NFORMATION AND CONTROL 37, 150–177 (1978)

Optimal Encodings of Linear Block Codes for Unequal Error Protection

LARRY A. DUNNING

Department of Mathematics, North Carolina State University, Raleigh, North Carolina 27607

AND

W. E. ROBBINS

Department of Computer Science, North Carolina State University, Raleigh, North Carolina 27607

It is possible for a linear block code to provide more protection for selected message positions than is guaranteed by the minimum distance of the code. The protection provided a message position can be measured by associating a number with that position called its separation. The separation of a message position measures the protection provided to that position in a manner analogous to that in which the minimum distance of a code measures the protection provided the entire message. This paper proves that any fixed linear block code has an encoding which is optimal with respect to the error protection provided the individual message positions. More precisely, among those encodings of the code for which the separations associated with the message positions are arranged in nondecreasing order, there is at least one which simultaneously maximizes all the separations associated with the message positions. A procedure is given which may be used to construct optimal encodings for linear codes of small dimension. When the Hamming metric is employed, the procedure builds a generator matrix which is as sparse as possible for the given code. At each iteration the procedure adds a row to a partially constructed generator matrix. A code word of minimum weight is chosen for this purpose—subject to the restriction that the rows of the generator matrix must be linearly independent. A more general result is that any generator matrix which is as sparse as possible induces an optimal encoding of its row space. A similar result holds when the Lee metric is used to model a channel. Theorems dealing with cyclic codes and product codes are developed. Under suitable restrictions, an optimal generator matrix for a cyclic code may be formed by concatenating the generator matrices of the minimal ideals which are contained in it. When the Hamming metric is employed, an optimal generator matrix for a product code may be obtained by taking the Kronecker product of optimal generator matrices for the component codes.

0019-9958/78/0372-0150$02.00/0

1. Introduction and Preliminaries

We shall restrict our attention to (n, k) block codes. Let $F \triangleq GF(q)$ be any finite field where q is a prime power. Throughout this paper, an (n, k) *code* will be a subset of F^n of cardinality q^k where $k \leqslant n$. By an *encoding* of a code, C, we mean any bijection $\eta: F^k \to C$. If C is a vector subspace of F^n, then it is said to be a linear code. In this case we will say that a $k \times n$ matrix with entries from F is a generator matrix for C if its rows form a basis for C. There is a natural one-to-one correspondence between the linear encodings and the generator matrices of a linear code. Every generator matrix, G, induces a linear encoding, L, defined by the formula,

$$L(m) \triangleq mG \qquad \forall m \in F^k,$$

where the message vector, m, and all vectors throughout the paper are identified with row matrices. For most applications of block codes, it is sufficient to study codes without reference to their encodings. However, this has not always been the case.

The construction of codes in which some message positions might be provided protection against a greater number of errors than others has been considered by several authors (Masnick and Wolf, 1967; Gore and Kilgus, 1971; Kilgus and Gore, 1972a; Mandelbaum, 1972). Masnick and Wolf (1967) proved that cyclic codes in systematic form provide equal error protection for every information digit. A nonsystematic cyclic code which provides one "information digit" protection against errors, beyond that guaranteed by the minimum distance of the code, was exhibited by Gore and Kilgus (1971). Thus, it became apparent that the protection against error afforded individual message positions depends not only on the code used, but also upon the encoding used. A direct means of establishing this result is to inspect the mappings η_1, η_2: $GF(2)^2 \to GF(2)^4$ given in Table I. η_1 and η_2 are two different encodings for the same code. Given a received word containing at most a single error, one can determine whether the code word originally transmitted was of the form $\eta_1(m_1, 0)$ or of the form $\eta_1(m_1, 1)$. Thus, the encoding, η_1, allows determination of the second message bit, m_2, despite any single error. However, consideration of the received word 1000 shows that the encoding η_2 fails to protect either message bit against all single errors.

TABLE I

$\eta_i(m_1, m_2)$	$\eta_1(m_1, 0)$	$\eta_1(m_1, 1)$	$\eta_2(m_1, 0)$	$\eta_2(m_1, 1)$
$m_1 = 0$	0000	0111	0000	0111
$m_1 = 1$	1100	1011	1011	1100

One place where *unequal-error-protection* codes were expected to find application was in the transmission of digital telemetry data. Here it may be desirable to give high order digits more protection than low order digits. Calculated data for several such codes so employed were given by Kilgus and Gore (1972b). Recently, several papers (Crimmins, 1976; Crimmins *et al.* 1969; Crimmins and Horowitz, 1970; Redinbo, 1976; Redinbo and Wolf, 1974; and Wolf and Redinbo, 1974) studying the *mean-square-error* protection afforded numeric data by block coding schemes have appeared. The approach in these papers is not to construct codes, but rather to find optimal encoding[1] and decoding schemes for a fixed linear code. Crimmins *et al.* (1969) gave a restricted formulation of the problem in which each encoding of a binary linear code generates a decoding scheme in a prescribed manner. They gave a procedure for finding linear encodings, which are optimal in the set of all encodings, linear and nonlinear, of the fixed binary linear code under consideration.

Our purpose is to investigate the encodings of a fixed linear code. However, we shall use the unequal-error-protection approach to evaluate and compare these encodings instead of the mean-square-error evaluation. The mean-square-error evaluation method of Crimmins *et al.* (1969) associates a nonnegative real number with each encoding. Since each code has only finitely many possible encodings, one of the encodings must have mean-square-error as small (good) as possible for the code. Thus, it is immediate that every code has an encoding which is optimal with respect to the mean-square-error evaluation. Using the unequal-error-protection approach we will prove that optimal encodings exist for linear codes. In doing this a procedure will be found for obtaining an encoding which is optimal in the set of all encodings, linear and nonlinear. This result parallels that of Crimmins *et al.* (1969), and the procedure found is similar to theirs. Further, when the encodings of a linear code are evaluated using a measure of unequal-error-protection based on either the Hamming or the Lee metric, the procedure will yield a linear encoding which is optimal among all encodings of the fixed linear code under consideration. In these cases, any generator matrix which has minimal Hamming or Lee weight, respectively, among all generator matrices for its row space, induces an encoding which is optimal for its row space.

Masnick and Wolf (1967) assign each information position an *error protection level*. Under this scheme, if an information position has error protection level, f, and not more than f errors occur in the reception of a code word, then the original value of the position in question can be determined correctly even though it may be impossible to determine the entire code word correctly. Instead of using this generalization of the error correcting capability of a code, we employ

[1] When numeric data are encoded using a 1–1 mapping (e.g., Crimmins *et al.*, 1969) from $\{0, 1, \ldots, 2^k - 1\}$ onto a code, we identify these integers with their binary representations (following Mitryayev, 1963) to obtain an equivalent encoding.

a generalization of the minimum distance of a block code. Given an encoding of a block code, for each message position we will define an associated *separation*, which is related to its error protection level in the same manner that the minimum distance of a block code is related to its error correcting capability. Encodings which we find to be optimal will necessarily be optimal with respect to their error protection levels.

Block codes may be used to detect errors, correct errors, fill in erasures, or combinations of these things. Fortunately, one parameter, minimum distance, suffices to measure the capabilities of a block code regardless of the type of protection desired—provided that one stays within the list given. However, different decoding algorithms are used depending on the task at hand. Given a particular encoding, the separation associated with a message position measures the capability of a block code to detect errors which may cause that position to be in error, determine that position despite errors, determine that position despite erasures, or combinations of these things in an analogous manner. The decoding algorithms, given later, differ very little from those used when all positions receive the same protection. Depending on the types of protection desired, the message positions may be decoded separately or as a unit. Treating the message as a unit will not necessarily preclude giving different positions varying degrees of protection.

As was mentioned before, the protection provided to the message positions depends upon the encoding as well as the code. The generator matrices and encodings of interest are frequently nonsystematic. That is, the message positions may not appear explicitly in the code words. We will *not* be able to make reference to "the information positions" of the code word. Before an encoding function can be used, an inverse mapping must be constructed for use as a part of the decoding rule. When we speak of choosing an optimal encoding, we will also be choosing decoding rules which will depend both on the encoding and on the type of error protection desired for each message position.

In order to handle the Hamming and Lee metrics simultaneously, we will develop results with respect to a function, $w: F^n \to \mathbb{R}$, which has the property that the function $d: F^n \times F^n \to \mathbb{R}$ given by

$$d(x, y) \triangleq w(x - y)$$

is a metric. Such a function, w, will be called a *weight function*. We will have occasion to refer to the Hamming weight function specifically and will denote it by h. One can easily verify that necessary and sufficient conditions for a function, $w: F^n \to \mathbb{R}$, to be a weight function are that for all $x, y \in F^n$

(i) $w(x) = 0$ if and only if $x = \overline{0}_n$,

(ii) $w(x) = w(-x)$,

(iii) $w(x + y) \leqslant w(x) + w(y)$,

where $\overline{0}_n$ denotes the zero vector in F^n.

Suppose $X \subseteq F^n$. It will be convenient to abbreviate $w[\Phi] = +\infty$, else

$$w[X] \triangleq \min_{x \in X} w(x).$$

This is not to be confused with the usual conventions for extending point functions to sets, i.e., for example, $w(X) \triangleq \{w(x): x \in X\}$.

Since we will be dealing with linear codes, which are the row space of their generator matrices, it is desirable to develop some notational devices to assist in arguments involving the rows of a matrix. Given a $k \times n$ matrix M with entries in F, we denote the entry in row i and column j by M_{ij}, the ith row (vector) by $M_{i\cdot}$ and the jth column (vector) by $M_{\cdot j}$. The set of all rows of M is denoted by

$$M_r \triangleq \{M_{1\cdot}, \ldots, M_{k\cdot}\}.$$

For any function, $f: F^n \to S$, where S is any set, define

$$f_r(M) \triangleq \begin{pmatrix} f(M_{1\cdot}) \\ \vdots \\ f(M_{k\cdot}) \end{pmatrix}.$$

Thus, $f_r(M) \in S^k$ is a vector whose ith component is found by applying f to the ith row of M.

Our main line of argument will require only some elementary knowledge of linear algebra. When listed, vectors will always be enclosed in parentheses and matrices in brackets. In our discussion of product codes, some properties of the (left) Kronecker product of matrices will be required. In particular, recall that

$$(A \otimes B)(C \otimes D) = AC \otimes BD.$$

We shall take Kronecker products over both F and $\mathbb{R}$ and will denote the respective operators by $\otimes_F$ and $\otimes_{\mathbb{R}}$.

The Hamming weight function, h, applied to a matrix will count the number of nonzero entries in that matrix. The Lee weight function applied to a matrix will add the Lee weights of its entries. It is easy to show that given two vectors $a \in F^k$, $b \in F^l$ their Hamming weights are related by

$$h(a \otimes_F b) = h(a)h(b). \tag{1}$$

This result may be used to prove that, given any two matrices A, B with entries in F,

$$h_r(A \otimes_F B) = h_r(A) \otimes_{\mathbb{R}} h_r(B). \tag{2}$$

We will denote the span operator by $\langle \cdot \rangle$. Given a *set* S of vectors taken from a finite vector space, V, over the field F,

$$\langle S \rangle = \left\{ \sum_{s \in S} \alpha_s s : \alpha \in F^S \right\}$$

DOCKET ALARM

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.