Discrete Mathematics 83 (1990) 301–314
North-Holland

CODES WITH MULTI-LEVEL ERROR-CORRECTING CAPABILITIES*

Mao-Chao LIN
National Taiwan University, Taipei, Taiwan, ROC

Shu LIN
University of Hawaii, Honolulu, Hawaii 96822, USA

Received 29 December 1987
Revised 29 August 1988

In conventional channel coding, all the information symbols of a message are regarded equally significant, and hence codes are devised to provide equal protection for each information symbol against channel errors. However, in some circumstances, some information symbols in a message are more significant than the other symbols. As a result, it is desirable to devise codes with multi-level error-correcting capabilities. In this paper, we investigate block codes with multi-level error-correcting capabilities, which are also known as unequal error protection (UEP) codes. Several classes of UEP codes are constructed. One class of codes satisfies the Hamming bound on the number of parity-check symbols for systematic linear UEP codes and hence is optimal.

1. Introduction

In conventional channel coding, all the information symbols of a message are regarded equally significant, and hence redundant (or parity-check) symbols are added to provide equal protection for each information symbol against channel errors. However, in some occasions, some information symbols in a message are more significant than the other information symbols in the same message. Therefore, it is desirable to devise coding schemes which provide higher protection for the more significant information symbols. Suppose a message from an information source consists of m parts, each has a different level of significance and requires a different level of protection against channel errors. An obvious way to accomplish this is to use a separate code for each message part and then time share the codes. The redundant symbols of each code are designed to provide an appropriate level of error-correcting capability for the corresponding message part. This encoding scheme requires a separate encoder and decoder pair for each code. A more efficient way is to devise a single code for all the message parts. The redundant symbols are designed to provide m levels of error protection for m parts of a message. It has been proved that a single code with m levels of

* This research was supported by NSF Grant DCI-8418248-01 and NASA Grant NAG 5-407.
0012-365X/90/$03.50

error-correcting capability usually requires less redundant symbols than that required by time-sharing m separate codes with the same m levels of error-correcting capability [1–8]. Moreover, a single code requires only one encoder and one decoder. This may be desirable in many situations. A code with multi-levels of error-correcting capabilities is known as an unequal error protection (UEP) code. UEP codes were first studied by Masnick and Wolf [9], than by other coding theorists [5, 6, 10–20].

In this paper, we investigate codes with multi-level error-correcting capabilities. Two classes of multi-level UEP codes are presented. Each code in the first class is obtained by combining codes of shorter lengths. We find that a subclass of such codes meets the Hamming bound on the parity-check symbols for systematic linear UEP codes. Each of the second class of codes is achieved by taking direct sums of product codes. The minimum distances of such codes are greater than those for the simple product codes of comparable dimensions, besides, some message bits have extra error protection.

2. Cloud structure and the separation vector of a block code

Let $\{0, 1\}^n$ denote the vector space of all n-tuples over the binary field GF(2). Let V and W be two subsets of $\{0, 1\}^n$. Let $\boldsymbol{v}$ and $\boldsymbol{w}$ denote two vectors from V and W respectively. We define the separation between V and W, denoted $d(V, W)$, as follows:

$$d(V, W) \triangleq \min\{d(\boldsymbol{v}, \boldsymbol{w}) : \boldsymbol{v} \in V \text{ and } \boldsymbol{w} \in W\}, \tag{1}$$

where $d(\boldsymbol{v}, \boldsymbol{w})$ denotes the Hamming distance between $\boldsymbol{v}$ and $\boldsymbol{w}$. Clearly the separation $d(V, W)$ between V and W is simply a measure of distance between the two sets, V and W. Let $\boldsymbol{r}$ be a vector in $\{0, 1\}^n$. Then it is easy to show that the separations between $\{\boldsymbol{r}\}$, V and W satisfy the following triangle inequality,

$$d[\{\boldsymbol{r}\}, V] + d[\{\boldsymbol{r}\}, W] \geqslant d(V, W). \tag{2}$$

Consider a message space M which is the product of m component message spaces, $M_1, M_2, \ldots, M_m$. For $1 \leqslant i \leqslant m$, let $\boldsymbol{x}_i$ denote a message from the message space M_i. Then the product space M consists of the following set of m-tuples,

$$M = \{(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m) : \boldsymbol{x}_i \in M_i \text{ for } 1 \leqslant i \leqslant m\}. \tag{3}$$

Let C be a binary block code of length n for the product message space M. Let $\boldsymbol{v}(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m)$ denote the codeword for the message $(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m)$ from M. Let $\boldsymbol{a}$ be a specific message in M_i. Consider the following subset of codewords in C,

$$Q_i(\boldsymbol{a}) = \{\boldsymbol{v}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{i-1}, \boldsymbol{a}, \boldsymbol{x}_{i+1}, \ldots, \boldsymbol{x}_m) : \boldsymbol{x}_j \in M_j \text{ for } 1 \leqslant j \leqslant m \text{ and } j \neq i\}. \tag{4}$$

This set $Q_i(\boldsymbol{a})$ is called an *i-cloud* of C corresponding to the message $\boldsymbol{a}$ in M_i.

There are $|M_i|$ i-clouds in C corresponding to $|M_i|$ messages in M_i. These i-clouds form a partition of C. For two distinct i-clouds, $Q_i(\boldsymbol{a})$ and $Q_i(\boldsymbol{b})$, the separation between them is $d(Q_i(\boldsymbol{a}), Q_i(\boldsymbol{b}))$. Then we define the minimum separation among the i-clouds of C as follows:

$$s_i \triangleq \min\{dQ_i(\boldsymbol{a}), Q_i(\boldsymbol{b})) : \boldsymbol{a}, \boldsymbol{b} \in M_i \text{ and } \boldsymbol{a} \neq \boldsymbol{b}\}. \tag{5}$$

It follows from (1), (4) and (5) that

$$s_i = \min\{d[\boldsymbol{v}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_i, \ldots, \boldsymbol{x}_m), \boldsymbol{v}(\boldsymbol{x}_1', \ldots, \boldsymbol{x}_i', \ldots, \boldsymbol{x}_m')] : \\ \boldsymbol{x}_l, \boldsymbol{x}_l' \in M_l \text{ for } 1 \leqslant l \leqslant m \text{ and } x_i \neq x_i'\}.$$

Geometrically, we may view that the code C consists of $|M_i|$ i-clouds, where any two i-clouds are separated by a distance at least s_i. This distance structure of i-clouds determines the level of error protection for component message $\boldsymbol{x}_i$. The m-tuple,

$$\boldsymbol{s} \triangleq (s_1, s_2, \ldots, s_m),$$

is called the *separation vector* of the block code C for the product space $M = M_1 \times M_2 \times \cdots \times M_m$. This separation vector determines the levels of error protection for the m component messages, $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m$. We readily see that the minimum Hamming distance of C is $d_{\min} = \min\{s_i : 1 \leqslant i \leqslant m\}$.

Now we are ready to show that the minimum separation s_i of the i-clouds of a block code C determines the level of error protection (or error correction) for the ith component message $\boldsymbol{x}_i$ from M_i. To do this we devise a nearest cloud decoding algorithm for which each component message is decoded independently. Suppose a codeword $\boldsymbol{v}$ is transmitted and a vector $\boldsymbol{r}$ is received. To decode the ith component message, we compute the separation between $\{\boldsymbol{r}\}$ and every i-cloud. Let $Q_i(\boldsymbol{a})$ be the i-cloud such that

$$d[\{\boldsymbol{r}\}, Q_i(\boldsymbol{a})] < d[\{\boldsymbol{r}\}, Q_i(\boldsymbol{x}_i)]$$

for any $\boldsymbol{x}_i \in M_i$ and $\boldsymbol{x}_i \neq \boldsymbol{a}$. Then the ith component message is decoded into $\boldsymbol{a}$. The ith component message contained in $\boldsymbol{r}$ will be decoded correctly provided that there are $\lfloor (s_i - 1)/2 \rfloor$ or fewer transmission errors in $\boldsymbol{r}$. To see this, let $\boldsymbol{v} = \boldsymbol{v}(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m)$ be the transmitted codeword. For $\boldsymbol{x}_i' \neq \boldsymbol{x}_i$, it follows from (2) that

$$d[\{\boldsymbol{r}\}, Q_i(\boldsymbol{x}_i)] + d[\{\boldsymbol{r}\}, Q_i(\boldsymbol{x}_i')] \geqslant d[Q_i(\boldsymbol{x}_i), Q_i(\boldsymbol{x}_i')]. \tag{6}$$

Since $d[Q_i(\boldsymbol{x}_i), Q_i(\boldsymbol{x}_i')] \geqslant s_i$ and $d(\boldsymbol{r}, \boldsymbol{v}) \geqslant d[\{\boldsymbol{r}\}, Q_i(\boldsymbol{x}_i)]$, we have

$$d[\{\boldsymbol{r}\}, Q_i(\boldsymbol{x}_i')] \geqslant s_i - d(\boldsymbol{r}, \boldsymbol{v}). \tag{7}$$

If there are $t_i = \lfloor (s_i - 1)/2 \rfloor$ or fewer transmission errors in $\boldsymbol{r}$, then $d(\boldsymbol{r}, \boldsymbol{v}) \leqslant t_i$. It follows from (6) and (7) that $d[\{\boldsymbol{r}\}, Q_i(\boldsymbol{x})] \leqslant t_i$ and $d[\{\boldsymbol{r}\}, Q_i(\boldsymbol{x}_i')] > t_i$. Hence, $d[\{\boldsymbol{r}\}, Q_i(\boldsymbol{x}_i)] < d[\{\boldsymbol{r}\}, Q_i(\boldsymbol{x}_i')]$ for $\boldsymbol{x}_i \neq \boldsymbol{x}_i'$. Thus, the decoding algorithm described above results in the correct i-cloud, $Q_i(\boldsymbol{x}_i)$, and hence the correct component message $\boldsymbol{x}_i$. However, if there are more than t_i errors in the received vector $\boldsymbol{r}$, the inequality $d[\{\boldsymbol{r}\}, Q_i(\boldsymbol{x}_i)] < d[\{\boldsymbol{r}\}, Q_i(\boldsymbol{x}_i')]$ for $\boldsymbol{x}_i \neq \boldsymbol{x}_i'$ may not hold. As a result, the ith component message is decoded incorrectly into some $\boldsymbol{x}_i' \neq \boldsymbol{x}_i$. Theorem 1 characterizes the multi-level error-correcting capabilities of a block code.

Theorem 1. *Let C be a block code for the product of m message spaces, $M_1, M_2, \ldots, M_m$. Let $\boldsymbol{s} = (s_1, s_2, \ldots, s_m)$ be the separation vector of C. Then, for $1 \leqslant i \leqslant m$, the ith component message contained in a received vector can be correctly decoded provided that the number of transmission errors in the received vector is $\lfloor (s_i - 1)/2 \rfloor$ or less.*

A code C with a separation vector $\boldsymbol{s} = (s_1, s_2, \ldots, s_m)$ is called a $(t_1, t_2, \ldots, t_m)$-error-correcting code where $t_i = \lfloor (s_i - 1)/2 \rfloor$ for $1 \leqslant i \leqslant m$ and is the error correcting capability of the code for the ith component message $\boldsymbol{x}_i$. If $t_1, t_2, \ldots, t_m$ are all distinct, then C provides m levels of error-correcting capabilities, one for each component message. In this case, C is called a m-level error-correcting code or a m-level UEP code. Without loss of generality, we assume that $s_1 \geqslant s_2 \geqslant \cdots \geqslant s_m$ throughout of this paper.

The concept of separation vector was first introduced by Dunning and Robbins [13]. The separation vector defined in this paper is a generalization of Dunning and Robbins', which applies for either linear or nonlinear codes. Note that the minimum separation s_i for the i-clouds depends on how a code is partitioned into the i-clouds. Different encoding (or mapping) of M onto C yields different partitions of C. As a result, the separation vector of C depends on the encoding mapping.

3. Direct-sum codes for unequal error protection

An approach for constructing multi-level UEP codes is to take direct-sums of linear component codes. For $1 \leqslant i \leqslant m$, let C_i be a binary (n, k_i) linear block code for the message space $M_i = \{0, 1\}^{k_i}$. For $i \neq j$, we require that $C_i \cap C_j$ contains only the all-zero n-tuple $\mathbf{0}$. Let $\boldsymbol{v}(\boldsymbol{x}_i)$ denote the codeword in C_i for the message $\boldsymbol{x}_i \in M_i$. Let C be the direct-sum of $C_1, C_2, \ldots, C_m$, denoted $C = C_1 \oplus C_2 \oplus \cdots \oplus C_m$. Then C is an (n, k) linear code for the product message space $M = M_1 \times M_2 \times \cdots \times M_m$ where $k = k_1 + k_2 + \cdots + k_m$. For any message $(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m)$ in M, the corresponding codeword is

$$\boldsymbol{v}(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m) = \boldsymbol{v}(\boldsymbol{x}_1) + \boldsymbol{v}(\boldsymbol{x}_2) + \cdots + \boldsymbol{v}(\boldsymbol{x}_m). \quad (8)$$

Let $\{j_1, j_2, \ldots, j_l\}$ be a subset of $\{1, 2, \ldots, m\}$. Consider the direct-sum,

$$C(j_1, j_2, \ldots, j_l) = C_{j_1} \oplus C_{j_2} \oplus \cdots \oplus C_{j_l}.$$

Then $C(j_1, j_2, \ldots, j_l)$ is a subcode of C. An i-cloud of C for the component message $\boldsymbol{x}_i$ from M_i is simply the following set:

$$Q_i(\boldsymbol{x}_i) = \boldsymbol{v}(\boldsymbol{x}_i) \oplus C(1, \ldots, i-1, i+1, \ldots, m).$$

The vector $\boldsymbol{v}(\boldsymbol{x}_i)$ is in the i-cloud $Q_i(\boldsymbol{x}_i)$ and is called the center of $Q_i(\boldsymbol{x}_i)$. A vector in $Q_i(\boldsymbol{x}_i)$ is of the form $\boldsymbol{v}(\boldsymbol{x}_i) + \boldsymbol{w}$, where $\boldsymbol{w} \in C(1, \ldots, i-1, i+1, \ldots, m)$.

Let $w(\boldsymbol{v})$ denote the Hamming weight of the vector $\boldsymbol{v}$. Since $d(\boldsymbol{v}, \boldsymbol{u}) = w(\boldsymbol{v} + \boldsymbol{u})$, the minimum separation of the i-clouds of C is

$$s_i = \min\{w[\boldsymbol{v}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_i, \ldots, \boldsymbol{x}_m)] : \boldsymbol{x}_i \neq 0\}. \tag{9}$$

Theorem 2. *Consider an (n, k) linear code C which is the direct sum of codes $C_1, C_2, \ldots, C_m$, where C_i is an (n, k_i) linear code for the component message space $M_i = \{0, 1\}^{k_i}$ for $1 \leqslant i \leqslant m$. If the minimum weight of codewords in $C - C(i+1, i+2, \ldots, m)$ is at least d_i and $d_1 \geqslant d_2 \geqslant \cdots \geqslant d_m$, then C is an m-level error-correcting code for the product message space $M = M_1 \times M_2 \times \cdots M_m$ with separation vector $\boldsymbol{s} = (s_1, s_2, \ldots, s_m)$, where $s_i \geqslant d_i$ for $i = 1, 2, \ldots, m$.*

Proof. Note that for each codeword in $C(i+1, i+2, \ldots, m)$, the corresponding component message $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_i$ are all zero. Each codeword of C, $\boldsymbol{v}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_i, \ldots, \boldsymbol{x}_m)$ with $\boldsymbol{x}_i \neq 0$, is not in $C(i+1, i+2, \ldots, m)$ and hence has weight at least d_i. The proof then follows from (9). □

Theorem 2 describes a method of constructing multi-level UEP codes by taking direct sums of linear component codes. With this method, we are able to construct two classes of UEP codes.

4. Construction of linear multi-level UEP codes by combining shorter codes

Let $\boldsymbol{H}_{aa}$ and $\boldsymbol{H}_a = [\boldsymbol{H}_{aa}^{\mathrm{T}} \boldsymbol{H}_{ab}^{\mathrm{T}}]^{\mathrm{T}}$ be the parity-check matrices of an $(n_a, k_a + r)$ linear code C_{aa} and an (n_a, k_a) linear code C_a respectively, where $\boldsymbol{H}_{aa}$ is an $(n_a - k_a - r) \times n_a$ matrix, $\boldsymbol{H}_{ab}$ is a $r \times n_a$ matrix, $\boldsymbol{H}_a$ is an $(n_a - k_a) \times n_a$ matrix and T denotes the transpose operation. Let $\boldsymbol{H}_{bb}$ and $\boldsymbol{H}_b = [\boldsymbol{H}_{bb}^{\mathrm{T}} \boldsymbol{H}_{bb}^{\mathrm{T}}]^{\mathrm{T}}$ be the parity-check matrices of an $(n_b, k_b + r)$ linear code C_{bb} and an (n_b, k_b) linear code C_b respectively, where $\boldsymbol{H}_{bb}$ is an $(n_b - k_b - r) \times n_b$ matrix, $\boldsymbol{H}_{ba}$ is a $r \times n_b$ matrix and $\boldsymbol{H}_b$ is an $(n_b - k_b) \times n_b$ matrix. Consider the $(n_a + n_b, k_a + k_b + r)$ linear code C with the following parity-check matrix,

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{H}_{aa} & \boldsymbol{0} \\ \boldsymbol{H}_{ab} & \boldsymbol{H}_{ba} \\ \boldsymbol{0} & \boldsymbol{H}_{bb} \end{bmatrix}, \tag{10}$$

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**

fastcase® Smarter legal research.