^m) + \sum_{j=1}^{i-1} r(x_1^m j)$$

and

$$i_h(x_1^{m+1}) = i_h(x_1^m) + r(x_1^m) - 1.$$

We proceed in this way until the size of the index range is smaller than the size of the input alphabet A . We arbitrarily choose the low end of the range to be the codeword for the input string and since we started with a range of $[1, M]$ we can code each output index with $\lceil \log M \rceil$ bits. If $\hat{P}(x_1^l)$ goes to zero as l increases for each infinite string x then the method described above always terminates and thus describes a tree with M leaves that can be used for variable-to-fixed length coding. Note that we need not build and store the tree, we can compute the parsing and the output index if $\hat{P}(0|x)$ can be computed for each string $x \in A^*$. The decoder must have access to the same \hat{P} and then the coding process of the encoder can be reversed at the decoding end. We have two choices once we code a phrase: we can either use past information to estimate probabilities of symbols or we can reset and start afresh. If we do not reset we are in effect adapting the code tree and so $E(L)$ is not a meaningful measure for the code. We will consider both these alternatives.

As an example of this coding method, consider the case when the source is binary and we code using distribution $\hat{P}(0) = 1 - \hat{P}(1) = 0.25$. Also, let $M = 4$. Initially our index range is $[1, 4]$. If we see a 0, the new size of index set is $\lceil (4-2)*0.25 \rceil = 1$ (the index set is $[1]$) and hence 0 is a codeword. If we see a 1, the index set is $[2, 4]$. Repeating this procedure, we see that the codewords are 0, 10, 110, 111.

A. Performance of the Coding Method

We will investigate in this section the performance of the coding method for Markov sources. Although the rate that we show the coding scheme achieves, is not optimal, it is close to being optimal. The coding method described needs a method to estimate probabilities. We will use the Krichevsky–Trofimov estimator [2]. This estimates the probability of seeing output i in state j as

$$\hat{P}(i|j, x^n, s_0) = \frac{N_{j,i}(x^n, s_0) + 1/a}{N_j(x^n, s_0) + 1}$$

after seeing a string x^n starting from state s_0 . As before, $N_{j,i}(x^n, s_0)$ is the number of times output i occurs when the source is in state j and $N_j(x^n, s_0) + 1$ is the number of times

that state j occurs. Thus, the estimate of the probability of a string x^n is

$$\hat{P}(x^n|s_0) = \prod_{i=1}^n \hat{P}(x_i|s_{i-1}, x^{i-1}, s_0).$$

We fix a given initial state s_0 and omit it from the notation. Thus, we will write $\hat{P}(x^n)$ for $\hat{P}(x^n|s_0)$.

To determine the performance of the coding method we will need to upper-bound the probability of the leaves of the coding tree. Consider a string x_1^l which is a leaf of the coding tree described above. Since x_1^l is a leaf we know that after l stages the size of the index range $r(x_1^l)$ is at most $a - 1$. Using this fact and (13) we have that $r(x_1^{l-1}) * \hat{P}(x_l|x_1^{l-1}) - a \leq a - 1$. Proceeding backward we have that

$$\begin{aligned} (\dots(((M\hat{P}(x_1) - a)\hat{P}(x_2|x_1) - a)\hat{P}(x_3|x_1^2) - a)\dots) \\ \cdot \hat{P}(x_l|x_1^{l-1}) - a \leq a - 1. \end{aligned}$$

Equivalently, we have

$$M\hat{P}(x_1^l) \leq 2a - 1 + a \sum_{j=1}^{l-1} \hat{P}(x_{j+1}^l|x_1^j). \quad (14)$$

The theorem below bounds the performance achieved by the coding method described above with the counts being reset after the coding of a phrase. Clearly, this does not make sense if the source is stationary, but it has advantages of not propagating errors and in bounding the peak rates. The code does not propagate errors because the output codeword is of a fixed length, and the code does not change with the bits coded so far. Note though that, in case of an error, we do lose synchronization.

Theorem 4: Consider a Markov source with nonzero entropy rate. For any $\delta > 0$ we have for sufficiently large M

$$R = \frac{\log M}{E(L)} \leq H + (1 + \delta) \frac{(k(a-1) + 2) \log \log M}{2 \log M} H$$

where k is the number of states and a is the size of the alphabet.

Proof: Using (39) and (40) from [7] we have

$$\begin{aligned} \sum_r P(G_r) \sum_{s \in G_r} q(s|G_r) \sum_{x^* \in D_s} P(x^*|s) \log \frac{1}{P(x^*|s)} \\ = E(L)H. \quad (15) \end{aligned}$$

As before, G_r are the irreducible sets that are induced by the k coding trees, $P(G_r)$ is the probability that the source will be in a state in G_r , and $q(s|G_r)$ is the probability the source is in state $s \in G_r$ given that the source state is in G_r . Since

$$\begin{aligned} \sum_{x^* \in D_s} P(x^*|s) \log \frac{1}{P(x^*|s)} = \sum_{x^* \in D_s} P(x^*|s) \log \frac{\hat{P}(x^*|s)}{P(x^*|s)} \\ + \sum_{x^* \in D_s} P(x^*|s) \log \frac{1}{\hat{P}(x^*|s)} \quad (16) \end{aligned}$$

we can lower-bound $E(L)$ by lower-bounding the two terms on the RHS of (16). Now, from [2, eq. (2.6)], we have

$$\log \frac{Q(x^*)}{\hat{Q}(x^*)} \leq C + \frac{a-1}{2} \log l(x^*)$$

where Q is a memoryless distribution, \hat{Q} is the Krichevsky–Trofimov estimate for the probability of x^* under the i.i.d. assumption, and C is a positive constant independent of x^* . In our case, since we have a Markov source, we can apply this result to k subsequences of x^* each subsequence consisting of symbols which share a common underlying state. Using this and the concavity of the logarithm twice we get

$$\begin{aligned} \sum_{x^* \in D_s} P(x^*|s) \log \frac{\hat{P}(x^*|s)}{P(x^*|s)} \\ \geq - \sum_{x^* \in D_s} P(x^*|s) \left(\frac{k(a-1)}{2} \log \left(\frac{l(x^*)}{k} \right) + kC \right) \\ \geq - \left(\frac{k(a-1)}{2} \log \left(\frac{E(L|s)}{k} \right) + kC \right). \end{aligned}$$

Consider now the second term. We have

$$\begin{aligned} \sum_{x^* \in D_s} P(x^*|s) \log \frac{1}{\hat{P}(x^*|s)} \\ \geq \sum_{x^* \in D_s} P(x^*|s) \log \frac{M}{2a - 1 + a \sum_{j=1}^{l(x^*)-1} \hat{P}(x_{j+1}^{l(x^*)}|x_1^j)} \\ \geq \log M - \sum_{x^* \in D_s} P(x^*|s) \log(al(x^*) + a - 1) \\ \geq \log M - \log(aE(L|s) + a - 1). \end{aligned}$$

The first inequality follows from (14), the second because $P(x_{j+1}^l|x_1^j) \leq 1$, and the third from the convexity of $-\log$.

Combining these two lower bounds with (15) we have

$$\begin{aligned} E(L)H \geq \sum_r P(G_r) \sum_{s \in G_r} q(s|G_r) \\ \cdot \left(-\frac{k(a-1)}{2} \log \left(\frac{E(L|s)}{k} \right) - Ck + \log M \right. \\ \left. - \log(aE(L|s) + a - 1) \right) \\ \geq - \left(\frac{k(a-1)}{2} \log \left(\frac{E(L)}{k} \right) + Ck \right) \\ + \log M - \log(aE(L) + a - 1) \end{aligned}$$

where the second inequality follows from the definition of $E(L)$ and the convexity of the $-\log$ function.

Using (15) and the fact that the entropy of a distribution on an alphabet of size M is no larger than $\log M$ we have that $HE(L) \leq \log M$. Thus, we have from the monotonicity of \log

$$\begin{aligned} E(L)H \geq - \left(\frac{k(a-1)}{2} \log \left(\frac{\log M}{Hk} \right) + Ck \right) \\ + \log M - \log(a \log M/H + a - 1). \end{aligned}$$

Equivalently, the compression ratio $R = \log M/E(L)$ is upper-bounded as shown in the equation at the bottom of the following page.

Thus, for any $\delta > 0$ we have for sufficiently large M depending on the source entropy and δ

$$R \leq \left(1 + (1 + \delta) \frac{(k(a-1) + 2) \log \log M}{2 \log M} \right) H. \quad \square$$

The redundancy we have shown is of the order of $H((k(a-1) + 2) \log \log M)/2 \log M$, thus, it exceeds the optimal redundancy by $H(\log \log M)/\log M$. We have seen the performance of the Nissenbaum–Feder coding method on a Markov source when the next-state function is known. If we do not know the next-state function, then we could use an average of the Krichevsky–Trofimov estimator over various next-state functions and since these are finite in number this will not affect the result of the previous theorem. Computationally we will be worse off since we will have to compute the average of a possibly large number (exponential in the number of states) of estimators. When the source is a tree source with memory at most D there are efficient ways to compute the average distribution without knowing the exact structure of the source [10]. These methods can be used with the Nissenbaum–Feder coding method to give efficient compression algorithms.

The Nissenbaum–Feder coding method can be used adaptively by not resetting the counts when a phrase end is reached. In Section VI-B, we look at the performance of adaptive and nonadaptive methods on individual sequences.

B. Performance on Individual Sequences

We consider the performance of the variable-to-fixed length coding method described above along with the Krichevsky–Trofimov estimator for estimating probabilities. We will bound our code length in terms of the best achievable code length from the set of codes which correspond to sources with a given set of states S , alphabet A , and next-state function T . Different sources in this class correspond to different parameter values. Let $r(x^n)$ be the number of phrases when x^n is coded and $P(x^n|s_0)$ denote the probability if the source and parameters given by the matrix P . Also, let $|S| = k$ and $|A| = a$. If we reset counts after the coding of each phrase then we have the bound given by the following theorem.

Theorem 5: For any $\delta > 0$ we have for sufficiently large M

$$\frac{r(x^n) \log M}{n} \leq \frac{1}{n} \min_P \left(\log \frac{1}{P(x^n|s_0)} \right) + (1 + \delta) C'' (k(a-1)/2 + 1) \frac{\log \log M}{\log M}$$

where

$$C'' = 1 + \left(\frac{k(a-1)}{2} + 1 \right) \frac{a \log e}{2a-3}.$$

Proof: From (14), we have for any phrase $x^{(i)}$, $M \hat{P}(x^{(i)}|s_{i-1}) \leq 2a-3 + al(x^{(i)})$. Thus, we have

$$\begin{aligned} & \frac{r(x^n) \log M}{n} \\ & \leq \frac{1}{n} \sum_{i=1}^{r(x^n)} \left(\log \frac{1}{\hat{P}(x^{(i)}|s_{i-1})} + \log \left(2a-3 + al(x^{(i)}) \right) \right) \end{aligned}$$

$$\begin{aligned} & \leq \frac{1}{n} \sum_{i=1}^{r(x^n)} \left(\log \frac{1}{P(x^{(i)}|s_{i-1})} + kC + k \frac{a-1}{2} \right. \\ & \quad \left. \cdot \log \left(\frac{l(x^{(i)})}{k} \right) + \log \left(2a-3 + al(x^{(i)}) \right) \right) \\ & \leq \frac{1}{n} \log \frac{1}{P(x^n|s_0)} + C \frac{rk}{n} + \frac{rk(a-1)}{2n} \log \left(\frac{n}{rk} \right) \\ & \quad + \frac{r}{n} \log \left(2a-3 + \frac{an}{r} \right) \\ & \leq \frac{1}{n} \log \frac{1}{P(x^n|s_0)} + C \frac{rk}{n} + \left(\frac{k(a-1)}{2} + 1 \right) \\ & \quad \cdot \frac{r}{n} \log \left(2a-3 + \frac{an}{r} \right). \end{aligned} \quad (17)$$

Now $\log(1+x) \leq x(\log e)$ and $\frac{1}{n} \min_P \log \frac{1}{P(x^n|s_0)} \leq 1$, thus we have

$$\begin{aligned} \frac{r(x^n) \log M}{n} & \leq 1 + C \frac{rk}{n} \\ & \quad + \left(\frac{k(a-1)}{2} + 1 \right) \frac{r}{n} \left(\log(2a-3) + \frac{an \log e}{r(2a-3)} \right) \end{aligned}$$

or, equivalently,

$$\frac{r(x^n)}{n} \leq \frac{C''}{\log M - C'}$$

where

$$C' = Ck + \left(\frac{k(a-1)}{2} + 1 \right) \log(2a-3).$$

Substituting this back into (17) and using the fact that $x \log(2a-3 + a/x)$ is increasing we get

$$\begin{aligned} \frac{r(x^n) \log M}{n} & \leq \frac{1}{n} \log \frac{1}{P(x^n|s_0)} + \frac{kCC''}{\log M - C'} \\ & \quad + \left(\frac{k(a-1)}{2} + 1 \right) \frac{C''}{\log M - C'} \\ & \quad \cdot \log \left(2a-3 + a \frac{(\log M - C')}{C''} \right). \end{aligned}$$

Since this equation holds for all valid P we have the required result. \square

In the above analysis, we assumed that the counts are reset when a phrase is coded. If we do not reset these counts we will perform better if the source is stationary. We now analyze the case when the counts are not reset, i.e., the whole past is used to estimate probabilities. This is just a minor modification to the previous analysis.

Theorem 6: For any $\delta > 0$ we have for sufficiently large M

$$\frac{r(x^n) \log M}{n} \leq \frac{1}{n} \min_P \left(\log \frac{1}{P(x^n|s_0)} \right) + (1 + \delta) \frac{\log \log M}{\log M}.$$

$$R \leq \frac{\log M}{\log M - \log \left(a \frac{\log M}{H} + a - 1 \right) - \left(\frac{k(a-1)}{2} \log \left(\frac{\log M}{Hk} \right) + Ck \right)} H.$$

Proof: From (14), we have for any phrase $x^{(i)}$, $M\hat{P}(x^{(i)}|s_{i-1}) \leq 2a - 3 + al(x^{(i)})$. Thus, we have

$$\begin{aligned} & \frac{r(x^n) \log M}{n} \\ & \leq \frac{1}{n} \sum_{i=1}^{r(x^n)} \left(\log \frac{1}{\hat{P}(x^{(i)}|s_{i-1})} + \log (2a - 3 + al(x^{(i)})) \right) \\ & \leq \frac{1}{n} \log \frac{1}{\hat{P}(x_1^n|s_0)} + \frac{r}{n} \log \left(2a - 3 + a \frac{n}{r} \right). \end{aligned}$$

We can now proceed as in the proof of the preceding theorem to obtain the required upper bound. \square

APPENDIX

In this appendix we prove Lemma 3.

Lemma: For every $\delta > 0$, there exist c and M such that $P_{\text{avg}}(\mathcal{X}_{\mathbf{p}}^s) \geq 1 - \delta/2$ for all $\mathbf{p} \in B'_\delta$.

Proof: To lower-bound $P_{\text{avg}}(\mathcal{X}_{\mathbf{p}}^s)$ we will lower-bound $P(\mathcal{X}_{\mathbf{p}}^s|s)$ for each $s \in S$. By definition

$$\begin{aligned} & P(\mathcal{X}_{\mathbf{p}}^s|s) \\ & = P\left(\{x^* \in \mathcal{D}_s: l(x^*) \geq l_{\min}, \rho(\hat{\mathbf{p}}(x_1^{l_{\min}}, s), \mathbf{p}) \leq c/\sqrt{l_{\min}}\} | s\right) \\ & \geq 1 - P(\{x^* \in \mathcal{D}_s: l(x^*) < l_{\min}\} | s) \\ & \quad - \Pr_{\mathbf{p}}\left(\{x_1^{l_{\min}}: \rho(\hat{\mathbf{p}}(x_1^{l_{\min}}), \mathbf{p}) > c/l_{\min}\} | s\right) \\ & \geq 1 - A_{\mathbf{p}}^s - \frac{1}{q(s)} P\left(\rho(\hat{\mathbf{p}}(X_1^{l_{\min}}), \mathbf{p}) > c/\sqrt{l_{\min}}\right). \end{aligned} \quad (18)$$

We will first upper-bound

$$P(\rho(\hat{\mathbf{p}}(X_1^{l_{\min}}), \mathbf{p}) > c/\sqrt{l_{\min}}).$$

We drop the explicit dependence on the output sequence and the initial state and let

$$N_j = N_j(X_1^{l_{\min}}, S_1) \quad \text{and} \quad N_{j,x} = N_{j,x}(X_1^{l_{\min}}, S_1).$$

Then we have

$$\begin{aligned} & P\left(\rho(\hat{\mathbf{p}}(X_1^{l_{\min}}), \mathbf{p}) > c/\sqrt{l_{\min}}\right) \\ & \leq \sum_{j=1}^k \sum_{x=1}^{a-1} P\left(\left|\frac{N_{j,x}}{N_j} - P(x|j)\right| > \frac{c}{\sqrt{k(a-1)l_{\min}}}\right). \end{aligned} \quad (19)$$

Consider one of the terms in the above summation.

$$\begin{aligned} & P\left(\left|\frac{N_{j,x}}{N_j} - P(x|j)\right| > \frac{c}{\sqrt{k(a-1)l_{\min}}}\right) \\ & \leq P\left(\left|\frac{N_j}{l_{\min}} - q(j)\right| \geq \frac{c}{8\sqrt{l_{\min}}}\right) \\ & \quad + P\left(\left|\frac{N_{j,x}}{N_j} - P(x|j)\right| > \frac{c}{\sqrt{k(a-1)l_{\min}}}, \right. \\ & \quad \left. \cdot \left|\frac{N_j}{l_{\min}} - q(j)\right| < \frac{c}{\sqrt{l_{\min}}}\right). \end{aligned} \quad (20)$$

We can bound the first term in (20) using the Chebyshev inequality for $\mathbf{p} \in [\epsilon, 1]^{k, (a-1)}$

$$\begin{aligned} P\left(\left|\frac{N_j}{l_{\min}} - q(j)\right| \geq \frac{c}{8\sqrt{l_{\min}}}\right) & \leq \left(\frac{2l_{\min} + \epsilon}{\epsilon^3}\right) \frac{64}{c^2 l_{\min}} \\ & \leq \frac{\delta\epsilon}{16k(a-1)} \end{aligned}$$

for

$$c = \sqrt{3072k(a-1)/(\epsilon^4\delta)}.$$

Let Y_i^j be the random variable which gives the output value when the source enters state j for the i th time. The sequence of random variables $\{Y_i^j\}_{i=1}^\infty$ is an i.i.d. sequence for each state j . Then the second term in (20) can be bounded as follows:

$$\begin{aligned} & P\left(\left|\frac{N_{j,x}}{N_j} - P(x|j)\right| \geq \frac{c}{\sqrt{k(a-1)l_{\min}}}, \left|\frac{N_j}{l_{\min}} - q(j)\right| < \frac{c}{\sqrt{l_{\min}}}\right) \\ & \leq P\left(\frac{\sum_{i=1}^{l_{\min}q(j)+\sqrt{l_{\min}}c/8} 1(Y_i^j=x)}{l_{\min}q(j)-\sqrt{l_{\min}}c/8} - P(x|j) \geq \frac{c}{\sqrt{l_{\min}}}\right) \\ & \quad + P\left(\frac{\sum_{i=1}^{l_{\min}q(j)-\sqrt{l_{\min}}c/8} 1(Y_i^j=x)}{l_{\min}q(j)+\sqrt{l_{\min}}c/8} - P(x|j) \leq -\frac{c}{\sqrt{l_{\min}}}\right) \\ & \leq P\left(\frac{\sum_{i=1}^{l_{\min}q(j)+\sqrt{l_{\min}}c/8} 1(Y_i^j=x)}{l_{\min}q(j)+\sqrt{l_{\min}}c/8} - P(x|j) \geq \frac{c}{2\sqrt{l_{\min}}}\right) \\ & \quad + P\left(\frac{\sum_{i=1}^{l_{\min}q(j)-\sqrt{l_{\min}}c/8} 1(Y_i^j=x)}{l_{\min}q(j)-\sqrt{l_{\min}}c/8} - P(x|j) \leq -\frac{c}{2\sqrt{l_{\min}}}\right) \\ & \leq \frac{3}{\epsilon c^2}. \end{aligned}$$

where the second inequality follows for l_{\min} sufficiently large and the last inequality follows from the Chebyshev inequality for l_{\min} sufficiently large. Thus, we have

$$\begin{aligned} & P\left(\left|\frac{N_{j,x}}{N_j} - P(x|j)\right| > \frac{c}{\sqrt{k(a-1)l_{\min}}}, \left|\frac{N_j}{l_{\min}} - q(j)\right| < \frac{c}{\sqrt{l_{\min}}}\right) \\ & \leq \frac{\delta\epsilon}{16k(a-1)} \end{aligned}$$

for

$$c = \sqrt{48k(a-1)/\epsilon^2\delta}.$$

Thus we have for

$$c = \max\left(\sqrt{3072k(a-1)/(\epsilon^4\delta)}, \sqrt{48k(a-1)/\epsilon^2\delta}\right)$$

and sufficiently large l_{\min} that

$$P(\rho(\hat{\mathbf{p}}(X_1^{l_{\min}}), \mathbf{p}) > c/\sqrt{l_{\min}}) \leq \delta\epsilon/8.$$

Since $q(s) \geq \epsilon/2$ for $\mathbf{p} \in [\epsilon, 1]^{k(a-1)}$ and

$$A_{\mathbf{p}} < \frac{k(a-1) \log \log M}{2(1-\gamma) \log M}$$

we have for sufficiently large M , $P_{\text{avg}}(\mathcal{X}_{\mathbf{p}}^s) \geq 1 - \delta/2$. \square

REFERENCES

- [1] B. Nissenbaum and M. Feder, "An adaptive variable-to-fixed lossless coding scheme," in *Proc. IEEE Int. Symp. Information Theory*, Trondheim, Norway, 1994, p. 193.
- [2] R. E. Krichevsky and V. K. Trofimov, "The performance of Universal encoding," *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 199–207, Mar. 1981.
- [3] N. Merhav and D. L. Neuhoff, "Variable-to-fixed length codes provide better large deviations performance than fixed-to-variable length codes," *IEEE Trans. Inform. Theory*, vol. 38, pp. 135–140, Jan. 1992.
- [4] J. Rissanen, "Universal coding, information, prediction, and estimation," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 629–636, July 1984.
- [5] S. A. Savari, "Variable-to-fixed length codes and plurally parsable dictionaries," in *Proc. 1999 IEEE Data Compression Conf.*, 1999, pp. 453–462.
- [6] S. A. Savari and R. G. Gallager, "Generalized Tunstall codes for sources with memory," *IEEE Trans. Inform. Theory*, vol. 43, pp. 658–668, Mar. 1997.
- [7] T. J. Tjalkens and F. M. J. Willems, "Variable to fixed-length codes for Markov sources," *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 246–257, Mar. 1987.
- [8] —, "Universal variable-to-fixed-length source codes based on Lawrence's algorithm," *IEEE Trans. Inform. Theory*, vol. 38, pp. 247–253, Mar. 1992.
- [9] B. P. Tunstall, "Synthesis of noiseless compression codes," Ph.D. dissertation, Georgia Inst. Technol., Atlanta, GA, Sept. 1967.
- [10] F. M. J. Willems, Y. M. Shtarkov, and T. J. Tjalkens, "The context-tree weighting method: Basic properties," *IEEE Trans. Inform. Theory*, vol. 41, pp. 653–664, May 1995.
- [11] J. Ziv, "Variable-to-fixed length codes are better than fixed-to-variable length codes for Markov sources," *IEEE Trans. Inform. Theory*, vol. 36, pp. 861–863, July 1990.