Weight Distribution of Translates, Covering Radius, and Perfect Codes Correcting Errors of Given Weights

MARK KARPOVSKY, MEMBER, IEEE

Abstract—Let V be a binary linear (n, k) code defined by a check matrix H and let $h(x)$ be the characteristic function for the set of columns of H. Connections between the Walsh transform of $h(x)$ and the weight distributions of all translates of the code are obtained. Explicit formulas for the weight distributions of translates are given for small weights $i (i < 8)$. The computation of the weight distribution of all translates (including the code itself) for $i < 8$ requires at most $7(n-k)2^{n-k}$ additions and subtractions, $6 \cdot 2^{n-k}$ multiplications and 2^{n-k+1} memory cells. This method may be very effective if there is an analytic expression for $h(x)$. A simple method for computing the covering radius of the code by the Walsh transform of $h(x)$ is described. The implementation of this method requires for large n at most $2^{n-k}(n-k)\log_2(n-k)$ arithmetical operations and 2^{n-k+1} memory cells. We define the concept L-perfect for codes, where L is a set of weights. After describing several linear and nonlinear L-perfect codes, necessary and sufficient conditions for a code to be L-perfect in terms of the Walsh transform of $h(x)$ are established. An analog of the Lloyd theorem for such codes is proved.

I. Introduction

SECTION II of this paper is a companion to [8], with which we assume the reader is familiar. Suppose that the binary linear (n, k) code V is defined by its $(n-k) \times n$ check matrix H with columns $h_1, \cdots, h_n$. We assume that the code V has a distance dist $(V) > 2$.

Let $f \in \{0,1\}^n$ ($\{0,1\}^n$ is the set of all binary n-vectors), and $A_i(f)$ be the number of vectors of weight i which belong to the translate (coset) $V \oplus f$ of our code V (the symbol $\oplus$ stands for component wise addition mod 2). The problem of determining $\{A_i(f)\}$ (and especially the weight distribution $\{A_i(0)\}$ of the code itself) has been studied intensely (see [1]–[8]).

Let $h(x)$ be the characteristic function for the set of columns of H, that is, $h(x) = 1$ if $x \in \{h_1, \cdots, h_n\}$ and $h(x) = 0$ otherwise. Then $h(x)$ is a Boolean function of $n-k$ arguments $x^{(1)}, \cdots, x^{(n-k)}$, and the Walsh (Hadamard, Fourier) transform $\hat{h}(\omega)$ of $h(x)$ [8], [9], may be defined by the formula

$$\hat{h}(\omega) = \sum_x h(x)(-1)^{x \cdot \omega}, \tag{1}$$

where $\omega = (\omega^{(1)}, \cdots, \omega^{(n-k)})$ is any binary $(n-k)$ vector and

$$x \cdot \omega = \sum_{r=1}^{n-k} x^{(r)} \omega^{(r)}. \tag{2}$$

Connections between $\hat{h}(\omega)$ and the weight distribution $\{A_i(0)\}$ of the code were established in [8], where they led to another proof of the Pless ith power moment identities and a simple method for computing $A_i(0)$ for small i $(i \leq 7)$.

In Section II of this paper, we generalize the method described in [8] to the case of weight distribution of translates. Namely, we shall establish the connections between $\hat{h}^i(Hf)$ (where $\hat{h}^i$ is the ith power of $\hat{h}$) and $A_i(f)$ for all f. This will provide us with a simple method for simultaneous computing of the weight distributions of all translates of the code (including the code itself). This method requires about $s(n-k)2^{n-k}$ additions and subtractions, $(s-1)2^{n-k}$ multiplications and 2^{n-k+1} memory cells for the computations of $A_i(f)$ for all $f \in \{0,1\}^n$ and all $i = 0, \cdots, s$ for small s $(s < 8)$. Since the complexity of computations depends only on $n-k$ and s, this method may be very useful for the important practical case when n and k are large, and $n-k$ is comparatively small for codes with small distances and independent channel errors. We also note that for many cases we may derive simple analytical expressions for $A_i(f)$ for all f. This situation will be illustrated by examples in Section II.

The *covering radius* $t(V)$ (the true external distance) of the code V is defined by

$$t(V) = \max_{f \in \{0,1\}^n} \min_{v \in V} \text{dist}(v, f), \tag{3}$$

where dist (v, f) is the Hamming distance between v and f. The calculation of the covering radius for the given code is an important and difficult problem (see, e.g., [1]). One approach to the solution of this problem is described in Section III. This approach will be based on the results from Section II about weight distributions of translates; its implementation requires for $n \to \infty$ at most $2^{n-k}(n-k)\log_2(n-k)$ arithmetical operations and 2^{n-k+1} memory cells. The exact complexity estimations will also be given in Section III.

Let $L = \{0, l_1, \cdots, l_s\}$ and $0 < l_1 < l_2 < \cdots l_s \leq n$. Let us say that an (n, k) *code* V *corrects errors of weights* $l_1, \cdots, l_s$ if and only if (iff) there is at most one vector with

Manuscript received July 2, 1979; revised March 18, 1980. This work was supported in part by the National Science Foundation under Grant MCS-8008339.

The author is with the Computer Science Department, School of Advanced Technology, State University of New York at Binghamton, Binghamton, NY 13901.

0018-9448/81/0700-0462$00.75

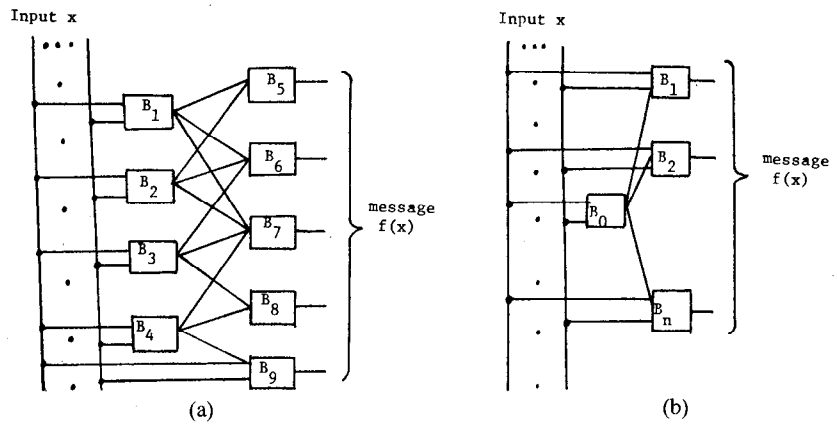

Fig. 1. Logical networks of generators of messages for Example I.

a weight from L in each translate of the code V. The problem of correction (or detection) or errors of given weights appears, for example, when the generator of messages is a logical network and errors in a message appear as a result of physical failures in the blocks of this network (see Example 1, just below), or in the case of a band-limited generator of noise. Another example may be the case when as a result of an error, either all components of a message are distorted or the number of distorted components is at most l (for this case $L = \{0, 1, \cdots, l, n\}$). The general problem of error detection and error correction of an arbitrary set of errors was considered in [10]–[12].

Example 1: Suppose that the generator of messages is represented by the logical network of Fig. 1(a), where $B_1, \cdots, B_9$ are some logical networks. For this network any physical failure in one of the logical networks $B_1, \cdots, B_9$ may result only in distortion of one or three components in the messages, $L = \{0, 1, 3\}$. Similarly, for the generator of messages represented by the hierarchical logical structure of Fig. 1(b), we have for any single failure in logical networks $B_0, \cdots, B_n$, $L = \{0, 1, n\}$. For an (n, k) code correcting errors of weights $l_1, \cdots, l_s$, we have the following analog of the Hamming bound:

$$\sum_{l \in L} \binom{n}{l} \leq 2^{n-k}, \tag{4}$$

where $L = \{0, l_1, \cdots, l_s\}$. Let us say that such an (n, k) code is *L-perfect* iff equality holds in (4). Intense study of the classical case $L = \{0, 1, \cdots, l\}$ [1] has shown that very few such perfect codes exist [13], [21].

We shall describe in Section IV that very few perfect codes correcting errors of given weights exist, as in the classical case $L = \{0, 1, \cdots, l\}$ [1], [13]. One class of nonlinear L-perfect codes for $L = \{0, 1, n\}$ is also described. The generalization of the results to the case of nonbinary codes over GF(q) is discussed in Section V.

II. Weight Distribution of Translates of a Code

Let V be the binary linear (n, k) code with the check matrix $H = (h_1, \cdots, h_n)$, and let $C_i(f)$ be the number of i-tuples $(h_{r_1}, \cdots, h_{r_i})$ of (not necessarily distinct) vectors from $\{h_1, \cdots, h_n\}$ such that 1) $h_{r_1} \oplus \cdots \oplus h_{r_i} = Hf$, and 2) there exists $\alpha, \beta \in \{1, \cdots, i\}$ such that $h_{r_\alpha} = h_{r_\beta}$ and $\alpha \neq \beta$. (Note that any rearrangement of an i-tuple counts as a different i-tuple.)

Theorem 1: In any translate $V \oplus f$ of the (n, k) code V the number of vectors of weight i is given by

$$A_i(f) = (i!)^{-1}\left(2^{-(n-k)}\hat{\hat{h}}^i(Hf) - C_i(f)\right), \qquad i = 0, 1, \cdots, n, \tag{5}$$

where $\hat{h}^i(Hf)$ is the ith power of $\hat{h}(Hf)$.

Proof: The same proof as that of Theorem 1 in [8] works for this result, if the reader makes only the necessary and obvious changes such as replacing "0" with "Hf" and "$\Sigma_\omega \hat{h}^i(\omega)$" with "$\hat{\hat{h}}^i(Hf)$."

Corollary 1: The translate $V \oplus f$ of the (n, k) code V with the check matrix $H = (h_1, \cdots, h_n)$ has weight t if and only if

$$\hat{\hat{h}}^i(Hf) = 0, \qquad \text{for all } i = 0, \cdots, t-1,$$

$$\hat{\hat{h}}^t(Hf) \neq 0.$$

For this case

$$A_t(f) = (t!)^{-1} 2^{-(n-k)} \hat{\hat{h}}^t(Hf). \tag{6}$$

Proof: From the definition of $C_i(f)$ and (1), (2) we have

1) if t is the weight of $V \oplus f$, then $C_i(f) = 0$ for all $i = 0, \cdots, t$;
2) if $\hat{\hat{h}}^i(Hf) = 0$ for all $i = 0, \cdots, t-1$, then $C_i(f) = 0$ for all $i = 0, \cdots, t-1$.

Corollary 1 follows now immediately from Theorem 1.

Let us consider now the behavior of weight distributions under translations of columns of the check matrix. Let V be a code with the check matrix H, and $h(x)$ be the characteristic function for H. We shall say that a code $V^{(T)}$

with the characteristic function $h^{(T)}(x)$ is a *check-translation* of V iff

$$h^{(T)}(x) = h(x \oplus T) \qquad \text{for some } T \in \{0,1\}^{n-k} \left(T \notin \{h_1,\cdots,h_n\}\right). \quad (7)$$

For a check matrix $H^{(T)} = (h_1^{(T)},\cdots,h_n^{(T)})$ of $V^{(T)}$ we have from (7)

$$h_{r_1}^{(T)} \oplus \cdots \oplus h_{r_i}^{(T)} = h_{r_1} \oplus \cdots \oplus h_{r_i} \oplus \frac{1}{2}\left(1-(-1)^i\right)T,$$

for any $1 \le r_1 < \cdots < r_i \le n$,

$$H^{(T)}f = Hf \oplus \frac{1}{2}\left(1-(-1)^{wt(f)}\right)T. \quad (8)$$

Thus, we have from (7), (8) for a weight distribution $\{A_i^{(T)}(f)\}$ of a check-translation $V^{(T)}$

$$A_i^{(T)}(f) = A_i\left(f \oplus \frac{1}{2}\left(1-(-1)^{i+wt(f)}\right)T\right). \quad (9)$$

If we obtain analytic expressions for $C_i(f)$, then Theorem 1 provides us with a simple method for the simultaneous computations of weight distributions for all translates of the given code (including the code itself). We shall describe now a combinatorial method for the computation of $C_i(f)$ which is expedient for comparatively small i (say, $i < 8$).

Let $P_s(i)$ be an s-partition for the integer i; for example, $P_4(7) = (3+2+1+1)$ is a 4-partition for 7. We shall use the notation (1^2), etc., to denote the partition $1+1$; for example, $P_4(7) = (3+2+1+1) = (3,2,1^2)$. For an s-partition $P_s(i) = (r_1^{\alpha_1},\cdots,r_t^{\alpha_t})$ $(\alpha_1+\cdots+\alpha_t = s,\ \alpha_1 r_1 + \cdots + \alpha_t r_t = i)$ we denote by $d(P_s(i), f)$ $(f \in \{0,1\}^n)$ the number of vectors $x = (x_1,\cdots,x_i)$ (with not necessarily distinct components) such that

1) $x_1,\cdots,x_i \in \{h_1,\cdots,h_n\}$, $\oplus_{j=1}^i x_j = Hf$; and
2) there exists α_1 elements of $\{h_1,\cdots,h_n\}$ such that each of them appears r_1 times as a component in x; there exists α_2 other elements of $\{h_1,\cdots,h_n\}$ such that each of them appears r_2 times as a component in the same vector x, etc.

For example, for $P_4(7) = (3,2,1^2)$, x may be chosen as $(x_1, x_1, x_1, x_2, x_2, x_3, x_4)$ or $(x_1, x_1, x_2, x_1, x_2, x_3, x_4)$, etc., where $x_1 \oplus x_3 \oplus x_4 = Hf$ and x_1, x_2, x_3, x_4 are all different. To compute $d((3,2,1^2), f)$ we note that there are $\binom{7}{3}$ different ways to choose three components where the given x_1 appear in x; $\binom{4}{2}$ different ways to choose two components for the given x_2; there are $A_3(f)3!$ ways to choose (x_1, x_3, x_4) such that $x_1 \oplus x_3 \oplus x_4 = Hf$; and $n-3$ ways to choose x_2 such that

$$x_2 \in \{h_1,\cdots,h_n\} - \{x_1, x_3, x_4\}.$$

Thus,

$$d\left((3,2,1^2), f\right) = A_3(f)3!\binom{7}{3}\binom{4}{2}(n-3).$$

We note that by definitions of $A_i(f)$ and $d(P_s(i), f)$ we have

$$A_i(f) = (i!)^{-1} d\left((1^i), f\right), \quad (10)$$

$$C_i(f) = \sum_{s=1}^{i-1} \sum_{P_s(i)} d(P_s(i), f). \quad (11)$$

Thus, the problem of determining $C_i(f)$ may be reduced to the computation of $d(P_s(i), f)$ for all s and $P_s(i)$ $(s = 1,\cdots,i-1)$. Values of $d(P_s(i), f)$ are given in Table I for $i = 1,\cdots,7$. Using the results from Table I, Theorem 1, and (10), (11), we have the following corollary.

Corollary 2: For any translate $V \oplus f$ of the (n,k) code V

$$A_0(f) = 2^{-(n-k)}\hat{h}^0(Hf) = \delta_{0,Hf},$$

$$A_1(f) = 2^{-(n-k)}\hat{h}(Hf) = h(Hf),$$

$$A_2(f) = (2!)^{-1}\left(2^{-(n-k)}\hat{h}^2(Hf) - A_0(f)n\right), \quad (12)$$

$$A_3(f) = (3!)^{-1}\left(2^{-(n-k)}\hat{h}^3(Hf) - A_1(f)(3n-2)\right), \quad (13)$$

$$A_4(f) = (4!)^{-1}\left(2^{-(n-k)}\hat{h}^4(Hf) - A_2(f)\cdot 2!2(3n-4) - A_0(f)n(3n-2)\right), \quad (14)$$

$$A_5(f) = (5!)^{-1}\left(2^{-(n-k)}\hat{h}^5(Hf) - A_3(f)3!10(n-2) - A_1(f)(15n^2-30n+16)\right), \quad (15)$$

$$A_6(f) = (6!)^{-1}\left(2^{-(n-k)}\hat{h}^6(Hf) - A_4(f)\cdot 4!5(3n-8) - A_2(f)2!\,(45n^2-150n+136) - A_0(f)(15n^2-30n+16)n\right), \quad (16)$$

$$A_7(f) = (7!)^{-1}\left(2^{-(n-k)}\hat{h}^7(Hf) - A_5(f)5!7(3n-10) - A_3(f)3!7(13+5(n-3)(3n-5)) - A_1(f)(1+7(n-1)(9+15(n-1)(n-2)))\right). \quad (17)$$

We note that (12)–(16) generalize the corresponding results from [8].

TABLE I
VALUES OF $d(P_s(i), f)$ FOR $i = 1, \cdots, 7$

i	s	$P_s(i)$	$d(P_s(i), f)$
1	1	(1)	$A_1(f) = h(Hf)$
2	1	(2)	$A_0(f)n = n\delta_{0, Hf}$
	2	(1^2)	$A_2(f)2!$
3	1	(3)	$A_1(f)$
	2	(2, 1)	$A_1(f)\binom{3}{1}(n-1)$
	3	(1^3)	$A_3(f)3!$
4	1	(4)	$A_0(f)n$
	2	(3, 1)	$A_2(f)2!\binom{4}{1}$
	2	(2^2)	$A_0(f)(2!)^{-1}\binom{4}{2}n(n-1)$
	3	$(2, 1^2)$	$A_2(f)2!\binom{4}{2}(n-2)$
	4	(1^4)	$A_4(f)4!$
5	1	(5)	$A_i(f)$
	2	(4, 1)	$A_1(f)\binom{5}{1}(n-1)$
	2	(3, 2)	$A_1(f)\binom{5}{2}(n-1)$
	3	$(3, 1^2)$	$A_3(f)3!\binom{5}{3}$
	3	$(2^2, 1)$	$A_1(f)(2!)^{-1}\binom{5}{2}\binom{3}{1}(n-1)(n-2)$
	4	$(2, 1^3)$	$A_3(f)3!\binom{5}{2}(n-3)$
	5	(1^5)	$A_5(f)5!$
6	1	(6)	$A_0(f)n$
	2	(5, 1)	$A_2(f)2!\binom{6}{1}$
	2	(4, 2)	$A_0(f)\binom{6}{2}n(n-1)$
	2	(3^2)	$A_2(f)2!(2!)^{-1}\binom{6}{3}$
	3	$(4, 1^2)$	$A_2(f)2!\binom{6}{4}(n-2)$
	3	(3, 2, 1)	$A_2(f)2!\binom{6}{3}\binom{3}{2}(n-2)$
	3	(2^3)	$A_0(f)(3!)^{-1}\binom{6}{2}\binom{4}{2}n(n-1)(n-2)$
	4	$(3, 1^3)$	$A_4(f)4!\binom{6}{3}$
	4	$(2^2, 1^2)$	$A_2(f)2!(2!)^{-1}\binom{6}{2}\binom{4}{2}(n-2)(n-3)$
	5	$(2, 1^4)$	$A_4(f)4!\binom{6}{2}(n-4)$
	6	(1^6)	$A_6(f)6!$
7	1	(7)	$A_1(f)$
	2	(6, 1)	$A_1(f)\binom{7}{1}(n-1)$
	2	(5, 2)	$A_1(f)\binom{7}{2}(n-1)$
	2	(4, 3)	$A_1(f)\binom{7}{3}(n-1)$
	3	$(5, 1^2)$	$A_3(f)3!\binom{7}{2}$
	3	(4, 2, 1)	$A_1(f)\binom{7}{4}\binom{3}{2}(n-1)(n-2)$
	3	$(3^2, 1)$	$A_3(f)3!(2!)^{-1}\binom{7}{3}\binom{4}{3}$
	3	$(3, 2^2)$	$A_1(f)(2!)^{-1}\binom{7}{3}\binom{4}{2}(n-1)(n-2)$
	4	$(4, 1^3)$	$A_3(f)3!\binom{7}{4}(n-3)$
	4	$(3, 2, 1^2)$	$A_3(f)3!\binom{7}{3}\binom{4}{2}(n-3)$
	4	$(2^3, 1)$	$A_1(f)(3!)^{-1}\binom{7}{2}\binom{5}{2}\binom{3}{2}\cdot(n-1)(n-2)(n-3)$
	5	$(3, 1^4)$	$A_5(f)5!\binom{7}{3}$
	5	$(2^2, 1^3)$	$A_3(f)3!(2!)^{-1}\binom{7}{2}\binom{5}{2}(n-3)(n-4)$
	6	$(2, 1^5)$	$A_5(f)5!\binom{7}{2}(n-5)$
	7	(1^7)	$A_7(f)7!$

For any (n, k) code and any $f \in \{0, 1\}^n$

$$\sum_{i=0}^{n} A_i(f) = 2^k \tag{18a}$$

$$A_{n-i}(f) = A_i(\mathbf{1} \oplus f), \qquad \text{where } \mathbf{1} = (1, \cdots, 1). \tag{18b}$$

Thus, by (12)–(17) and (18) we may immediately obtain the weight distributions of all translates for any code with $n < 18$.

It follows from Theorem 1 and (12)–(18) that the computation of weight distributions $\{A_i(f)\}$ of translates may be reduced to the computation of $\hat{h}^i(Hf)$. The computation of $\hat{h}^i(\tau)$ for all τ and all $i = 1, \cdots, s$ by the fast Walsh transform [9], [14] requires at most $s(n-k)2^{n-k}$ additions and subtractions, $(s-1)2^{n-k}$ multiplications and 2^{n-k+1} memory cells. This method of the computations of weight distributions $\{A_i(f)\}$ of translates may be much simpler (especially for codes with small distances) than the well-known alternative based on the computation of the external distance of the code and the expansion of the corresponding annihilator polynomial in terms of Krawtchouk polynomials (see, e.g., [1, pp. 166–170]).

We note also that if an analytic expression for $h(x)$ is available, then sometimes we may find $\hat{h}^i(Hf)$ and $A_i(f)$ immediately (without application of the algorithm of the fast Walsh transform). This situation will be illustrated below by several examples. Tables of $\hat{h}(\omega)$ for a large number of classes of Boolean functions $h(x)$ may be found in [9].

Example 2: We consider the (n, k) codes with $n = 2^\alpha - 2^{\alpha-t}$, $k = 2^\alpha - 2^{\alpha-t} - \alpha$ $(t = 2, \cdots, \alpha)$ obtained by deleting from the check matrix for the $(2^\alpha - 1, 2^\alpha - \alpha - 1)$ Hamming code all columns $h_j = (h_m^{(1)}, \cdots, h_j^{(n-k)})$ for which $h_j^{(1)} = \cdots = h_j^{(t)} = 0$. Thus [8],

$$\hat{h}(\omega) = n\delta_{0,\omega} - 2^{n-k-t}\delta_{0,\omega^{(t+1)}} \cdot \delta_{0,\omega^{(t+2)}} \cdots \delta_{0,\omega^{(n-k)}}(1 - \delta_{0,\omega}), \tag{19}$$

where $\delta_{i,j}$ is the Kronecker delta. Hence

$$\begin{aligned}\hat{h}^i(\tau) &= \hat{h}^i(\tau^{(1)}, \cdots, \tau^{(n-k)}) \\ &= n^i + 2^{i(n-k-t)}(-1)^i \\ &\quad \cdot \left(2^t\delta_{0,\tau^{(1)}}\delta_{0,\tau^{(2)}} \cdots \delta_{0,\tau^{(t)}} - 1\right).\end{aligned} \tag{20}$$

Denote $\lambda = Hf$, $\lambda = (\lambda^{(1)}, \cdots, \lambda^{(n-k)})$, then by (12)–(17) and (20) we have for the case when the first t coordinates of λ are all zero

$$A_0(f) = \delta_{0,Hf}, \quad A_1(f) = 0, \tag{21}$$

$$A_2(f) = (2!)^{-1}n(1 - \delta_{0,Hf}), \tag{22}$$

$$A_3(f) = (3!)^{-1}n(n - 2^{n-k-t}), \tag{23}$$

$$A_4(f) = (4!)^{-1}\left(2^{2(n-k)} - 3n\left(2^{n-k-t} + 2 - \delta_{0,Hf}\right) - 6\delta_{0,Hf} + 8\right), \quad (24)$$

$$A_5(f) = (5!)^{-1} n(n - 2^{n-k-t})(n^2 + 2^{2(n-k-t)} - 10n + 20), \quad (25)$$

$$A_6(f) = (6!)^{-1}\Big(n\big(n^4 - n^3 2^{n-k-t} + n^2 2^{2(n-k-t)} - n2^{3(n-k-t)} + 2^{4(n-k-t)}\big) - n\left(1 - \delta_{0,Hf}\right) \cdot (45n^2 - 150n + 136) - 5!\,(3n-8)A_4(f)\Big), \quad (26)$$

$$A_7(f) = (7!)^{-1} n(n - 2^{n-k-t})\big(n^4 - 21n^3 + 175n^2 - 630n + 784 + 2^{2(n-k-t)} \cdot (n^2 - 21n + 70 - 2^{2(n-k-t)})\big). \quad (27)$$

For the case when there are nonzero coordinates among the first t coordinates of λ we have by (12)–(17) and (20)

$$A_0(f) = 0, \quad A_1(f) = 1,$$

$$A_2(f) = (2!)(n - 2^{n-k-t}), \quad (28)$$

$$A_3(f) = (3!)^{-1}(n^2 - n2^{n-k-t} + 2^{2(n-k-t)} - 3n + 2) \quad (29)$$

$$A_4(f) = (4!)^{-1}(n - 2^{n-k-t})(n^2 + 2^{2(n-k-t)} - 6n + 8), \quad (30)$$

$$A_5(f) = (5!)^{-1}\Big(n^4 - n^3 2^{n-k-t} + n^2 2^{2(n-k-t)} - n2^{3(n-k-t)} + 2^{4(n-k-t)} + 10\,2^{n-k-t} \cdot (n - 2^{n-k-t})(n-2) - 10n^3 + 35n^2 - 50n + 24\Big), \quad (31)$$

$$A_6(f) = (6!)^{-1}(n - 2^{n-k-t})\big(n^4 + n^2 2^{2(n-k-t)} + 2^{4(n-k-t)} - 5(3n-8)2^{2(n-k-t)} - 15n^3 + 85n^2 - 210n + 174\big), \quad (32)$$

$$A_7(f) = (7!)^{-1}\Big((n^7 + 2^{7(n-k-t)})(n + 2^{n-k-t})^{-1} - 1 - 7(n-1)(9 + 15(n-1)(n-2)) - A_3(f)3!7(13 + 5(n-3)(3n-5)) - A_5(f)5!7(3n-10)\Big). \quad (33)$$

Formulas for $A_n(f), A_{n-1}(f), \cdots, A_{n-7}(f)$ for these codes may be derived by the substitution of $\mathbf{1} \oplus f$ instead of f in (21)–(33). We also note that formulas (21)–(26) are generalizations of the corresponding results from [8]. We shall illustrate now the application of Corollary 2 to weight distributions of a code V and its dual $V^{\perp}$.

Example 3: Consider the weight distributions of the $(2k+1, k)$ codes with the check matrices

$$H = \left(\begin{array}{c|c} I_{k+1} & \begin{array}{c} I_k \\ \hline 1\,1\cdots 1 \end{array} \end{array}\right),$$

where I_m is the $(m \times m)$ identity matrix. Then

$$h(x) = h(x^{(1)}, \cdots, x^{(k+1)}) = h_1(x^{(1)}, \cdots, x^{(k+1)}) + h_2(x^{(1)}, \cdots, x^{(k+1)}),$$

where

$$h_1(x^{(1)}, \cdots, x^{(k+1)}) = \begin{cases} 1, & \text{if } \sum_{i=1}^{k} x^{(i)} = 1; \\ 0, & \text{otherwise;} \end{cases}$$

$$h_2(x^{(1)}, \cdots, x^{(k+1)}) = \begin{cases} 1, & \text{if } x^{(1)} = \cdots = x^{(k)} = 0, \\ & x^{(k+1)} = 1; \\ 0, & \text{otherwise.} \end{cases} \quad (34)$$

Thus, we have [9]

$$\hat{h}_1(\omega) = \hat{h}_1(\omega^{(1)}, \cdots, \omega^{(k+1)}) = \delta_{0,\omega^{(k+1)}} 2(k - 2wt(\omega)),$$

$$\hat{h}_2(\omega) = (-1)^{\omega^{(k+1)}},$$

$$\hat{h}(\omega) = \hat{h}_1(\omega) + \hat{h}_2(\omega) = \delta_{0,\omega^{(k+1)}} 2(k + 1 - 2wt(\omega)) - 1, \quad (35)$$

and

$$\hat{\hat{h}}^i(0) = \sum_{\omega} \hat{h}^i(\omega) = \sum_{j=0}^{k} \binom{k}{j} (2k + 1 - 4j)^i + (-1)^i 2^k. \quad (36)$$

From (36) and (12)–(17) with $f = 0$ we finally have for the weight distributions of these codes

$$A_0(0) = 1, \quad A_1(0) = A_2(0) = 0,$$

$$A_3(0) = (3!)^{-1}\left(\sum_{j=0}^{k} 2^{-(k+1)} \binom{k}{j} \cdot (2k + 1 - 4j)^3 - 1/2\right), \quad (37)$$

$$A_4(0) = (4!)^{-1}\left(\sum_{j=0}^{k} 2^{-(k+1)} \binom{k}{j} (2k + 1 - 4j)^4 + 1/2 - (2k+1)(6k+1)\right), \quad (38)$$

$$A_5(0) = (5!)^{-1}\left(\sum_{j=0}^{k} 2^{-(k+1)} \binom{k}{j} (2k + 1 - 4j)^5 - 1/2 - A_3(0)3!10(2k-1)\right), \quad (39)$$

$$A_6(0) = (6!)^{-1}\left(\sum_{j=0}^{k} 2^{-(k+1)} \binom{k}{j} (2k + 1 - 4j)^6 + 1/2 - A_4(0)4!5(6k-5) - (2k+1)(60k^2+1)\right), \quad (40)$$

$$A_7(0) = (7!)^{-1}\left(\sum_{j=0}^{k} 2^{-(k+1)} \binom{k}{j} (2k+1-4j)^7 - 1/2 - A_5(0)5!7(6k-7) - A_3(0) \cdot 3!7(13+5(2k-2)(6k-2)) \right). \quad (41)$$

We note also that the Walsh transform $\hat{h}(\omega)$ may be used for the computation of weight distributions for dual codes, since [8]

$$wt(\omega H) = (1/2)\big(n - \hat{h}(\omega)\big), \quad (42)$$

where ωH is the vector of length n obtained by multiplying the check matrix H by the row vector ω.

Thus we have, for example, from (42) and (19) for the weight distributions $B_j(0)$ of $(2^\alpha - 2^{\alpha-t}, \alpha)$ codes which are dual to the codes from Example 2

$$B_j(0) = \delta_{0,j} + (2^{\alpha-t}-1)\delta_{2^{\alpha-1},j} + (2^\alpha - 2^{\alpha-t})\delta_{2^{\alpha-1}-2^{\alpha-t-1},j}. \quad (43)$$

For the $(2k+1, k+1)$ codes, which are dual to the codes from Example 3, we have from (42) and (35) for even k

$$B_j(0) = \begin{cases} \binom{k}{i}, & \text{if } j = 2i, \quad i = 0,\cdots,k; \\ 2^k, & \text{if } j = k+1; \\ 0, & \text{otherwise;} \end{cases} \quad (44)$$

and for odd k

$$B_j(0) = \begin{cases} \binom{k}{i}, & \text{if } j = 2i, \quad i = 0,\cdots,k, \\ & i \neq (1/2)(k+1); \\ 2^k + \binom{k}{(1/2)(k+1)}, & \text{if } j = k+1; \\ 0, & \text{otherwise.} \end{cases} \quad (45)$$

III. Covering Radius of Linear Codes

For the (n, k) code V the covering radius (the true external distance) $t(V)$ is

$$t(V) = \max_{f \in \{0,1\}^n} \min_{v \in V} \operatorname{dist}(v, f)$$

Calculating the covering radius $t(V)$ for the given code V is an important and difficult problem (see, e.g., [1]). We describe in this section one approach to the solution of this problem, which will require for the computation of $t(V)$ for $n \to \infty$ at most $2^{n-k}(n-k)\log_2(n-k)$ arithmetical operations. The exact complexity estimations will be also given in this section. This solution is based on the following theorem.

Theorem 2: Let V be the binary (n, k) code with check matrix $H = (h_1,\cdots,h_n)$ and

$$g_i(\omega) = \sum_{j=1}^{i} \hat{h}^j(\omega), \qquad i = 1,\cdots,n. \quad (46)$$

Then the covering radius $t(V)$ of the code V is equal to the minimal t such that $\hat{g}_t(\tau) \neq 0$ for all $\tau \neq 0$.

Proof: It follows from (1), (2) that

$$\hat{\hat{h}}^i = 2^{n-k} \sum_{x_i \oplus \cdots \oplus x_i = \tau,} h(x_1)\cdots h(x_i). \quad (47)$$

If for the given $\tau \neq 0$ $\hat{\hat{h}}^i(\tau) \neq 0$, then for any f such that $Hf = \tau$ there exists $u \in V \oplus f$ with $wt(u) \leq i$. But, if $u \in V \oplus f$ and $wt(u) \leq i$, then there exists $v \in V$ such that $u = v \oplus f$ and $\operatorname{dist}(v, f) \leq i$. Since for every $\tau \neq 0$ there exists $f \in V$ such that $Hf = \tau$, the covering radius of V is the minimal t such that

$$\sum_{i=1}^{t} \hat{\hat{h}}^i(\tau) \neq 0$$

for all $\tau \neq 0$. Using the definition of $g_t(\omega)$ and linearity of the Walsh transform we have

$$\sum_{i=1}^{t} \hat{\hat{h}}^i(\tau) = \hat{g}_t(\tau). \quad (48)$$

This completes the proof of Theorem 2.

Theorem 2 provides us with simple algorithms for the computation of the covering radius $t(V)$ for the code V. For small $t(V)$ the following algorithm may be used for the code V with the check matrix $H = (h_1,\cdots,h_n)$.

1) Compute $\Delta_1 = \{0,1\}^{n-k} - \{0, h_1,\cdots,h_n\}$. If $\Delta_1 = \varnothing$ (empty set), then $t(V) = 1$.
2) Compute $\hat{h}(\omega)$ by the fast Walsh transform, then $\hat{\hat{h}}^2(\omega)$, $\hat{\hat{h}}^2(\tau)$, and $\Delta_2 = \Delta_1 - \{\tau | \hat{\hat{h}}^2(\tau) \neq 0\}$. If $\Delta_2 = \varnothing$, then $t(V) = 2$.

⋮

t) Using the fast Walsh transform, compute $\hat{\hat{h}}^t(\tau)$ from the previously computed $\hat{h}(\omega)$ and $\hat{\hat{h}}^{t-1}(\omega)$; then compute $\Delta_t = \Delta_{t-1} - \{\tau | \hat{\hat{h}}^t(\tau) \neq 0\}$. If $\Delta_t = \varnothing$, then $t(V) = t$. If $\Delta_t \neq \varnothing$, go to $t+1$).

The algorithm requires at most $t(V)(n-k)2^{n-k}$ additions and subtractions, $(t(V)-1)2^{n-k}$ multiplications and $3 \cdot 2^{n-k}$ memory cells. We shall describe in the proof of Corollary 3 another algorithm which is more efficient in the case of big $t(V)$ or in the case when no good upper bound for $t(V)$ is available.

Denote by $V(n, k)$ the set of all linear (n, k) codes with distance > 2. For the given $V \in V(n, k)$ let $L(V)$ be the minimal number of arithmetical operations for the computation of $t(V)$ by any algorithm and

$$L(n, k) = \max_{V \in V(n,k)} L(V). \quad (49)$$

Let us construct now an asymptotical $(n \to \infty)$ upper bound for $L(n, k)$.

Corollary 3: For $n \to \infty$

$$L(n,k) \lesssim 2^{n-k}(n-k)\log_2(n-k)$$

$$\left(a(n,k) \lesssim b(n,k) \quad \text{iff} \quad \overline{\lim_{n\to\infty}}\, a(n,k)(b(n,k))^{-1} \le 1\right). \tag{50}$$

Proof: We have by (46)

$$g_t(\omega) = \begin{cases} t\hat{h}(\omega), & \text{if } \hat{h}(\omega) \in \{0,1\}; \\ \left(\hat{h}^{t+1}(\omega) - \hat{h}(\omega)\right)\left(\hat{h}(\omega) - 1\right)^{-1}, & \text{otherwise.} \end{cases} \tag{51}$$

Let us first compute $\hat{h}(\omega)$, $\hat{h}^2(\omega)$, $\hat{h}^4(\omega),\cdots,\hat{h}^{2^{m-1}}(\omega)$, where m is the smallest integer such that $n-k \le 2^m$. These computations require $(n-k)2^{n-k}$ additions and subtractions and $(m-2)2^{n-k}$ multiplications. After this we compute $g_{t_1}(\omega)$ for $t_1 = 2^{m-1}-1$ by (51) and $\hat{g}_{t_1}(\tau)$ by the fast Walsh transform. If $\hat{g}_{t_1}(\tau) \neq 0$ for all $\tau \neq 0$, then we compute $g_{t_2}(w)$ and $\hat{g}_{t_2}(\tau)$ for $t_2 = 2^{m-2}-1$; if $\hat{g}_{t_1}(\tau) = 0$ for some $\tau \neq 0$, then we compute $g_{t_2}(\omega)$ and $\hat{g}_{t_2}(\tau)$ for $t_2 = 2^{m-1}+2^{m-2}-1$, etc.

This algorithm requires $(n-k)2^{n-k} + m(n-k)2^{n-k}$ additions and subtractions, $2(m-1)2^{n-k}$ multiplications, and $m2^{n-k}$ memory cells. Since $t(V) \le n-k$ and $m \sim \log_2(n-k)$ ($a(n) \sim b(n)$ iff $a(n) \lesssim b(n)$ and $a(n) \gtrsim b(n)$), we finally have for $n \to \infty$

$$\begin{aligned} L(n,k) &\lesssim (n-k)(\log_2 n + 1)2^{n-k} \\ &\quad + 2\log_2(n-k)2^{n-k} \\ &\sim 2^{n-k}(n-k)\log_2(n-k). \end{aligned} \tag{52}$$

For many cases an upper bound $S(V)$ for $t(V)$ is known (for example, we may choose as $S(V)$ the external distance of the code V [1].) Let s be the minimal integer such that $S(V) \le 2^s$. Then using the method described in the proof of Corollary 3, we always can find the covering radius by at most $(s+1)(n-k)2^{n-k}$ additions and subtractions, $2(s-1)2^{n-k}$ multiplications and $s2^{n-k}$ memory cells. We note also that, if we have an analytical expression for the characteristic function $h(x)$, then sometimes we may find $t(V)$ immediately from $h(x)$ (see Example 4 later in this section).

Let us consider codes with small covering radii. For the code V with $\operatorname{dist}(V) > 2$, $t(V) = 1$ iff V is a Hamming code. Since $\hat{h}(\tau) = 2^{n-k}h(\tau)$, we have by Theorem 2, that $t(V) = 2$ iff $\hat{\hat{h}}^2(\tau) \neq 0$ for every $\tau \notin \{0, h_1,\cdots,h_n\}$. By the Wiener–Khinchin theorem for the Walsh transform [9], we have

$$\begin{aligned} \hat{\hat{h}}^2(\tau) &= 2^{n-k}B(\tau) \\ &= 2^{n-k} \sum_{x\in\{0,1\}^{n-k}} h(x)h(x\oplus\tau). \end{aligned} \tag{53}$$

The function $B(\tau)$, known as the logical autocorrelation function, is widely used in logical design, fault-tolerant computing, digital filtering, signal processing, etc. (see, e.g., [9], [15]–[17]). Thus, the code V with a check matrix $H = (h_1,\cdots,h_n)$ has the covering radius $t(V) = 2$ iff the characteristic function $h(x)$ for the set $\{h_1,\cdots,h_n\}$ has nonzero autocorrelation $B(\tau)$ for every $\tau \neq \{0, h_1,\cdots,h_n\}$. Tables of autocorrelation functions $B(\tau)$ for a large number of classes of Boolean functions $h(x)$ may be found in [9].

Example 4: Consider the (n,k) codes V with $n = 2^{\alpha-1}(2^\alpha - 1)$, $k = 2^{\alpha-1}(2^\alpha-1) - 2\alpha$ and $\operatorname{dist}(V) = 3$, generated by "nonrepetitive quadratic forms over GF(2)" through

$$h(x^{(1)},\cdots,x^{(n-k)}) = \bigoplus_{i,j}^{2\alpha} x^{(i)}x^{(j)}, \tag{54}$$

in which each of the arguments $x^{(s)}$ ($s = 1,\cdots,2\alpha$) appears exactly once [8], [9]. The weight distributions $\{A_i(0)\}$ for these codes for $i < 7$ are given in [8]. For the nonrepetitive quadratic form (54), we have [9]

$$\begin{aligned} B(\tau) &= 2^{-(n-k)}\hat{\hat{h}}^2(\tau) \\ &= 2^{2\alpha-2} - 2^{\alpha-1}, \qquad \text{for all } \tau \neq 0, \end{aligned} \tag{55}$$

and by Theorem 2, $t(V) = 2$ for every $\alpha \ge 2$ (we note that for these codes the stronger statement is valid, that for any (n,k) code V generated by the quadratic form (54) and any $f \notin V$ there exist $v_1,\cdots,v_r \in V$ such that $\operatorname{dist}(v_i, f) = 2$ $(i = 1,\cdots,r)$ where $r = 2^{2\alpha-3} - 2^{\alpha-2}$ (see Corollary 4 below). The "nonrepetitive quadratic forms" (54) are a special case of "bent functions" [1], [29], and the following corollary generalizes the results of Example 4.

Corollary 4: If for an (n,k) code $Vh(x)$ is a "bent function," then $t(V) = 2$, and for any $f \notin V$ there exist $v_1,\cdots,v_r \in V$ such that $\operatorname{dist}(v_i, f) = 2$ $(v_i \neq v_j;\ i,j = 1,\cdots,r;\ i \neq j)$, where

$$r = n/2 - 2^{n-k-3}. \tag{56}$$

Proof: If $h(x)$ is "bent" then [1], $n-k$ is even, $n-k = 2\alpha$, and $|\hat{H}(\omega)| = 2^\alpha$ where $H(x) = 1 - 2h(x)$. Thus,

$$\hat{h}(0) = n = 2^{\alpha-1}(2^\alpha \pm 1), \quad \hat{h}^2(0) = 2^{2(\alpha-1)}(2^\alpha \pm 1)^2,$$

and for any $\omega \neq 0$

$$|\hat{h}(\omega)| = 2^{\alpha-1}, \quad \hat{h}^2(\omega) = 2^{2(\alpha-1)}. \tag{57}$$

By (1), (2) and (53) we now have for any $\tau \neq 0$

$$\begin{aligned} B(\tau) &= 2^{-2\alpha}\hat{\hat{h}}^2(\tau) = 2^{-2}\left((2^\alpha \pm 1)^2 - 1\right) \\ &= 2^{\alpha-1}(2^{\alpha-1} \pm 1) = n - 2^{2\alpha-2}, \end{aligned} \tag{58}$$

and by Theorem 2 $t(V) = 2$. It follows from (53) that for any $f \notin V$ such that $Hf = \tau$ there exist $r = (1/2)B(\tau)$ vectors $u_1,\cdots,u_r \in V \oplus f$ with $wt(u_i) = 2$ $(i = 1,\cdots,r)$. Corollary 4 follows immediately from (58) with $v_i = u_i \oplus f$ $(i = 1,\cdots,r)$.

We note that if $h(x)$ is "bent," then formulas for A(2), A(4), and A(6) for these codes may be easily derived from $\hat{h}(0) = 2^{\alpha-1}(2^\alpha \pm 1)$, $|\hat{h}(\omega)| = 2^{\alpha-1}$ $(\omega \neq 0)$ by (12), (14), and (16).

TABLE II
THE PARAMETERS OF SOME BINARY LINEAR PERFECT CODES

Number	Code V	Parameters (n, k, dist)	Set L of Weights of Errors
1	$\{0^{2s+1}, 1^{2s+1}\}$	$(2s+1, 1, 2s+1)$	$\{l_0, l_1, \cdots, l_s \mid l_i \in \{i, 2s+1-i\};\ i = 0, 1, \cdots, s\}$
2	$\{0^n, 0^{n-2s-1}1^{2s+1}\}$ $(0 \le s \le (i/2)(n-1))$	$(n, 1, 2s+1)$	$\{0, 2, 4, \cdots, 2[(1/2)n]\}$
3	$\{0^{4s+2}, 0^{2i+1}1^{2j+1}, 1^{2i+1}0^{2j+1}, 1^{4s+2}\}$ $(i+j=2s)$	$(4s+2, 2, \min\{2i+1, 2j+1\})$	$\{l_0, l_1, \cdots, l_s \mid l_i \in \{2i, 4s+2-2i\};\ i = 0, 1, \cdots, s\}$
4	Hamming codes C_α	$(2^\alpha - 1, 2^\alpha - \alpha - 1, 3)$	$\{l_0, l_1 \mid l_0 \in \{0, 2^\alpha - 1\}, l_1 \in \{1, 2^\alpha - 2\}\}$
5	$\{v \mid v \in C_\alpha, wt(v) \text{ is even}\}$ (see Number 4)	$(2^\alpha - 1, 2^\alpha - \alpha - 2, 4)$	$\{0, 1, 2^\alpha - 2, 2^\alpha - 1\}$
6	$\{(v^{(1)}, \cdots, v^{(n-1)}) \mid v^{(n)} \in \{0, 1\},$ $(v^{(1)}, \cdots, v^{(n)}) \in C_\alpha, wt(v) \text{ is even}\}$ $(n = 2^\alpha - 1$; see Number 4)	$(2^\alpha - 2, 2^\alpha - \alpha - 2, 3)$	$\{0, l, 2^\alpha - 2 \mid l \in \{1, 2^\alpha - 3\}\}$
7	Golay code G	$(23, 12, 7)$	$\{l_0, l_1, l_2, l_3 \mid l_0 \in \{0, 23\}, l_1 \in \{1, 22\}, l_2 \in \{2, 21\},$ $l_3 \in \{3, 19\}\}$
8	$\{v \mid v \in G, wt(v) \text{ is even}\}$ (see Number 7)	$(23, 11, 8)$	$\{0, 1, 2, 3, 20, 21, 22, 23\}$
9	$\{(v^{(1)}, \cdots, v^{(22)}) \mid v^{(23)} \in \{0, 1\},$ $(v^{(1)}, \cdots, v^{(23)}) \in G, wt(v) \text{ is even}\}$ (see Number 7) [27]	$(22, 11, 7)$	$\{0, 1, 2, l, 20, 21, 22 \mid l \in \{3, 19\}\}$
10	$\{v \mid wt(v) \text{ is even}\}$	$(2s+1, 2s, 2)$	$(0, 2s+1\}$
11	$\{(v^{(1)}, \cdots, v^{(n)}) \mid v^{(1)} = 0\}$	$(n, n-1, 1)$	$\{0, n\}$

It was pointed out by H. F. Mattson, Jr. [23] that the codes from Example 4 and Corollary 4 are special cases of the structure codes considered in [24]–[26]. The codes from Example 4 and Corollary 4 are structure codes corresponding to coset leaders of the first-order Reed–Muller codes RM$(1, 2\alpha)$ with the weight of coset leaders equal to the covering radius $t(\text{RM}(1, 2\alpha)) = 2^{2\alpha-1} - 2^{\alpha-1}$ of RM$(1, 2\alpha)$.

If for an (n, k) code V with a check matrix $H = (h_1, \cdots, h_n)$ $h(x)$ is a "bent" function, then for any code V' with a check matrix H', obtained by addition to H of at most $2^{n-k} - n - 1$ different nonzero columns or by deletion from H of at most $n/2 - 2^{n-k-3} - 1$ columns, we have $t(V') = 2$. Similarly, if $V^{(T)}$ is a check-translation of V (see (7) in Section II), then $t(V^{(T)}) = 2$. For example, for the $(2^{\alpha-1}(2^\alpha - 1), 2^{\alpha-1}(2^\alpha - 1) - 2\alpha)$ codes V with $h(x^{(1)}, \cdots, x^{(2\alpha)}) = \oplus_{i,j}^{2\alpha}(x^{(i)} v x^{(j)})$ where each of the arguments appears exactly once, α is even, and v stands for logical summation ($x^{(i)} v x^{(j)} = 1$ unless both $x^{(i)}$ and $x^{(j)}$ are zero, we have from (54) with $T = (1, \cdots, 1)$ that $t(V) = 2$. We note also that Theorem 2 may be used for the computation of the radius $\rho(V)$ of the given code V. The *radius* $\rho(V)$ of V is defined as [1]

$$\rho(V) = \min_{f \in \{0,1\}^n} \max_{v \in V} \text{dist}(v, f) \tag{59}$$

and, from (59), $\rho(V) = n - t(V)$, hence $\rho(V) \ge k$.

IV. PERFECT CODES CORRECTING ERRORS OF GIVEN WEIGHTS

Let $L = \{0, l_1, \cdots, l_s\}$ and $0 < l_1 < \cdots < l_s \le n$. As before, we say that the (n, k) code V corrects errors of weights $l_1, \cdots, l_s$ iff for any $v, v' \in V$ $(v \ne v')$ and any e, e' such that $wt(e), wt(e') \in L$ $v \oplus e \ne v' \oplus e'$. The (n, k) code V is L-perfect iff it corrects errors of weights $l_1, \cdots, l_s$ and satisfies the Hamming bound (see Section I)

$$\sum_{l \in L} \binom{n}{l} = 2^{n-k}. \tag{60}$$

Note that if V is an L-perfect (n, k) code for $L = \{0, l_1, \cdots, l_s\}$, $M \subseteq L - 0$ and $\mathbf{1} = (1, \cdots, 1) \in V$, then V is also L-perfect for any $L' = (L - M) \cup_{\alpha \in M} \{n - \alpha\}$. Thus, there is no L-perfect code containing $\mathbf{1}$ if for some l, both l and $n - l$ are in L.

We note also that if V is L-perfect, then any translate $V \oplus f$ of V is also L-perfect for the same L. Several examples of simple L-perfect binary linear codes are given in Table II (where we use the notation:

$$0^i 1^j = (\underbrace{00 \cdots 0}_{i} \ \underbrace{11 \cdots 1}_{j})).$$

We note that for any perfect (n, k) code V from Table II an extended $(n+1, k+1)$ code $V \times \{0, 1\}$ with distance 1, obtained by adding a last component 0 or 1 to any $v \in V$, is also perfect for some L such that $0 \notin L$; and, if $l \in L$, then l is odd. (The L-perfect codes $V \times \{0, 1\}$ where V are repetition codes, Hamming codes, or the Golay code, are described in [19].) Some nonlinear binary perfect codes will be discussed later in this section.

Now, let us describe necessary and sufficient conditions for a code to be L-perfect. For $L = \{0, l_1, \cdots, l_s\}$ denote

$$\alpha_L(Hf) = 1 + \sum_{l \in L} (l!)^{-1} C_l(f), \tag{61}$$

where $C_l(f)$ was defined in Section II (formulas for $C_i(f)$ for $i = 0, \cdots, 7, n-7, n-6, \cdots, n$ may be derived immediately from (12)–(18)).

Theorem 3: An (n, k) code V with a check matrix $H = (h_1, \cdots, h_n)$ is L-perfect iff

$$\sum_{l \in L} (l!)^{-1} \hat{h}^l(\omega) = \hat{\alpha}_L(\omega), \qquad \text{for all } \omega \in \{0, 1\}^{n-k}. \tag{62}$$

Proof: A code V is L-perfect iff

$$\sum_{l \in L} A_l(f) = 1 \qquad \text{for all } f. \tag{63}$$

By Theorem 1 and (61) we see that (63) is equivalent to

$$\sum_{l\in L}(l!)^{-1}\hat{\hat{h}}^{l}(\tau)=2^{n-k}\alpha_L(\tau)\qquad\text{for all }\tau. \tag{64}$$

Since $\hat{\hat{a}}(x)=2^{n-k}a(x)$ for any Boolean function of $n-k$ arguments, we have (62) immediately from (64) using the linearity of the Walsh transform.

Example 5: Let us construct perfect codes correcting errors in any one component and errors in all components in a message. (These types of errors may appear, for example, as a result of physical failures in a generator of messages if a generator has the hierarchical structure of Fig. 1b; see Example 1.) Thus, $L=\{0,1,n\}$, and we have by (5) and (18)

$$\begin{aligned}\sum_{l\in L}(l!)^{-1}\hat{\hat{h}}^{l}(Hf)&-2^{n-k}(n!)^{-1}C_n(f)\\&=\hat{\hat{h}}^{0}(Hf)+(n!)^{-1}\hat{\hat{h}}^{n}(Hf)\\&\quad-2^{n-k}(n!)^{-1}C_n(f)+\hat{\hat{h}}(Hf)\\&=2^{n-k}\big(\delta_{0,Hf}+h(Hf)+A_n(f)\big)\\&=2^{n-k}\big(\delta_{0,Hf}+h(Hf)+A_0(\mathbf{1}\oplus f)\big)\\&=2^{n-k}\big(\delta_{0,Hf}+h(Hf)+\delta_{0,H(\mathbf{1}\oplus f)}\big).\end{aligned} \tag{65}$$

From (61), (63), (65) and $C_0(f)=C_1(f)=0$ for all f (see (12)) we have the following necessary and sufficient condition for an (n,k) code to be a perfect code correcting errors of weights one and n:

$$\delta_{0,\tau}+h(\tau)+\delta_{0,\tau\oplus H\mathbf{1}}=1\qquad\text{for all }\tau. \tag{66}$$

Let

$$h(\tau)=\begin{cases}1, & \tau\notin\{(0,\cdots,0),\mathbf{1}=(1,\cdots,1)\};\\ 0, & \text{otherwise.}\end{cases} \tag{67}$$

Then, by definition (67) of $h(\tau)$, we have $H\mathbf{1}=\mathbf{1}$, and (66) is satisfied. Thus, the $(2^\alpha-2,2^\alpha-\alpha-2)$ codes defined by (67) are L-perfect for $L=\{0,1,2^\alpha-2\}$. Similarly we can show that these codes are also L-perfect for $L=\{0,2^\alpha-3,2^\alpha-2\}$. Note that the condition (62) of Theorem 3 (or the equivalent condition (64)) provides us with simple checks to verify whether the given code V is L-perfect with respect to the given set L.

We note also that very few perfect codes may exist. We have seen that perfect (n,k) codes with $k<3$ exist for $L=\{0,2,4,\cdots,2s\}$ (see Table II, numbers 2 and 3). K. A. Post [18] pointed out that there is no perfect code for $L=\{0,2,4,\cdots,2s\}$ with $k\geq 3$. Indeed, if V is a perfect (n,k) code for $L=\{0,2,4,\cdots,2s\}$, denote $V_0=\{v\in V\,|\,wt(v)\text{ is even}\}$ and $V_0'=\{(v^{(1)},\cdots,v^{(n-1)})\,|\,v^{(n)}\in\{0,1\}(v^{(1)},\cdots,v^{(n)})\in V_0\}$. If $|V_0|>1$, then $\operatorname{dist}(V_0')\geq 4s+1$ and, since

$$2^{n-k}=\sum_{i=0}^{s}\binom{n}{2i}=\sum_{i=0}^{2s}\binom{n-1}{i},$$

V_0' is a perfect code for $L'=\{0,1,2,\cdots,2s\}$. Thus [1], $V_0'=\{0^{4s+1},1^{4s+1}\}$, $V_0=\{0^{4s+2},1^{4s+2}\}$, and $k<3$. We shall prove now the analog of the well-known Lloyd's theorem [1], which will provide us with a strong necessary condition for any (not necessarily linear) code to be L-perfect.

Theorem 4: Let $V\subset\{0,1\}^n$, $V^\perp=\{\omega\,|\,(x\cdot\omega)=0,\ x\in V\}$, $\{B_i(0)\}$ the weight distribution for $V^\perp$, and $I=\{i\,|\,B_i(0)\neq 0,\ i\neq 0\}$. Denote $L(x)=\sum_{i\in L}P_i(x)$, where

$$P_i(x)=\sum_{s=0}^{i}(-1)^s\binom{x}{s}\binom{n-x}{i-s}$$

is the Krawtchouk polynomial [1]. If V is L-perfect, then for every $i\in I$ we have $L(i)=0$.

Proof: Denote

$$\begin{aligned}v(x)&=\begin{cases}1, & x\in V,\\ 0, & x\notin V,\end{cases}\\ l_i(x)&=l_i(x^{(1)},\cdots,x^{(n)})\\&=\begin{cases}1, & wt(x)=i,\\ 0, & wt(x)\neq i,\end{cases}\end{aligned}$$

$l(x)=\sum_{i\in L}l_i(x)$. If V is L-perfect, then for any $f\in\{0,1\}^n$

$$\sum_{x\in\{0,1\}^n}v(x)l(f\oplus x)=1. \tag{68}$$

Using the convolution theorem for the Walsh transform [9], we have from (68)

$$\hat{v}(\omega)\hat{l}(\omega)=0,\qquad\text{for all }\omega\neq 0. \tag{69}$$

For $\hat{v}(\omega)$ we have [9] for $\omega\neq 0$

$$\hat{v}(\omega)\neq 0,\qquad\text{iff }\omega\in V^\perp\text{ or }wt(\omega)\in I. \tag{70}$$

For the elementary symmetrical Boolean function $l_i(x)$, we have [9]

$$\hat{l}_i(\omega)=P_i(wt(\omega)). \tag{71}$$

Theorem 4 follows from (69)–(71). It may be shown by Theorem 4 that the only perfect codes correcting exactly two errors ($L=\{0,2\}$) are (2, 1), (3, 1), and (6, 2) codes (described in Table II), and there is no perfect code correcting exactly three errors ($L=\{0,3\}$) and having a distance of more than 2.

Moreover, it was shown in [19] that for $L\subset\{0,1,\cdots,[n/2]\}$ nontrivial L-perfect codes exist only for the following parameters (n,k,L): $(2^\alpha-1,2^\alpha-\alpha-1,\{0,1\})$, $(2s+1,1,\{0,1,\cdots,s\})$, $(23,12,\{0,1,2,3\})$, $(2^\alpha,2^\alpha-\alpha,\{1\})$, $(2s+2,2,\{s,s-2,\cdots,\epsilon\})$ (where $s=2\alpha+\epsilon$, $\epsilon\in\{0,1\}$) and $(24,13,\{1,3\})$. We note, however, that for the given parameters (n,k,L) several nonequivalent perfect codes with different distances may exist. To illustrate this, we shall give now the example of *nonlinear* codes correcting errors of weights 1 and n (perfect linear codes for $L=\{0,1,n\}$ were constructed in Example 6). For any binary vector u denote $\pi(u)$ is zero if $wt(u)$ is even and 1 if odd.

Theorem 5: Let V be a perfect (m, t) single error correcting code (not necessarily linear) with $m = 2^s - s - 1$, $s \geq 3$ and for any $v \in V$

$$\lambda(v) = \begin{cases} 1, & v = 0, \\ 0, & v \neq 0. \end{cases} \tag{72}$$

Then

$$C = \Big\{u|v^{(1)}|u \oplus (v^{(2)},\cdots,v^{(m)})|\pi(u) \oplus \lambda(v)|: u \in \{0,1\}^{m-1}, (v^{(1)}, v^{(2)},\cdots,v^{(m)}) \in V\Big\} \tag{73}$$

is a perfect $(2^{s+1} - 2, 2^{s+1} - s - 3)$ code correcting errors of weights one and $2^{s+1} - 2$. This code is not equivalent to any linear code.

Proof: Denote

$$C_V = \big\{u|u \oplus v|\pi(u) \oplus \lambda(v)|: u \in \{0,1\}^m, v \in V\big\}. \tag{74}$$

Then $\mathbf{1} \in C_V$, and for any $x \in C_V$, $\bar{x} = \mathbf{1} \oplus x \in C_V$. For any $x = (x^{(1)},\cdots,x^{(2m)})$ denote $0x = (0, x^{(1)},\cdots,x^{(2m)})$. We note that from (73) and (74) $0x \in C_V$ Iff $x \in C$. Let us prove now that for any $y \in \{0,1\}^{2m}$ there exists unique $x \in C$ such that $\text{dist}(x, y) \in \{0, 1, 2m\}$. Since C_V is the Vasiliev's perfect single error correcting code [20], for any $y \in \{0,1\}^{2m}$ there exists unique $x^{(0)}x = (x^{(0)}, x^{(1)},\cdots,x^{(2m)}) \in C_V$ such that

$$\text{dist}\,(0y, x^{(0)}x) \leq 1. \tag{75}$$

We shall consider the two cases $x^{(0)} = 0$ and $x^{(0)} = 1$. For $x^{(0)} = 0$, since $0x \in C_V$, $x \in C$ and $\text{dist}(x, y) \leq 1$. If there exists $z \in C$ $(z \neq x)$ such that $\text{dist}(y, z) \leq 1$, then $\text{dist}(0y, 0z) \leq 1$, $0z \in C_V$, and we have a contradiction. Similarly, if $z \in C$ and $\text{dist}(y, z) = 2m$, then $\text{dist}(0y, 1\bar{z}) = 1$, $0z \in C_V$, $1\bar{z} \in C_V$, and again we have a contradiction.

For $x^{(0)} = 1$ from (75), $y = x$, $1x \in C_V$, $0\bar{x} \in C_V$, $\bar{x} \in C$, $\text{dist}(y, \bar{x}) = 2m$. It is easy to check that for $x^{(0)} = 1$ there is no $z \in C$ $(z \neq x)$ such that $\text{dist}(y, z) \leq 1$.

Since C_V is not equivalent to any linear code [20], there exists $x^{(0)}x$, $y^{(0)}y \in C_V$ such that $x^{(0)}x \oplus y^{(0)}y \notin C_V$ and $x^{(0)} = y^{(0)} = 0$ (if, e.g., $x^{(0)} = 1$, we have to consider $0\bar{x}$ instead of $1x$). Then x, $y \in C$, and $x \oplus y \notin C$. This completes the proof of Theorem 5.

We note that by extending of perfect $(2^{s+1} - 2, 2^{s+1} - s - 3)$ codes C defined by (73) (adding to any $v \in C$ a last component 0 or 1) we obtain the nonlinear L-perfect $(2^{s+1} - 1, 2^{s+1} - s - 2)$ codes for $L = \{1, 2^{s+1} - 1\}$.

V. Generalization to Nonbinary Codes

Most of the results of previous sections may be easily generalized to the case of linear codes over GF(q) (q is a prime). To this end, we need only make two changes in the basic definitions.

First, we replace the check matrix $H = (h_1,\cdots,h_n)$ by the "extended check matrix" with columns $h_1, 2h_1,\cdots,(q-1)h_1,\cdots,h_n, 2h_n,\cdots,(q-1)h_n$ (all the multiplications are carried out in GF(q)). Second, we replace $(-1)^{x\cdot\omega}$ by $\xi^{x\cdot\omega}$ in (1) where ξ is a primitive complex qth root of 1. This generalization is very similar to the generalization of formulas for weight distributions of binary codes to the nonbinary case [8].

For any $\tau = (\tau^{(1)},\cdots,\tau^{(n-k)})$ $(\tau^{(i)} \in \{0,\cdots,q-1\})$ denote $\bar{\tau} = (\bar{\tau}^{(1)},\cdots,\bar{\tau}^{(n-k)})$ where

$$\bar{\tau}^{(i)} = \begin{cases} q - \tau^{(i)}, & \text{if } \tau^{(i)} \neq 0; \\ 0, & \text{if } \tau^{(i)} = 0. \end{cases}$$

Then, for example, Theorem 1 may be modified to yield

$$A_i(f) = (i!)^{-1}\left(q^{-(n-k)}\hat{\hat{h}}^i(\overline{Hf}) - C_i(f)\right), \qquad i = 0, 1,\cdots,n,$$

where $h(x)$ is the characteristic function of columns for the extended check matrix, $\hat{h}(\omega)$ is the Fourier transform over GF(q) of $h(x)$ (defined by (1) with the replacement of $(-1)^{x\cdot\omega}$ by $\xi^{x\cdot\omega}$) and $C_i(f)$ is the number of i-tuples of (not necessarily distinct) q-ary vectors from $\{h_1, 2h_1,\cdots,(q-1)h_1,\cdots,h_n, 2h_n,\cdots,(q-1)h_n\}$ such that for every i-tuple $(h_{r_1},\cdots,h_{r_i})$ we have

1) $h_{r_1} \oplus \cdots \oplus h_{r_i} = Hf$;
2) there exists $\alpha, \beta \in \{1,\cdots,i\}$ and $A \in \{1,\cdots,q-1\}$ such that $h_{r_\alpha} = Ah_{r_\beta}$ (all the summations and multiplications here are carried out in GF(q)). We note also that $C_i(f)$ depends on q.

Formulas for $A_i(0)$ for small i and $q = 3$ are given in [8]. For weight distributions of dual codes we have the following generalization of (42) to the nonbinary case

$$wt(\omega H) = q^{-1}\big((q-1)n - \hat{h}(\omega)\big). \tag{76}$$

To compute $\hat{\hat{h}}^i$ it is expedient to use twice the algorithm of the corresponding fast Fourier transform [9], [14], which requires only $(n-k)q^{n-k}$ arithmetical operations and q^{n-k} memory cells. The computation of $C_i(f)$ for small i may be carried out by the method described in Section II.

We note also that the previous results may be further generalized to the case of codes over GF(q^s). In this case we need only take trace from GF(q^s) to GF(q) of the dot product $x\cdot\omega$ [28, p. 367].

Acknowledgment

The author would like to thank K. A. Post, Dr. H. F. Mattson, Jr., and both referees for their helpful comments.

References

[1] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error Correcting Codes*. New York: North-Holland, 1977.
[2] F. J. MacWilliams, "A theorem on the distribution of weights in a systematic code," *Bell Syst. Tech. J.*, vol. 42, pp. 79–94, 1963.
[3] V. Pless, "Power moment identities on weight distributions in error correcting codes," *Inform. Contr.*, vol. 6, pp. 147–152, 1963.
[4] F. J. MacWilliams, C. L. Mallows and N. J. A. Sloane, "Generalizations of Gleason's theorem on weight enumerators of self-dual

codes," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 794–805, Nov. 1972.
[5] C. L. Mallows and N. J. A. Sloane, "An upper bound for self-dual codes," *Inform. Contr.*, vol. 6, pp. 79–94, 1963.
[6] G. Cohen, P. Godlewski, and S. Perrine, "Sur les idempotents des codes," C. R. *Acad. Sci.*, vol. 284, Feb. 28, 1977.
[7] P. Delsarte, "Four fundamental parameters of a code and their combinatorial significance," *Inform. Contr.*, vol. 23, pp. 407–438, 1973.
[8] M. G. Karpovsky, "On the weight distribution of binary linear codes," *IEEE Trans. Inform. Theory*, vol. IT-25, pp. 105–109, Jan. 1979.
[9] ——, *Finite Orthogonal Series in the Design of Digital Devices*. New York: Wiley; Jerusalem: IUP, 1976.
[10] M. Deza, "Comparison of arbitrary additive noises," (in Russian), *Problemy Peredatchi Informatsii*, vol. 3, pp. 29–38, 1965.
[11] M. Deza and F. Hoffman, "Some results related to generalized Varshamov–Gilbert bound," *IEEE Trans. Inform. Theory*, pp. 517–518, July 1977.
[12] M. G. Karpovsky and V. D. Milman, "On subspaces contained in subsets of finite homogenous spaces," *Discrete Mathematics*, vol. 22, pp. 273–280, 1978.
[13] A. Tietavainen, "On nonexistence of perfect codes and related topics in combinatorics," (M. Hall, Jr. and J. H. Van Lint, Eds.), *Math. Center Tracks*, vol. 55, pp. 158–178, 1974.
[14] K. C. Andrews and K. L. Caspari, "A generalized technique for spectral analysis," *IEEE Trans. Comput.*, vol. C-19, pp. 16–25, 1970.
[15] M. G. Karpovsky and E. S. Moskalev, "Utilization of autocorrelation functions for the realization of systems of logical functions," *Automat. and Remote Contr.*, vol. 31, N2, pp. 243–250, Feb., 1970, (translated from *Automatika i Telemekhanika*, N2, pp. 83–90, 1970, Russian).
[16] M. G. Karpovsky and E. A. Trachtenberg, "Linear checking equations and error-correcting capability for computation channels," *Proc. 1977 IFIP Congress*. New York: North–Holland, 1977.
[17] N. Ahmed and K. R. Rao, *Orthogonal Transforms for Digital Signal Processing*. New York: Springer–Verlag, 1975.
[18] K. A. Post, private communication.
[19] G. Cohen and P. Frankl, "On tilings of the binary vector space," submitted to *Discrete Mathematics*.
[20] J. Vasiliev, "On nongroup closed packed codes," *Probl. Kibern.*, vol. 8, pp. 337–339, 1962.
[21] A. Tietavainen, "Nonexistence of perfect codes," *SIAM J. Appl. Math.*, vol. 24, pp. 88–96, 1973.
[22] O. S. Rothaus, "On Bent Functions," *J. Combinatorial Theory* (A), vol. 20, pp. 300–305, 1976.
[23] H. F. Mattson, Jr., private communication.
[24] N. J. A. Sloane and R. J. Dick, "On the enumeration of cosets of first-order Reed–Muller codes," *IEEE Int. Conf. on Commun.*, 7, Montreal, 1971.
[25] H. F. Mattson, Jr. and J. R. Schatz, "Maximum-Leader codes," to appear.
[26] J. J. Mykkeltveit, "The covering radius of the (128, 8) Reed–Muller code is 56," *IEEE Trans. Inform. Theory.*, vol. IT-26, pp. 359–362, May 1980.
[27] G. Cohen, Private Communication.
[28] E. F. Assmus, Jr. and H. F. Mattson, Jr., "Coding and combinatorics," *SIAM Rev.*, vol. 16, no. 3, pp. 345–388, July 1974.

Properties of Cross-Entropy Minimization

JOHN E. SHORE, SENIOR MEMBER, IEEE, AND RODNEY W. JOHNSON

***Abstract*— The principle of minimum cross-entropy (minimum directed divergence, minimum discrimination information) is a general method of inference about an unknown probability density when there exists a prior estimate of the density and new information in the form of constraints on expected values. Various fundamental properties of cross-entropy minimization are proven and collected in one place. Cross-entropy's well-known properties as an information measure are extended and strengthened when one of the densities involved is the result of cross-entropy minimization. The interplay between properties of cross-entropy minimization as an inference procedure and properties of cross-entropy as an information measure is pointed out. Examples are included and general analytic and computational methods of finding minimum cross-entropy probability densities are discussed.**

I. Introduction

THE PRINCIPLE of minimum cross-entropy provides a general method of inference about an unknown probability density $q^\dagger$ when there exists a prior estimate of $q^\dagger$ and new information about $q^\dagger$ in the form of constraints on expected values. The principle states that, of all the densities that satisfy the constraints, one should choose the *posterior* q with the least cross-entropy $H[q, p] = \int dx\, q(x) \log (q(x)/p(x))$, where p is a *prior* estimate of $q^\dagger$.

Cross-entropy minimization was first introduced by Kullback [1], who called it minimum directed divergence and minimum discrimination information. The principle of maximum entropy [2], [3] is equivalent to cross-entropy minimization in the special case of discrete spaces and uniform priors. Cross-entropy minimization has a long history of applications in a variety of fields (for a list of references, see [4]). Recently, the theory has been applied to problems in spectral analysis [5], speech coding [6], and pattern recognition [7].

It is useful and convenient to view cross-entropy minimization as one implementation of an abstract information operator $\circ$ that takes two arguments—a prior and new information—and yields a posterior. Thus, we write the posterior q as $q = p \circ I$, where I stands for the known

Manuscript received October 18, 1979; revised March 14, 1980.
The authors are with the Naval Research Laboratory, Code 7591, Washington, DC 20375.

0018-9448/81/0700-0472$00.75