[13] D. H. Lehmer, "A note on trigonometric algebraic numbers," *Amer. Math. Monthly*, vol. 40, pp. 165-166, 1933.
[14] C. Loeffler, A. Ligtenberg, and G. S. Moschytz, "Algorithm-architecture mapping for custom DCT chips," *Proc. Int. Symp. Circuits Syst.*, Helsinki, Finland, June 1988, pp. 1953-1956.
[15] J. Makhoul, "A fast cosine transform in one and two dimensions," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 27-34, Feb. 1980.
[16] M. J. Narasinha and A. M. Peterson, "On the computation of the discrete cosine transform," *IEEE Trans. Commun.*, vol. COM-26, pp. 934-936, June 1978.
[17] I. Niven, *Irrational Numbers*. New York: John Wiley, 1967, pp. 37-38.
[18] K. Shanmugam, "Comments on discrete cosine transforms," *IEEE Trans. Comput.*, vol. C-24, p. 759, July 1975.
[19] N. Suehiro and M. Hatori, "Fast algorithms for the DFT and other sinusoidal transforms," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 642-644, June 1986.
[20] R. Tolimieri, M. An, and C. Lu, *Algorithms for Discrete Fourier Transform and Convolution*. New York: Springer-Verlag, 1989.
[21] B. D. Tseng and W. C. Miller, "On computing the discrete cosine transform," *IEEE Trans. Comput.*, vol. C-27, pp. 966-968, Oct. 1978.
[22] M. Vetterli and H. J. Nussbaumer, "Simple FFT and DCT algorithms with reduced number of operations," *Signal Processing*, pp. 267-278, Aug. 1984.
[23] M. Vetterli, "Fast 2-D discrete cosine transform," *Proc. Int. Conf. Acoust., Speech, Signal Processing*, Tampa, FL, Mar. 1985, pp. 1538-1541.
[24] S. Winograd, "On computing the discrete Fourier transform," *Math. Computat.*, vol. 32, pp. 175-199, 1978.
[25] ——, "On the multiplicative complexity of the discrete Fourier transform," *Advances Math.*, vol. 32, no. 2, pp. 83-117, 1979.
[26] ——, "Arithmetic complexity of computations," *CBMS-NSF Reg. Conf. Series in Appl. Math.*, 1980.

Source Matching Problems Revisited

Chein-I Chang and Laurence B. Wolfe

Abstract—**The source matching problem is to find the minimax codes that minimize the maximum redundancies over classes of sources where relative entropy (cross entropy, discrimination information) is adopted as a criterion to measure the redundancy. The convergence of a simple approach different from Davisson and Leon-Garcia's algorithm for finding such minimax codes is presented and shown. This approach is applied as an example to the class of first-order discrete Markov sources. The sufficient statistic previously used by Lee is corrected in his attempt to produce results for the first-order Markov source matching problem. A computational complexity analysis and a numerical study further demonstrates that this simple algorithm significantly reduces the required computing time, when compared to Davisson and Leon-Garcia's algorithm.**

Index Terms—**Source matching, minimax codes, sufficient statistic.**

I. Introduction

A source matching problem is to find a minimax code that minimizes the maximum redundancy over a class of sources where relative entropy is adopted as a criterion to measure the redundancy. The redundancy of a code for a source S is defined as a measure of the discrepancy between the performance of the code and the best possible performance for S. Source matching problems were first studied by Davisson and Leon-Garcia [1] and applications have been identified in [3]-[5], for example.

While finding a closed-form solution for the minimax codes over a class of sources with a continuous parameter space is generally not possible, Davisson and Leon-Garcia [1] did develop an algorithm that generates an approximation to a minimax code by finding the source that is best matched to the class of sources. However, a recent simple algorithm proposed by Chang *et al.* in [3] can also be modified and applied to finding source matching solutions. Since it is known that a source can be completely characterized by its entropy, the simple algorithm utilizes relative entropy as a measure of similarity to group within subclasses, all sources in the class whose entropies are within a previously assigned level of discrepancy. This approach differs from Davisson and Leon-Garcia's algorithm which iteratively seeks the localize maxima over the class of sources.

In this correspondence, we show that the simple algorithm is convergent and has less complexity, when compared with Davisson and Leon-Garcia's algorithm. This great advantage makes the simple algorithm more attractive than Davisson and Leon-Garcia's algorithm.

This correspondence is organized as follows. In Section II, source matching problems are defined. The simple algorithm is applied to source matching problems in Section III where its complexity is determined and its convergence is proven. A numerical example is studied in Section IV and conclusions are drawn in Section V.

II. The Source Matching Approach

In this section, some of the source matching results in [1] are reviewed for reference.

Consider a discrete memoryless source S with probability mass function (pmf) $P = [p_1 \cdots p_n]$. Let $L = [l_1 \cdots l_n]$ be a length function corresponding to a complete variable length code C:

$$\sum_{i=1}^{n} 2^{-l_i} = 1.$$

The average codelength is $\bar{l}(L, P) = \sum_{i=1}^{n} p_i \cdot l_i$ with redundancy

$$r(L, P) = \bar{l}(L, P) - H(P), \tag{1}$$

where $H(P)$ is the entropy of the source S. Now, if we define a pmf $Q(L) = [q_1(L) \cdots q_n(L)]$ where $q_i(L) = 2^{-l_i}$, then (1) becomes

$$\begin{aligned} r(L, P) &= \sum_{i=1}^{n} p_i \cdot l_i + \sum_{i=1}^{n} p_i \cdot \log(p_i) \\ &= \sum_{i=1}^{n} p_i \cdot \log\left(\frac{p_i}{2^{-l_i}}\right) \\ &\equiv H(P; Q(L)). \end{aligned} \tag{2}$$

This implies that $H(P; Q(L))$ can be used as a criterion to measure the discrepancy between the performance of a code and the best possible performance for the single source S. Extending this result to a class of discrete memoryless sources S, let:

$$Q(\mathcal{L}) = \{Q(L): q_i(L) = 2^{-l_i}, \text{ for some } L \in \mathcal{L}\}.$$

Manuscript received July 16, 1990. This work was supported in part by the Minta Martin Fund from the College of Engineering, University of Maryland; in part by a Summer Faculty Fellowship; and in part by a Special Research Initiative Support from the University of Maryland, Baltimore County Campus.

The authors are with the Department of Electrical Engineering, University of Maryland, Baltimore County Campus, Baltimore, MD 21228.

IEEE Log Number 9107514.

Then the problem is to find a $Q(L^*)$ which satisfies the equation:

$$R_S = \sup_{P \in S} H(P; Q; (L^*)) = \min_{Q(L) \in Q(\mathscr{L})} \sup_{P \in S} H(P; Q(L)). \quad (3)$$

If S is finite (3) can be solved by exhaustive search. Otherwise, an approximation to the code must be sought since analytic solutions do not generally exist. Assume that S is characterized by random parameter Θ; $S = \{P^\theta : \theta \in \Theta\}$, where each source has the same alphabet A. Then the problem is to seek a best matched source described by a pmf Q^*:

$$R_S = \sup_{\theta \in \Theta} H(P^\theta; Q^*) = \min_{Q(L) \in Q(\mathscr{L})} \sup_{\theta \in \Theta} H(P^\theta; Q). \quad (4)$$

Davisson and Leon-Garcia further proved a major theorem.

Source Matching Theorem:

$$\min_{Q(L) \in Q(\mathscr{L})} \sup_{\theta \in \Theta} H(P^\theta; Q) = \sup_{W \in \Xi} \min_{Q(L) \in Q(\mathscr{L})} \mathscr{H}(W; Q), \quad (5)$$

where Ξ is the set of all prior distributions defined on Θ and,

$$\mathscr{H}(W; Q) = \int_\Theta H(P^\theta; Q)\, dw(\theta).$$

However, the right-hand side of (5) is just the channel capacity between Θ and A:

$$\sup_{W \in \Xi} \min_{Q(L) \in Q(\mathscr{L})} \mathscr{H}(W; Q) = \sup_{w \in \Xi} \int_\Theta \sum_{i=1}^{n} p_i^\theta \cdot \log \left(\frac{p_i^\theta}{\bar{p}_i} \right) dW(\theta), \quad (6)$$

where

$$\bar{p}_i = \int_\Theta p_i^\theta \cdot dW(\theta).$$

They also applied (6) to source matching problems for discrete memoryless sources with a continuous parameter space and designed an algorithm to approximate a solution.

III. A Simple Algorithm for Finding Minimax Codes

Since a source matching problem can be equivalently treated as a channel capacity problem, a recent simple algorithm [3] can be modified and used in place of Davisson and Leon-Garcia's algorithm to find the minimax codes for the source matching problems with a discrete source output space and a continuous parameter space. As will be shown in this section, the simple algorithm has less complexity than Davisson and Leon-Garcia's algorithm and produces a code which converges to the optimal code for S.

For simplicity of discussion, the simple algorithm will be given assuming that the parameter $\theta \in \Theta$ lies on the real line in the closed interval $[a, b]$. It should also be noted that the simple algorithm can be applied, in general, to discrete sources. In the following we describe the simple algorithm, leaving the details to [3].

Assume that the parameter space Θ that describes a class of sources is compact and that we are given a discrete source output alphabet $A = \{1, \cdots, M\}$. Therefore, the statistics of each source in the class are determined by P^θ defined on A where $\theta \in \Theta$.

The validity of utilizing a finite set of representatives chosen from an uncountable number of sources is supported by [7, Corollary 3, p. 96], which states that for a finite output space A there is a distribution W^* over Θ that assigns a nonzero probability to only a minimal number of sources and W^* gives rise to the optimal minimax code. Also m, the size of this minimal set can be no larger than the size of the output set M, i.e., $m \le M$. Therefore, the simple algorithm must select J representatives from the subclasses where J is the size of any set of sources that contains the minimal set, i.e., $m \le M \le J$.

A partition algorithm groups all sources into J subclasses using relative entropy to measure the similarity between any two sources. Thus, it produces a finite set of J parameters $\{\theta_j\}$, which partitions $[a, b]$ into J subclasses $\{S_j\}$, where $S_j = [\theta_{j-1}, \theta_j]$. The $\{S_j\}$ are chosen such that the relative entropy between any two sources in a subclass is less than a given tolerance ϵ.

The variable r determines how fine the partition will be and ensures that the number of subclasses will at least equal the size of the discrete source output space alphabet, i.e., $J \ge M$. Given this criterion as a stopping rule, the algorithm starts with $r = 0$ and an initial error tolerance of ϵ_0. If J is less than M a smaller error tolerance $\epsilon_1 = \epsilon_0 \cdot 2^{-r}$ is generated by increasing r by 1. This procedure is repeated until $J \ge M$. Therefore, the partition algorithm by construction ensures that J representatives are selected for the class of sources, where $J \ge M$.

Partition Algorithm

1) Initialization: Set ϵ_0 = an unassigned error tolerance and $r = 0$.
2) Set $\theta_0 = a$, $\epsilon_0 \cdot 2^{-r}$ and $J = 0$.
3) Set $J = J + 1$. Find $z_J > \theta_{J-1}$ such that $H(P^{\theta_{J-1}}; P^{z_J}) = \epsilon_1$.
4) IF $z_J \ge b$, GO TO Step 7).
5) IF $\theta_{J+1} < b$, GO TO Step 3). Otherwise, continue.
6) IF $J < M$, let $r = r + 1$ and GO TO Step 2). Otherwise, let $\theta_J = b$ and output $\{\theta_j\}_{j=1}^J$ and $\{z_j\}_{j=1}^J$ and STOP.
7) If $J < M$ let, $r = r + 1$ and GO TO Step 2). Otherwise, let $z_J = b$ and output $\{\theta_j\}_{j=1}^{J-1}$ and $\{z_j\}_{j=1}^J$ and STOP.

The simple algorithm may now be given as follows.

Simple Algorithm

1) Initialization: Let ϵ = an assigned error tolerance.
2) Apply the Partition algorithm to produce $\{\theta_j\}$ and $\{z_j\}$.
3) Apply Blahut's channel capacity algorithm [6] where the output space is given by A and the input space is given by $\{z_j\}$. The $\{P^{z_j}(k)\}_{z_j \in F,\, k \in A}$ give the channel matrix $\{P(k \mid z_j)\}_{z_j \in F,\, k \in A}$.
4) STOP and output the channel capacity found in Step 3), which approximates the minimax redundancy produced by the code.

Clearly, Step 3) of the partition algorithm is an upper bound for the complexity of the simple algorithm. Although we utilized Newton's method in Step 3) of the partition algorithm to produce the results for the example in the next section, our analysis will not be predicated on a specific computational method but will be given based solely on the general requirements.

To determine the complexity, we begin by observing that regardless of the method used, Step 3) is equivalent to finding the roots of $2 \cdot J$ problems with variables z_j and θ_j. Clearly, the complexity of each problem is determined by the associated pmf. Let f^θ be defined as a function of θ that is the least upper bound (i.e., worst case) complexity of all pmf's indexed by z_j:

$$O(f^\theta) \ge O(P^{z_j}), \forall\, z_j \in \{z_j\}_{j=1}^J.$$

Therefore, the overall complexity of Step 2) and the simple algo-

rithm is given by

$$O(J \cdot f^{\theta}). \tag{7}$$

The complexity may be further reduced by observing that two sets of parameters $\{\theta_j\}$ and $\{z_j\}$ are produced by the simple algorithm although only $\{z_j\}$ is used to represent the parameter space. An obvious modification will reduce the complexity by one-half.

Davisson and Leon-Garcia's algorithm [1] however, solves a problem that is different but related to that solved by the simple algorithm. Their algorithm seeks the finite set of localized maxima $\{\theta_j\}$ over the continuous parameter space Θ to represent the class of sources S.

Davisson and Leon-Garcia's Algorithm

1) Initialization: Select an error tolerance ϵ and arbitrary set $\{\theta_j\}$ of size J, where $J \geq M$.
2) Apply Blahut's algorithm [6] to find the distribution Q.
3) Find $\{\theta_j^*\}$, the set of local maxima of $H(P^{\theta}; Q^*)$. $\{\theta_j\} \leftarrow \{\theta_j^*\}$.
4) If $\mathscr{H}(W; Q) - \max_{\theta \in \Theta} H(P^{\theta}; Q) \leq \epsilon$, then stop else GO TO 2.
5) Output the channel capacity that approximates the minimax redundancy.

While the local maxima may equal global maxima in the simplest classes of sources, more complex sources will have additional extrema. Therefore in general, the number of extrema to be evaluated will be a function of J, say $g(J)$, where $g(J) \geq J$. Although many methods are available to identify and evaluate extrema, a generalized approach will include taking derivatives of functions (i.e., pmf's in this case) and finding the roots of the resulting equations. Although repeated iterations may be required we will assume that derivatives and roots are found problems in just one iteration. To be consistent with our previous approach we again utilize $O(f^{\theta})$ as the complexity of finding roots. Thus, the complexity of each of the $g(J)$ problems is $O(f^{\theta} + 1)$ and the complexity of the algorithm is

$$O(|g(J)| \cdot f^{\theta} + |g(J)|). \tag{8}$$

Clearly the simple algorithm is more efficient since

$$O(|g(J)| \cdot f^{\theta} + |g(J)|) > O(J \cdot f^{\theta}). \tag{9}$$

It should also be noted that the redundancy as calculated by the simple algorithm converges to that for the optimal code.

Theorem 1: The approximation to the minimax code calculated by the simple algorithm converges to the optimal minimax code for S and, the redundancy for this approximate code is bounded from above by the redundancy for the approximation to the minimax code as calculated by Davisson and Leon-Garcia's algorithm.

Proof: To prove the first part we need only show that in the limit the redundancy as calculated by the simple algorithm converges to zero. This is true because [7, Theorem 4.5.1] tells us that channel capacity (i.e., minimax redundancy) is achieved when

$$\mathscr{H}(W; Q^*) = H(P^{\theta_j}; Q^*), \tag{10}$$

where $P^{\theta_j} > 0$. Also, Chang [6] previously proved convergence of Davisson and Leon-Garcia's algorithm, that is,

$$\lim_{\substack{N \to \infty \\ \epsilon \to 0}} H(P^{\theta_j}; Q^*) = 0. \tag{11}$$

However, the simple algorithm selects a representative z_j for the partition containing θ_j, and

$$H(P^{z_j}; P^{\theta_j}) \leq \epsilon.$$

Thus, for every ϵ, there exists a J depending on ϵ, such that

$$H(P^{z_j}; Q^*) - H(P^{\theta_j}; Q^*) \leq \epsilon.$$

Thus, in the limit,

$$\lim_{\substack{J \to \infty \\ \epsilon \to 0}} \left(H(P^{z_j}; Q^*) - H(P^{\theta_j}; Q^*)\right) = 0, \tag{12}$$

as was to be shown.

The second part of the theorem can be easily seen to be true by observing that the simple algorithm maps S to a new class of sources S'. This can be viewed as encoding S by S' and encoding Θ by $\{z_j\}$, then subsequently computing the channel capacity (i.e., minimax redundancy) between the source output space and Θ. The data processing theorem [7, p. 80] tells us that the channel capacity for A and $\{z_j\}$ cannot exceed the channel capacity for A and Θ. Thus, the minimax code as calculated by the simple algorithm, converges from below to the redundancy of the minimax code as calculated by Davisson and Leon-Garcia's algorithm. □

This result shows that Davisson and Leon-Garcia's algorithm may produce a better approximation to the optimal code. However, the advantage quickly diminishes since both produce codes that converge and the simple algorithm has less complexity.

IV. Numerical Example

A. Discrete First-Order Binary Markov Sources

In this example, we examine a class of sources with memory where the parameter $\theta = [\theta_0, \theta_1]$ takes values in $[0, 1] \times [0, 1]$ and has stochastic transition matrix

$$\begin{bmatrix} \theta_{00} & \theta_{01} \\ \theta_{10} & \theta_{11} \end{bmatrix} = \begin{bmatrix} 1 - \theta_0 & \theta_0 \\ \theta_1 & 1 - \theta_1 \end{bmatrix}, \tag{13}$$

where θ_{ij} represents the transition probability from the previous letter i to j. For simplicity of notation, θ_{01} is denoted by θ_0 and θ_{10} by θ_1. The stationary pmf is easily shown to be

$$\pi = \left(\frac{\theta_1}{\theta_0 + \theta_1}, \frac{\theta_0}{\theta_0 + \theta_1}\right) = (\pi_0, \pi_1),$$

and consequently the pmf for each Markov source indexed by θ is given by

$$p(x^N | \theta) = \pi_h \cdot (1 - \theta_0)^i \cdot \theta_0^j \cdot \theta_1^k \cdot (1 - \theta_1)^{N-i-j-k-1}, \tag{14}$$

where i, j, and k are the numbers of transitions and π_h is the probability of initial letter h. For example, if $x^5 = [10011]$, then

$$p(x^5 | \theta) = \pi_h \cdot \theta_1 \cdot (1 - \theta_0) \cdot \theta_0 \cdot (1 - \theta_1) = (\theta_0 + \theta_1)^{-1} \cdot (1 - \theta_0) \cdot \theta_0^2 \cdot \theta_1 \cdot (1 - \theta_1).$$

Although (14) may be utilized, the source matching problem soon

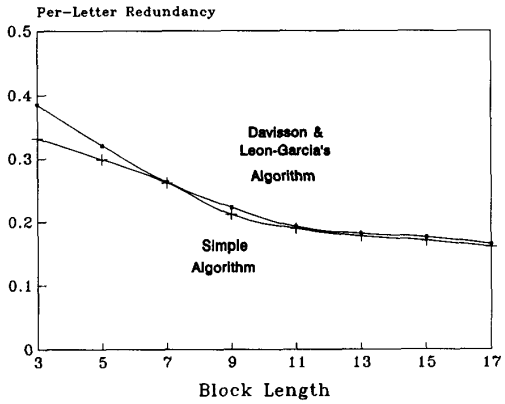

Fig. 1. Minimax redundancy for the class of first-order Markov sources.

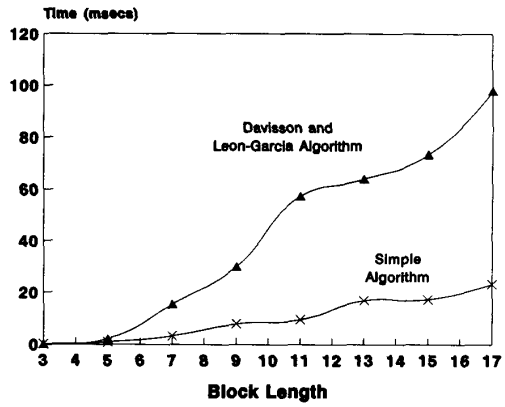

Fig. 2. Computer time for two algorithms.

becomes intractable for increasing block length. To overcome this difficulty, a sufficient statistic [8] based on the number of runs of consecutive 1's that appear in a message block may be used to reduce complexity. For example, $x^5 = [10011]$ has 1 run of length 1 and 1 run of 1's of length 2. Summarizing the sufficient statistic, given $x^N = [x_1 \cdots x_N]$ with n 1's, let a_i be the number of runs of consecutive 1's with length i, where i ranges from 1 to n. Let D_n be the total number of all the runs of 1's with length up to n. With x_1 and x_N, the first and last symbols of message block x^N fixed, (n, D_n, x_1, x_N) uniquely specifies $p(x^N \mid \theta)$. Define the following three cases:

$$(n, D_n, 1) \equiv (n, D_n, 0, 0),$$

$$(n, D_n, 2) \equiv (n, D_n, 0, 1) \text{ or } (n, D_n, 1, 0),$$

$$(n, D_n, 3) \equiv (n, D_n, 1, 1). \tag{15}$$

The sufficient statistic can now be defined as $T(x^N) = (n, D_n, i)$, where $i \in \{1, 2, 3\}$. Let $\delta_{i,j} = 1$ for $i = j$ and 0, elsewhere and $\binom{i}{j} = i!/j!(i-j)!$ if i, j and $(i-j) \geq 0$ and equals 0, elsewhere. Then,

$$\begin{aligned} M_{(n, D_n, i)} &\equiv \{x^N \in \{0,1\}^N : T(x^N) = (n, D_n, i)\} \\ &= (1 + \delta_{i,2}) \cdot \binom{N-n+1}{D_n - i + 1} \cdot \binom{n-1}{D_n - 1} \\ &\quad + \delta_{N-n,0} \cdot \delta_{D_n,1} + \delta_{n,0} \cdot \delta_{D_n,0} \end{aligned} \tag{16}$$

is the number of message blocks with (n, D_n, i) and clearly

$$\sum_{\substack{(n, D_n, i) \\ n<N \\ i=1,2,3}} M_{(n, D_n, i)} = 2^N.$$

Consequently, (16) is used to yield the desired pmf

$$\begin{aligned} p(T(x^N) = (n, D_n, i) \mid \theta) = \frac{M_{(n, D_n, i)}}{(\theta_0 + \theta_1)} \\ \cdot (1 - \theta_0)^{N-n-D_n+i-2} \cdot \theta_0^{D_n} \\ \cdot \theta_1^{D_n - i + 2} \cdot (1 - \theta_1)^{n - D_n}. \end{aligned} \tag{17}$$

This sufficient statistic corrects one proposed by Lee in [2], which did not account for boundary conditions and reduces the number of calculations needed from 2^N to

$$\sum_{n, D_n, i} (1 - \delta_{0,(n, D_n, i)}).$$

The simple algorithm and Davisson and Leon-Garcia's algorithm were both applied to this source matching problem and the results are shown in Figs. 1 and 2.

V. Conclusion

In this correspondence we revised source matching problems in order to present a simple algorithm for finding a source matching code that achieves the minimax redundancy. This simple algorithm has a simple structure that can be easily implemented on computers. The simple algorithm was applied to a numeric example and a sufficient statistic was also derived that corrects Lee's [2] prior attempt to extend the source matching approach to first-order Markov sources. The example and a computational complexity analysis show that the proposed approach is more efficient than Davisson and Leon-Garcia's algorithm [1]. In particular, the simple algorithm performs nearly as well as Davisson and Leon-Garcia's algorithm with a significant savings in computing time.

References

[1] L. D. Davisson and A. Leon-Garcia, "A source matching approach to finding minimax codes," *IEEE Trans. Inform. Theory*, vol. IT-26, pp. 166-174, Mar. 1980.

[2] D. H. Lee, "The source matching approach for Markov sources," *IEEE Trans. Inform. Theory*, vol. IT-29, pp. 754-755, Sept. 1983.

[3] C.-I. Chang, S. C. Fan, and L. D. Davisson, "A simple method of calculating channel capacity and finding minimax codes for source matching problems," presented at *1988 Conf. Inform. Sci. and Syst.*, Princeton Univ., Princeton, NJ, Mar. 16-18, 1988, pp. 362-366.

[4] C.-I. Chang and L. D. Davisson, "An entropy constrained quantization approach for a source characterized by random parameters," in *Abstracts of Papers in Proc. 1990 IEEE Int. Symp. Inform. Theory*, San Diego, CA, Jan. 14-19, 1990.

[5] C.-I. Chang, L. Fan, and L. D. Davisson, "Computation of the rate distortion function for a source with uncertain statistics," in *Proc. IEEE Int. Conf. Commun. Syst.*, Singapore, Oct. 31-Nov. 3, 1988, pp. 1180-1184.

[6] R. E. Blahut, "Computation of channel capacity and rate distortion functions," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 460-473, July 1972.

[7] R. G. Gallagher, *Information Theory and Reliable Communications*. New York: Wiley, 1968.

[8] L. B. Wolfe and C.-I. Chang, "Source matching problems revisited," in *Proc. Int. Conf. Signal Processing '90*, Beijing, China, Oct. 22-26, 1990, pp. 119-122.

On Extremal Self-Dual Ternary Codes of Lengths 28 to 40

W. Cary Huffman

Abstract—**The extremal self-dual ternary codes of lengths 28, 32, and 36 with monomial automorphisms of prime order $r \geq 5$, and of length 40 with monomial automorphisms of prime order $r > 5$ are enumerated. For each length and prime considered, we find all inequivalent extremal codes with an automorphism of that order.**

Index Terms—**Ternary codes, self-dual codes, extremal codes.**

I. Introduction

In [4] and [5], we developed a general decomposition theory for self-dual linear codes $\mathscr{C}$ over a finite field F_q when $\mathscr{C}$ has a permutation automorphism of prime order r relatively prime to q. In Section II, we summarize these results. Also in [4], [5], [7], and [16] methods were developed for deciding, under certain conditions, whether two codes with the same automorphism are equivalent. In Section II, we extend these methods to allow us to treat the ternary codes under consideration.

In [1], [12], and [14] all self-dual ternary codes of lengths 20 or less are enumerated. In [10] Leon, Pless, and Sloane show that there are only 2 inequivalent [24, 12, 9] ternary self-dual codes. This classification required the complete enumeration of all 24×24 Hadamard matrices (see [8]). Such complete enumeration using these techniques seems infeasible for higher lengths. In the present paper, we give a partial enumeration of extremal codes for the next four lengths in which such codes exist. In particular, in Section III we enumerate the extremal self-dual ternary codes of lengths 28, 32, and 36 having a monomial automorphism of prime order $r \geq 5$ and of length 40 having a monomial automorphism of prime order $r > 5$. Similar classifications for quaternary codes of lengths 18, 20, 22, 26, and 28 are given in [5] and [6]. It is interesting to note that the numbers of extremal ternary codes of lengths 28, 32, 36, and 40 are quite a bit larger than the numbers of extremal quaternary codes of lengths 18, 20, 22, 24, 26, and 28. (See also [9].) For example, there are 239 inequivalent [32, 16, 9] ternary self-dual codes with an automorphism of order 5, far exceeding the numbers for any prime and any length considered in [5] or [6].

General references to coding theory are [11] and [15].

II. Decomposition and Equivalence

We summarize the theory developed in [4] and [5] on the decomposition of codes. Let F_q be the finite field of order q and characteristic p. Suppose r is relatively prime to p. Let $R = F_q[X]/\langle X^r - 1\rangle$, where X is an indeterminate. Suppose $X^r - 1 = \Pi_{j=0}^{g} m_j(X)$, where $m_j(X)$ is irreducible over F_q and $m_0(X) = X - 1$. Let $I_j = \langle (X^r - 1)/m_j(X)\rangle$ be the ideal of R generated by $(X^r - 1)/m_j(X)$ for $0 \leq j \leq g$. By Lemma 1 of [4], $R = I_0 \oplus I_1 \oplus \cdots \oplus I_g$, I_j is a field for $0 \leq j \leq g$, and $I_j I_k = \{0\}$ if $j \neq k$. Let $\tau_{p^\alpha, u}: R \to R$ be given by $\tau_{p^\alpha, u}(\sum_{i=0}^{r-1} a_i X^i) = \sum_{i=0}^{r-1} a_i^{p^\alpha} X^{ui}$ where $\gcd(r, u) = 1$. By Lemma 1 of [5], $\tau_{p^\alpha, u}$ is a field automorphism of I_0, permutes $I_1, \cdots, I_g$, and if $\tau_{p^\alpha, u}(I_j) = I_k$, $\tau_{p^\alpha, u}$ is a field isomorphism of I_j onto I_k.

Let $\mathscr{C}$ be a linear code over F_q of length n and dimension k. The *weight* of a vector $x \in F_q^n$ is the number of nonzero entries of x. The *minimum distance* d of $\mathscr{C}$ is the minimum nonzero weight of all codewords in $\mathscr{C}$. $\mathscr{C}$ is called an $[n, k]$ or $[n, k, d]$ code. Let σ be a permutation of the coordinates of F_q^n. If $x \in F_q^n$ has ith coordinate x_i, define $(x\sigma)_i = x_{i\sigma^{-1}}$; σ is a *permutation automorphism* of $\mathscr{C}$ if $x\sigma \in \mathscr{C}$ for all $x \in \mathscr{C}$. Assume σ has only c r-cycles and f fixed points. Denote the r-cycles by $\Omega_1, \cdots, \Omega_c$ and the fixed points by $\Omega_{c+1}, \cdots, \Omega_{c+f}$. Let $x|_{\Omega_i}$ be the restriction of x to Ω_i. For $1 \leq i \leq c$, $x|_{\Omega_i}$ can be viewed as an element $a_0 + a_1 X + \cdots + a_{r-1} X^{r-1} \in R$, where $x\sigma|_{\Omega_i}$ is $(a_0 + a_1 X + \cdots + a_{r-1} X^{r-1}) X \in R$. Let $C(\sigma) = \{x \in \mathscr{C} \mid x\sigma = x\}$, and for $1 \leq j \leq g$, $E_j(\sigma) = \{x \in \mathscr{C} \mid x|_{\Omega_i} \in I_j$ for $1 \leq i \leq c$ and $x|_{\Omega_i} = 0$ for $c + 1 \leq i \leq c + f\}$. By Lemma 2 of [4], $C(\sigma)$ and $E_j(\sigma)$ are σ – invariant and $\mathscr{C} = C(\sigma) \oplus E_1(\sigma) \oplus \cdots \oplus E_g(\sigma)$. Let $E_j(\sigma)^*$ be $E_j(\sigma)$ with the fixed points $\Omega_{c+1}, \cdots, \Omega_{c+f}$ deleted and the codewords viewed as c-tuples from I_j^c.

Suppose we have the inner product $\langle \cdot, \cdot \rangle$ on F_q^n of the form

$$\langle u, v \rangle = \sum_{i=1}^{n} u_i v_i^{p^m}, \tag{1}$$

where $u, v \in F_q^n$ with $u = (u_1, \cdots, u_n)$ and $v = (v_1, \cdots, v_n)$. Define $\mathscr{C}^{\perp L} = \{u \in F_q^n \mid \langle u, v \rangle = 0$ for all $v \in \mathscr{C}\}$. $\mathscr{C}$ is *left self-orthogonal* under (1) if $\mathscr{C} \subseteq \mathscr{C}^{\perp L}$ and *left self-dual* if $\mathscr{C} = \mathscr{C}^{\perp L}$. Define $\mathscr{C}^{\perp R}$, right self-orthogonality, and right self-duality analogously. If $\mathscr{C}^{\perp L} = \mathscr{C}^{\perp R}$, define $\mathscr{C}^{\perp} = \mathscr{C}^{\perp L}$; in this case if $\mathscr{C} \subseteq \mathscr{C}^{\perp}$, $\mathscr{C}$ is *self-orthogonal*, and if $\mathscr{C} = \mathscr{C}^{\perp}$, $\mathscr{C}$ is self-dual. (This is the case when considering ternary codes as $p = q = 3$ and $p^m = 1$ in (1).) The decomposition theorem of [5] is as follows.

Theorem 1: Let s, t be nonnegative integers with $s \leq m$. Choose an integer u such that $p^s q^t u \equiv -1 \pmod r$. Let $(\cdot, \cdot)$ be the form on R^c given by

$$(x, y) = \sum_{i=1}^{c} x_i y_i^{p^s q^t}. \tag{2}$$

Let λ be the permutation on $1, \cdots, g$ where $\tau_{p^{m-s}, u}(I_i) = I_{\lambda(i)}$ and let $\mathscr{O}_1, \cdots, \mathscr{O}_l$ be the orbits of λ. If $\mathscr{C}$ is a left self-dual $[n, n/2]$ code under (1) with permutation automorphism σ, then $C(\sigma)$ is a left self-orthogonal $[n, (c + f)/2]$ code under (1), and for $1 \leq i \leq g$, $E_{\lambda(i)}(\sigma)^* = (\tau_{p^{m-s}, u}(E_i(\sigma)^*))^{\perp L}$ under (2). Conversely, if $C(\sigma)$ is a left self-orthogonal $[n, (c + f)/2]$ code under (1) and if $E_{\lambda(i)}(\sigma)^* = (\tau_{p^{m-s}, u}(E_i(\sigma)^*))^{\perp L}$ under (2) for $1 \leq i \leq g$, then $\mathscr{C}$ is left self-dual under (1). In addition, c is even if $|\mathscr{O}_j|$ is odd for some j.

Define $\mathscr{M}_n(q)$ as the group of all $n \times n$ monomial matrices over F_q, and define $\mathscr{M}_n^*(q)$ as the semidirect product of $\mathscr{M}_n(q)$ extended by $\mathrm{Gal}(F_q)$, which is the Galois group of F_q over F_p. A map $T \in \mathscr{M}_n^*(q)$ can be written as $T = PD\tau$ where P is a permutation matrix (*permutation part*), D is a diagonal matrix (*diagonal part*), and $\tau \in \mathrm{Gal}(F_q)$. Linear codes $\mathscr{C}$ and $\mathscr{C}'$ are *equivalent* whenever $\mathscr{C}' = \mathscr{C}T$ for some $T \in \mathscr{M}_n^*(q)$. Define $G(\mathscr{C}) = \{M \in \mathscr{M}_n(q) \mid \mathscr{C}M = \mathscr{C}\}$ and $G^*(\mathscr{C}) = \{T \in \mathscr{M}_n^*(q) \mid \mathscr{C}T = \mathscr{C}\}$; $G^*(\mathscr{C})$ is the *automorphism group* of $\mathscr{C}$. Note that in the case of ternary codes, $\mathscr{M}_n^*(3) = \mathscr{M}_n(3)$ and $G^*(\mathscr{C}) = G(\mathscr{C})$. Also if $\mathscr{C}$ is a self-dual ternary code under (1), with $p^m = 1$, $\mathscr{C}T$ for $T \in \mathscr{M}_n(3)$ is also self-dual. The following result implies that in the ternary case, when considering monomial automorphisms of prime order

Manuscript received January 8, 1991.

The author is with the Department of Mathematical Sciences, Loyola University of Chicago, Chicago, IL 60626.

IEEE Log Number 9107516.