The limiting value of the lower bound for $\bar{n}$ in (19) as $c \to \infty$ is

$$H(R) - h_1(\varepsilon) = \tfrac{1}{2} \log_D \sigma^2/\varepsilon.$$

As is well known, this coincides with $R(\varepsilon)$ for $0 < \varepsilon \leq \sigma^2$.

As in the previous example, it is possible to obtain the bound of Lemma 4 by an alternative method. In this case, the set $G(t,x)$ is just a sphere in R^N with radius $(Nt)^{1/2}$. Hence $\mu[G(t,x)]$ is the volume of this sphere,

$$\mu[G(t,x)] = \frac{(\pi N t)^{N/2}}{(N/2)!}.$$

An application of Stirling's formula to this expression gives an expression that agrees with the bound in (16) (up to factors that do not increase exponentially with N).

VII. Concluding Remarks

For certain classes of distortion measures we have succeeded in establishing generalizations of the Kraft inequality. As is the case with the Kraft inequality for unique decipherability, the inequality is independent of any particular probability distribution on the source and depends only on the fidelity of reproduction according to our definition of ε-decodability. This separation of fidelity from source probability may sometimes be useful.

It does not seem possible to get a simple extension of the Kraft inequality for all distortion measures. Some assumption, like row balance in the discrete case or a difference measure in the continuous case, seems necessary in order to permit treatment of $d(\alpha,\beta)$ as a function of one variable instead of two variables. Jelinek [8] has shown that the Lagrange multiplier method [4], [5] of evaluating the rate-distortion function for balanced distortion measures can be extended to nonbalanced measures. In a sense, the Chernoff bounding method of Lemmas 1 and 2 takes the place of the Kuhn–Tucker conditions in the Lagrange multiplier method. However, the assumption of row balance plays an important role in the proof of Lemma 1. It is not clear whether this assumption can be relaxed in the same way that the corresponding assumption of column balance was relaxed by Jelinek [8].

It is conceivable that tighter versions of the Kraft inequality exist in those cases where our lower bound on mean length does not coincide with the rate-distortion function. In any case, the versions presented here have the virtue of simplicity.

References

[1] R. G. Gallager, *Information Theory and Reliable Communication.* New York: Wiley, 1968.
[2] H. Chernoff, "A measure of asymptotic efficiency for tests of a hypothesis based on a sum of observations," *Ann. Math. Statist.*, vol. 23, pp. 493–507, 1952.
[3] J. Karush, "A simple proof of an inequality of McMillan," *IRE Trans. Inform. Theory* (Corresp.), vol. IT-7, p. 118, Apr. 1961.
[4] J. T. Pinkston, "An application of rate-distortion theory to a converse to the coding theorem," *IEEE Trans. Inform. Theory*, vol. IT-15, pp. 66–71, Jan. 1969.
[5] C. E. Shannon, "Coding theorems for a discrete source with a fidelity criterion," in *Information and Decision Processes*, R. E. Machol, Ed. New York: McGraw-Hill, 1960, pp. 93–126.
[6] R. B. Ash, *Information Theory.* New York: Interscience, 1965.
[7] T. Berger, *Rate Distortion Theory.* Englewood Cliffs, N.J.: Prentice-Hall, 1971, p. 93.
[8] F. Jelinek, "Evaluation of distortion rate functions for low distortions," *Proc. IEEE* (Lett.), vol. 55, pp. 2067–2068, Nov. 1967.

Enumerative Source Encoding

THOMAS M. COVER

***Abstract*—Let S be a given subset of binary n-sequences. We provide an explicit scheme for calculating the index of any sequence in S according to its position in the lexicographic ordering of S. A simple inverse algorithm is also given. Particularly nice formulas arise when S is the set of all n-sequences of weight k and also when S is the set of all sequences having a given empirical Markov property. Schalkwijk and Lynch have investigated the former case. The envisioned use of this indexing scheme is to transmit or store the index rather than the sequence, thus resulting in a data compression of $(\log|S|)/n$.**

Manuscript received October 21, 1971; revised June 1, 1972. This work was supported in part by AFOSR Contract F44620-69-C-0101 and by NSF Contract GK-34363.
The author was on leave at the Massachusetts Institute of Technology, Cambridge, Mass., and Harvard University, Cambridge, Mass. He is now with the Department of Electrical Engineering and Statistics, Stanford University, Stanford, Calif.

I. Introduction

Let $\{0,1\}^n$ denote the set of binary n-sequences and let $x = (x_1,x_2,\cdots,x_n)$ denote a generic element of this set. Let S be a subset of $\{0,1\}^n$. Define n_S to be the number of elements in S and let $n_S(x_1,x_2,\cdots,x_k)$ denote the number of elements in S for which the first k coordinates are given by $(x_1,x_2,\cdots,x_k)$.

By lexicographic ordering we mean the ordinary dictionary ordering under the interpretation that $0 < 1$. More formally, $x < y$ if $x_k < y_k$ for the least index k such that $x_k \neq y_k$. For example, $00101 < 00110$.

The following formula provides the desired $1 - 1$ lexicographic mapping $S \to \{0,1,2,\cdots, |S| - 1\}$.

Proposition 1: The lexicographic index of $x \in S$ is given by

$$i_S(\boldsymbol{x}) = \sum_{j=1}^{n} x_j n_S(x_1,x_2,\cdots,x_{j-1},0). \quad (1)$$

Proof: Words with prefix $(x_1,x_2,\cdots,x_{j-1},0)$ lexicographically precede words with prefix $(x_1,x_2,\cdots,x_{j-1},1)$. For each j such that $x_j = 1$, we simply count the number of elements of S, given by $n_S(x_1,x_2,\cdots,x_{j-1},0)$, which first differ from $\boldsymbol{x}$ in the jth term and therefore have lower lexicographic index. By adding these numbers for $j = 1,2,\cdots,n$, we eventually count all the elements in S of lower index than $\boldsymbol{x}$.

The extension of these results to arbitrary finite alphabet sizes is immediate. The formula is as follows.

Proposition 2: The lexicographic index of $\boldsymbol{x} \in S \subseteq \{1,2,3,\cdots,M\}^n$ is given by

$$i_S(\boldsymbol{x}) = \sum_{j=1}^{n} \sum_{m=1}^{x_j - 1} n_S(x_1,x_2,\cdots,x_{j-1},m).$$

The inverse function is also easily calculated.

Here is the inverse for Proposition 1.

Inverse Algorithm: Let i and S be given. The following algorithm finds $\boldsymbol{x}$ such that $i_S(\boldsymbol{x}) = i$.

Step 1: If $i > n_S(0)$ set $x_1 = 1$ and set $i = i - n_S(0)$; otherwise set $x_1 = 0$.

Step 2: For $k = 2,\cdots,n$, if $i > n_S(x_1,x_2,\cdots,x_{k-1},0)$ set $x_k = 1$ and set $i = i - n_S(x_1,x_2,\cdots,x_{k-1},0)$; otherwise set $x_k = 0$.

II. Some Applications

Example 1—Enumeration of All Binary n-Sequences:

$$S = \{0,1\}^n.$$

Then $n_S(x_1,x_2,\cdots,x_k) = 2^{n-k}$, and

$$i_S(\boldsymbol{x}) = \sum_{i=1}^{n} x_i 2^{n-i}. \quad (2)$$

The inverse function of i is the standard base-two expansion $x_1 x_2 \cdots x_n$ of the integer i. There is nothing new here.

Example 2—Enumeration of Sequences of Weight w:

$$S = \left\{\boldsymbol{x} \in \{0,1\}^n : \sum_{j=1}^{n} x_j = w\right\}.$$

Here

$$n_S(x_1,x_2,\cdots,x_{k-1},0) = \binom{n-k}{n(w,k)}$$

where

$$n(w,k) = w - \sum_{j=1}^{k-1} x_j$$

since there are only this number of ways of placing the last $n(w,k)$ 1's in the last $n-k$ terms of the sequence. Therefore,

$$i_S(\boldsymbol{x}) = \sum_{k=1}^{n} x_k \binom{n-k}{n(w,k)}. \quad (4)$$

For example, for $w = 3$, $n = 7$,

$$i(1000101) = \binom{6}{3} + \binom{2}{2} + \binom{0}{1}$$
$$= 20 + 1 + 0 = 21$$
$$i(1110000) = \binom{6}{3} + \binom{5}{2} + \binom{4}{1}$$
$$= 20 + 10 + 4 = 34,$$

which is in agreement with

$$n_S - 1 = \binom{7}{3} - 1 = 34.$$

This function is quite simple to compute, as is its inverse. The resulting data compression is approximately $H(w/n)$, where H is the entropy function.

Suppose that even this encoding scheme is considered too complex by virtue of the complexity of the calculation of $\binom{n}{i}$, or in the general case, the calculation of $n_S(x_1,x_2,\cdots,x_k,0)$. The obvious answer is to approximate $\binom{n}{i}$ or $n_S(x_1,x_2,\cdots,x_k,0)$ by suitable functions of low complexity. Any integer upper bound will do, but sufficiently loose bounds will diminish the compression, although the mapping will still be invertible and no information will be lost.

The indexing scheme for sequences of length n and weight w is well known in the combinatorial literature (see, e.g., Lehmer [1]). The author does not know whether the general scheme of Proposition 2 has been previously published, although the idea seems to be implicit in any lexicographic indexing scheme. The derivation of the mapping of Example 2 for data compression has been given by Lynch [2] and Schalkwijk [3] (see also Davisson [4]). Schalkwijk's interesting paper provided the impetus for the generalized applications we are presenting here. The following application is not found elsewhere.

Example 3:

$$S = \left\{\boldsymbol{x} \in \{0,1\}^n : \left|\sum_{i=1}^{n} x_i - w\right| \le m\right\}.$$

Here we wish to enumerate (in order) all the sequences for which the number of 1's lies in the interval $[w - m, w + m]$. The application to the compression of a Bernoulli source with parameter p is obvious: simply setting $w = np$ and $m = (w(n-w)/n)^{1/2}\alpha$ guarantees $\Pr\{S\} \approx 2(1 - \Phi(\alpha))$. Thus a compression $H(p)$ can be achieved with probability approximately one by an appropriate moderately large choice of α.

In this example

$$n_S(x_1,x_2,\cdots,x_{k-1},0) = \sum_{t=w-m}^{w+m} \binom{n-k}{n(t,k)}. \quad (5)$$

Thus the index of $\boldsymbol{x}$ is given by

$$i_S(\boldsymbol{x}) = \sum_{k=0}^{n} \sum_{t=w-m}^{w+m} x_k \binom{n-k}{n(t,k)}. \quad (6)$$

Thus, for $w = 3$, $m = 1$, $i_S(11000110) = \binom{7}{2} + \binom{7}{3} + \binom{7}{4} + \binom{6}{1} + \binom{6}{2} + \binom{6}{3} + \binom{2}{1} + \binom{1}{0} = 115$.

Example 4—Enumeration of Permutations: Let S be the set of all permutations of the integers $\{1,2,\cdots,n\}$. Suppose that we wish to enumerate the $n!$ elements of S lexicographically. Thus, for example, $i(1,2,\cdots,n) = 0$, and $i(n, n-1,\cdots, 1) = n! - 1$. Given the permutation $(x_1, x_2,\cdots,x_n)$, let r_i equal the number of elements x_j, $j = i+1, i+2,\cdots, n$, such that $x_j < x_i$. Then by Proposition 2,

$$i(x_1,x_2,\cdots,x_n) = \sum_{i=1}^{n} r_i(n-i)!.$$

This is precisely Lehmer's [1] enumeration formula for permutations.

Example 5—Enumeration of Monotone Functions: This problem was suggested by the recent work of Elias [5] on simple encodings for monotone functions. Elias' encodings are only slightly less efficient in the length of the encoding than the optimal encoding given here and are recommended for certain practical applications.

Let F: $\{0,1,2,\cdots,m\} \to \{0,1,2,\cdots,n\}$ be an integer-valued function. We shall say that F is *monotone nondecreasing* if $F(i) \le F(i+1)$, $i = 0,1,\cdots, m-1$; and we shall say that F is *strictly monotone* if $F(i) < F(i+1)$, $i = 0,1,\cdots, m-1$. Finally, F will be said to be a *distribution function* if $F(m) = n$. We consider the following four separate enumeration problems:

$$\mathscr{F}_1 = \{F\text{: } F \text{ monotone nondecreasing}\}$$

$$\mathscr{F}_2 = \{F\text{: } F \text{ monotone nondecreasing distribution function}\}$$

$$\mathscr{F}_3 = \{F\text{: } F \text{ strictly monotone}\}$$

$$\mathscr{F}_4 = \{F\text{: } F \text{ strictly monotone distribution function}\}.$$

Clearly $\mathscr{F}_4 \subseteq \mathscr{F}_2 \subseteq \mathscr{F}_1$ and $\mathscr{F}_4 \subseteq \mathscr{F}_3 \subseteq \mathscr{F}_1$.

Now let $C_k(m,n)$ denote the number of elements in $\mathscr{F}_k$, where the functions F have domain $\{0,1,\cdots,m\}$ and range $\{0,1,\cdots,n\}$. To determine $C_k(m,n)$, define the jumps $n_i = F(i) - F(i-1)$, $i = 1,2,\cdots,m$, and define $n_{m+1} = n - F(m)$. Note that $n_{m+1} = 0$ if F is a distribution function and also that $n_i \ge 0$ and $n_i \ge 1$ imply weak and strict monotonicity, respectively. It can be seen that $C_k(m,n)$ is the number of integer solutions of

$$(k = 1)\text{: } \sum_{j=1}^{m+1} n_j = n, \qquad n_j \ge 0$$

$$(k = 2)\text{: } \sum_{j=1}^{m} n_j = n, \qquad n_j \ge 0$$

$$(k = 3)\text{: } \sum_{j=1}^{m+1} n_j = n + 1, \qquad n_j \ge 1$$

$$(k = 4)\text{: } \sum_{j=1}^{m} n_j = n, \qquad n_j \ge 1.$$

The number of solutions of these equations is simply the number of ways of placing a given number of balls in a given number of cells and can be found by an argument in Feller [6, pp. 38, 39] to be

$$C_1(m,n) = \binom{n+m+1}{m+1}$$

$$C_2(m,n) = \binom{n+m}{m}$$

$$C_3(m,n) = \binom{n+1}{m+1}$$

$$C_4(m,n) = \binom{n}{m}.$$

To define a lexicographic ordering on $\mathscr{F}$ we shall say $F_1 < F_2$ if $\exists k \in \{0,1,2,\cdots, m-1\}$ such that $F_1(i) = F_2(i)$, $i \le k$, $F_1(k+1) < F_2(k+1)$. The key observation is that the number of monotone functions for which $F(0),F(1),\cdots,F(k-1)$ are fixed and $F(k) > r$ is simply $C_i(m-k, n-r-1)$; i.e., the number of monotone functions with domain $\{0,1,\cdots, m-k\}$ and range $\{0,1,\cdots, n-r\}$. We apply Proposition 2 to count the number of functions having a *higher* index than F and then subtract this from the total number of functions in $\mathscr{F}_k$ to obtain the following expression for the index of F in $\mathscr{F}_k$:

$$i_k(F) = C_k(m,n) - 1 - \sum_{i=0}^{m} C_k(m-i, n-F(i)-1), \qquad k = 1,2,3,4.$$

Thus, for example, there are $\binom{4}{3} = 4$ strictly monotone functions and $\binom{6}{3} = 20$ weakly monotone functions from $\{0,1,2\}$ to $\{0,1,2,3\}$; and $i_3(0,1,2) = i_1(0,0,0) = 0$.

Example 6—Enumeration of Convex Functions F: $\{0,1,\cdots,m\} \to \{0,1,\cdots,n\}$: The local nature of the definition of convex functions yields a nice application of Proposition 2. The results, found with Bolour, will not be given here.

Many other examples can be given, but we shall proceed to one last example in which there is some empirical dependence among the terms of the sequence.

III. Application to Markov Processes

The following application is given because of its utility in encoding Markov processes and not because of its ease of derivation. We envisage the use as follows. Let $\boldsymbol{v}$ denote the sequence statistics given in (7). First an encoding of $\boldsymbol{v} = (v_{01},v_{10},v_{00},v_{11})$ (requiring at most $4 \log n$ bits) will be given, followed by an encoding of $\boldsymbol{x} \in S_{\boldsymbol{v}}$. This will result in a compression of $\boldsymbol{x}$ that is arbitrarily close to the optimal compression of a first-order stationary Markov source having statistics corresponding to $\boldsymbol{v}$. The difference comes from the asymptotically negligible $(4 \log n)/n$ bits per symbol necessary to encode $\boldsymbol{v}$. If $\boldsymbol{v}$ is known *a priori*, there is no loss whatsoever. The encoding of the *statistics* $\boldsymbol{v}$ of the sequence is similar to the idea of universal codes in Babkin [7] and Shtarkov and Babkin [8].

Define the number of 01, 10, 00, and 11 blocks in x by

$$\begin{aligned} v_{01} &= \sum_{i=1}^{n-1} \bar{x}_i x_{i+1} \\ v_{10} &= \sum_{i=1}^{n-1} x_i \bar{x}_{i+1} \\ v_{00} &= \sum_{i=1}^{n-1} \bar{x}_i \bar{x}_{i+1} \\ v_{11} &= \sum_{i=1}^{n-1} x_i x_{i+1}. \end{aligned} \tag{7}$$

Let

$$\boldsymbol{v} = (v_{01}, v_{10}, v_{00}, v_{11}). \tag{8}$$

Let $S = S_v$ denote the set of all $x \in \{0,1\}^n$ satisfying (7).

Remark: By definition we see that $v_{01} + v_{10} + v_{00} + v_{11} = n - 1$. Note also that a 2-state stationary Markov process, with transition probabilities P_{ij}, has

$$\begin{aligned} E\{v_{01}\} &= (n-1) p_{01} p_{10} / (p_{01} + p_{10}) \\ E\{v_{10}\} &= (n-1) p_{10} p_{01} / (p_{01} + p_{10}) \\ E\{v_{00}\} &= (n-1) p_{00} p_{10} / (p_{01} + p_{10}) \\ E\{v_{11}\} &= (n-1) p_{11} p_{01} / (p_{01} + p_{10}). \end{aligned} \tag{9}$$

From this it can be seen that the statistics p_{ij} of the process are completely recoverable from $E\{v\}$. Finally, for ergodic processes ($p_{00} \neq 1, p_{11} \neq 1$), $v \to E\{v\}$ with probability one. Thus almost all sample functions $x = (x_1, x_2, \cdots, x_n)$ will have $v(x)$ near $E\{v\}$. This is made explicit in the Shannon–MacMillan theorem (Ash [9, p. 197]).

We shall introduce the definition

$$g(v) = \text{number of sequences } x \text{ satisfying (7) for which } x_1 = 0. \tag{10}$$

Thus $g(v) = n_{S_v}(0)$ in the previous notation.

The critical calculation of g is given by the following lemma.

Lemma 1:

$$g(v_{01}, v_{10}, v_{00}, v_{11}) = \begin{cases} \dbinom{v_{00} + v_{10}}{v_{10}} \dbinom{v_{11} + v_{01} - 1}{v_{01} - 1}, & v_{01} = v_{10} \text{ or } v_{01} = v_{10} + 1 \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

Proof: $v(x) = v$, $x_1 = 0$, implies there are $v_{10} + 1$ runs of 0's and v_{01} runs of 1's; and also that there are a total of $v_{10} + v_{00} + 1$ 0's and $v_{01} + v_{11}$ 1's. Since the runs of 0's and 1's alternate, beginning with a run of 0's, we have $g = 0$ unless $v_{01} = v_{10}$ or $v_{01} = v_{10} + 1$.

Now from Feller [6, pp. 36–37] we know that there are $\binom{r-1}{m-1}$ positive integer solutions to $r_1 + r_2 + \cdots + r_m = r$. Thus there are $\binom{v_{10}+v_{00}}{v_{10}}$ ways to choose the (nonzero) run lengths of the 0's and $\binom{v_{10}+v_{11}-1}{v_{01}-1}$ ways to choose the run lengths of the 1's. The choices can be made independently, and they completely characterize the sequence x. Then (11) results and the lemma is established.

Finally, applying the fundamental counting result of Proposition 1 in Section I, we have the following calculation of the index of x in the set $S = S_v$ of all sequences with Markov property v.

Proposition 3:

$$i_S(x) = \sum_{k=1}^{n} x_k g(v - v(x_1, x_2, \cdots, x_{k-1}, \bar{x}_k)),$$

where $\bar{x}_k = 1 - x_k$.

Proof: We observe from (7) that $v = v(x_1, \cdots, x_n) = v(x_1, \cdots, x_k) + v(x_k, \cdots, x_n)$. Thus a sequence $(\bar{x}_k, z_{k+1}, \cdots, z_n)$ will be an acceptable termination of $(x_1, x_2, \cdots, \bar{x}_k)$ (i.e., $v(x_1, \cdots, \bar{x}_k, z_{k+1}, \cdots, z_n) = v$ and hence $(x_1, \cdots, \bar{x}_k, z_{k+1}, \cdots, z_n) \in S_v$) if and only if $v(\bar{x}_k, z_{k+1}, \cdots, z_n) = v - v(x_1, x_2, \cdots, \bar{x}_k)$. Since g counts the number of continuations for a given v for which the first term is zero, and since $x_k g = 0$ unless $\bar{x}_k = 0$, the proposition follows.

Proposition 4: The number of elements in S_v is given by $g(v_{01}, v_{10}, v_{00}, v_{11}) + g(v_{10}, v_{01}, v_{11}, v_{00})$.

Proof: The number of sequences with initial term 1 and property v is simply the number of sequences with initial term 0 and property $\tilde{v} = (v_{10}, v_{01}, v_{11}, v_{00})$; as is easily seen by complementing all the 0's and 1's. Thus $g(v)$ counts the elements in S_v with initial term 0, and $g(\tilde{v})$ counts the elements in S_v with initial term 1.

Comment: The inverse function $i \to x$ is easily calculated by the recurrence algorithm in Section I.

Example: Consider the sequence $x = (0,1,1,0,0,1,1,0)$. Here $v(x) = (v_{01}, v_{10}, v_{00}, v_{11}) = (2,2,1,2)$. The equivalence class S_v has $g(2,2,1,2) + g(2,2,2,1) = \binom{3}{2}\binom{3}{1} + \binom{4}{2}\binom{2}{1} = 9 + 12 = 21$ elements. The index of x in S_v is given by

$$\begin{aligned} i(x) &= x_2 g(2,2,0,2) + x_3 g(1,1,1,2) + x_6 g(1,1,-1,1) \\ &\quad + x_7 g(0,0,0,1) \\ &= \binom{2}{2}\binom{3}{1} + \binom{2}{1}\binom{2}{0} + 0 + 0 \\ &= 3 + 2 = 5. \end{aligned}$$

Thus there are 5 sequences in S_v that are lexicographically less than x. We verify the calculation by exhibiting the first few elements in S_v.

$$v = (v_{01}, v_{10}, v_{00}, v_{11}) = (2,2,1,2):$$

	0	0	1	0	1	1	1	0	$i = 0$
	0	0	1	1	0	1	1	0	$i = 1$
	0	0	1	1	1	0	1	0	$i = 2$
	0	1	0	0	1	1	1	0	$i = 3$
	0	1	0	1	1	1	0	0	$i = 4$
$x =$	0	1	1	0	0	1	1	0	$i = 5$
	0	1	1	0	1	1	0	0	$i = 6$
	0	1	1	1	0	0	1	0	$i = 7$
	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯
	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯
	1	1	1	0	1	0	0	1	$i = 20.$

Hence, since an arbitrary integer in $\{0,1,\cdots,20\}$ can be expressed with 5 binary digits, we can compress the sequences in S_v from length 8 to length 5. By using Stirling's approximation on (11) and counting the elements of S_v using Proposition 4, we obtain an approximate compression ratio of $1/n[(v_{00} + v_{10})H(v_{10}/(v_{00} + v_{10})) + (v_{11} + v_{01})H(v_{01}/(v_{00} + v_{10}))]$, the same as would be obtained from a Markov process with corresponding statistics.

REFERENCES

[1] D. H. Lehmer, "Teaching combinatorial tricks to a computer," in *Proc. Symp. Applied Mathematics*, vol. 10: *Combinatorial Analysis*. Providence, R.I.: Amer. Math. Soc., 1960.

[2] T. J. Lynch, "Sequence time coding for data compression," *Proc. IEEE* (Lett.), vol. 54, pp. 1490–1491, Oct. 1966.

[3] J. P. M. Schalkwijk, "An algorithm for source coding," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 395–399, May 1972.

[4] L. D. Davisson, "Comments on 'Sequence time coding for data compression'," *Proc. IEEE* (Lett.), vol. 54, p. 2010, Dec. 1966.

[5] P. Elias, "On binary representations of monotone sequences," in *Proc. 6th Princeton Conf. Systems Theory*, Apr. 1972.

[6] W. Feller, *An Introduction to Probability Theory and Its Applications*, vol. 1. New York: Wiley, 1957.

[7] V. F. Babkin, "Method of universal coding for a source of independent messages with non-exponential complexity of calculations" (in Russian), *Probl. Inform. Transmiss.*, vol. 7, No. 4, 1971.

[8] Yu. M. Shtarkov and V. F. Babkin, "Combinatorial method of universal coding for discrete stationary sources" (in Russian), *Probl. Contr. Inform. Transmiss.*, to be published.

[9] R. Ash, *Information Theory*. New York: Wiley, 1965.

A 2-Cycle Algorithm for Source Coding With a Fidelity Criterion

J. B. ANDERSON AND FREDERICK JELINEK

Abstract—**Since the birth of rate-distortion theory with the landmark paper of Shannon, research has centered on the extension of his theory to new situations and on the calculation of rate-distortion functions. Comparatively little has been done toward developing efficient algorithms for encoding information sources at rates near Shannon's rate-distortion limit. This paper reports one such algorithm, the 2-cycle algorithm, used with randomly chosen tree codes.**

In analyzing the 2-cycle algorithm, we present the first theoretical analysis of a realizable algorithm for tree coding of sources with a fidelity criterion. The algorithm may prove to be the first practical method of coding whose design actually derives from rate-distortion theory. Analysis proceeds by bounding methods related to difference equations and branching processes. Upper bounds on average distortion and encoding work are obtained and the latter are shown bounded as long as the coding rate exceeds Shannon's rate-distortion function. Of particular interest is the stack-searched branching process bound of Section IV. The two cycles of the algorithm are described fully in terms of expectations and probabilities, and recursions are listed to compute these quantities. Concluding the paper is numerical analysis, by theory and by simulation, with respect to the binary i.i.d. source and Hamming distortion measure. These numerical results are sufficient to optimize the algorithm over its free parameters.

I. INTRODUCTION: THE 2-CYCLE ALGORITHM

SINCE the birth of rate-distortion theory with the landmark paper of Shannon [1], research has centered on the extension of his theory to new situations and on the calculation of rate-distortion functions. Comparatively little has been done toward developing efficient algorithms for encoding information sources at rates near Shannon's rate-distortion limit. We have studied several such algorithms used with randomly chosen tree codes and report here on one of these. In so doing, we introduce several new techniques of analysis.

Consider a discrete memoryless source (DMS) with an additive measure of distortion

$$d(z^k,\hat{z}^k) = \sum_{i=1}^{k} d(z_i,\hat{z}_i),$$

where z^k is a source sequence and $\hat{z}^k$ a codeword, both of length k letters, taken from the source and reproducer alphabets, respectively.

Definition 1: A DMS is called *symmetric* if the source distribution $Q(z)$ is uniform and if the distortion matrix $[d(z,\hat{z})]$ is such that the entries in each row and each column are permutations of the same set $\{d(0,0),\cdots, d(0, C-1)\}$. Without loss of generality $d(0,0) = 0$, $d(0,\hat{z}) \geq 0$, $\hat{z} \neq 0$.

While some of our results are completely general, their totality applies to symmetric sources only.

This paper concerns the performance of an algorithm encoding the source into codewords that can be arranged in a tree structure of rate $R = \log_2 d/n$ bits per source digit, with d branches stemming from each tree node and n codeword letters on each branch [2]. Each node is located by successive digits of a path map, in which 0 represents an uppermost branch taken in reaching the node, 1 represents the next uppermost branch, etc.

The object of the encoder is to find a path of branches through the tree whose letters approximate the source

Manuscript received November 12, 1971; revised March 31, 1972. This work was supported in part by NASA Contract NAS 2-5643.

J. B. Anderson is with the Department of Electrical Engineering, McMaster University, Hamilton, Ontario, Canada.

F. Jelinek is with the School of Electrical Engineering, Cornell University, Ithaca, N.Y. 14850.