Hamming codes and some theory of linear error correcting codes

Burton Rosenberg

Preface

These notes were written for a combined undergraduate/graduate cryptography course. (Csc609/507 Spring 2003.) This material can mostly be found in the text *Introduction to Cryptography with Coding Theory* by Trappe and Washington. However the coverage has been trimmed to a very hands-on version closely motivated by the example of hamming codes.

Last update: 21 March 2003.

Introduction

The topic is how to reduce the error of communication in the presence of noise. Block encoding is an expansion function from a block of plaintext alphabet symbols to a block of communication alphabet symbols where the block lengths are k and n respectively, and $k \leq n$. Without loss of generality we consider both alphabets to be bit strings. The notation for an i length bit string is $\sigma \in \{\,0,1\,\}^i$. Hence our expansion (or encoding) function e is:

$$e : \{\,0,1\,\}^k \to \{\,0,1\,\}^n.$$

The set $M_n = \{\, e(\sigma) \,|\, \sigma \in \{\,0,1\,\}^l \,\}$ is the set of *codewords* for the error correcting code. The aim is that two distinct codewords $m_1, m_2 \in M_n$ should be adequately dissimilar that a few bit errors in transmission cannot cause the receiver confusion as to what codeword was sent. The errors will be corrected and the sent codeword recovered.

Shannon's noisy channel theorem says that codes exist with code rates arbitrarily close to the channel capacity. The question is how to construct such codes:

1. How to choose codewords so that any pair are adaquately dissimilar.
2. How to efficiently detect and repair errors in a damaged codeword.

We describe the Hamming code which is simple, can correct one error, has a very simple error correction procedure, and uses very few extra bits. We present it first straightforward, and then indicate that it is in the class of linear codes. We then give some general theory of linear error correction codes.

Hamming codes

Let r be a non-negative integer, the dimension of the parity space. Let $n = 2^r - 1$ be the code length and $k = n - r$ be the number of bits we will encode in each codeword. The codewords will have minimum Hamming distance of $d = 3$, so that one error can be corrected, two errors detected. Note that one can argue that in order to correct one error, the error's bit position must be determined. For an n bit code, $\log_2 n$ bits are therefore required. Since $k \approx n - \log n$ we can claim that we are sending at the maximal rate given the requirement of 1 bit error correction.

Let $\langle i \rangle_r$ be the r-bit vector of the representation of integer i, e.g. $\langle 1 \rangle_3 = [\ 1\ 0\ 0\ ]$. The i-th bit of a bit vector $b \in \{\, 0, 1 \,\}^n$ is written b_i, i.e. $b = b_1\ b_2\ \ldots\ b_n$. Then the set of codewords for the $[\ n, k\ ]$ *Hamming code* is the set of zero-parity words of length n,

$$M_n = \{\, b \in \{\, 0, 1 \,\}^n \mid \oplus_i\ b_i \cdot \langle i \rangle_{n-k} = 0 \,\}$$

We verbally describe the encoding procedure. To encode k bit string c, place each c_i into position b_j, consecutively, but skip over positions j such that $j = 2^s$, for some s. Now chose bit values for the omitted positions to force the result to be a codeword. This is easy because bit positions 2^s contribute simply to the calculation of parity,

$$\langle 2^s \rangle_{n-k} = [\ 0 \ldots 0\ 1\ 0 \ldots 0\ ]$$

with the 1 in the $s + 1$ position.

Example: Encode $[\ 0\ 1\ 0\ 1\ ]$ in a $[7, 4]$ Hamming code.

$$[\ 0\ 1\ 0\ 1\ ] \mapsto [\ *\ *\ 0\ *\ 1\ 0\ 1\ ]$$

which has parity,

$$\langle 5 \rangle_3 \oplus \langle 7 \rangle_3 = [\ 0\ 1\ 0\ ]$$

so we need set only bit 2 among the omitted bits:

$$[\ *\ *\ 0\ *\ 1\ 0\ 1\ ] \mapsto [\ 0\ 1\ 0\ 0\ 1\ 0\ 1\ ]$$

Error correction properties of Hamming codes

We now investigate the effects of a single bit error in a Hamming code. Suppose bit b_j is flipped (a 1 becomes a 0, a 0 becomes a 1). Notate this as $\bar{b_j}$. Then,

$$\begin{aligned}
\left(\bigoplus_{i \neq j} b_i \cdot \langle i \rangle_r \right) \oplus \bar{b_j} \cdot \langle j \rangle_r &= \left(\bigoplus_i b_i \cdot \langle i \rangle_r \right) \oplus b_j \cdot \langle j \rangle_r \oplus \bar{b_j} \cdot \langle j \rangle_r \\
&= b_j \cdot \langle j \rangle_r \oplus \bar{b_j} \cdot \langle j \rangle_r \\
&= \langle j \rangle_r
\end{aligned}$$

(recall $r = n - k$). Hence if the parity sums non-zero, the non-zero value is the index position of the bit error. Flip this bit to recover the correct codeword. Next, discard the r parity bits to recover the k information bits.

If there are two bit errors, we can show that the parity sum will not be zero. However, we will not be able to distinguish this with a single bit error, and continuing in the error correct process we will flip a third bit, further mangling the codeword. This is the nature of Hamming codes.

Furthermore, we can construct examples where three bit flips of a codeword yeilds another code word, hence, the Hamming distance of a Hamming code is 3.

Perfect codes

Hamming is *perfect* in that it has the largest number of codewords possible given the parameters, according to the *Hamming bound*, also known as the sphere packing bound.

Given $d = 2t + 1$, imagine that the space of words $\{0, 1\}^n$ is allocated to spheres with radius t, as measured by Hamming distance. In order that error correction of t errors be possible, the spheres cannot overlap, else a single word would be within t errors of two different codewords and it could possibly have been, originally, either of these codewords.

Hence the total area covered by the spheres cannot exceed the total number of words in the entire space. This gives the bound:

$$|M_n|(\text{area per sphere}) \leq 2^n$$

In the case of $d = 3$, the points contained in a sphere of radius 1 are the center and the n words resulting from bit flips in each of the n bit positions,

$$|M_n|(n + 1) \leq 2^n$$

In the specific case of Hamming codes,

$$\begin{aligned} |M_n| &\leq 2^n/(n + 1) \\ &= 2^{k+r}/(2^r - 1 + 1) \\ &= 2^k \end{aligned}$$

Since any k bits can be encoded uniquely, $M_n \geq 2^k$, so $M_n = 2^k$. We have shown that Hamming codes are perfect.

In the general case we need formulas for spheres of radius t other than 1, and for alphabets with q characters, rather than just two characters. Thus generalized, we have the Hamming bound as stated in the text.

Theorem 1 (Hamming bound) *Let M_n be a code of block length n over an alphabet of q symbols and with minimal Hamming distance between codewords $d \geq 2t + 1$. Then,*

$$|M_n| \leq \frac{q^n}{\sum_{i=0}^{t} \binom{n}{i}(q - 1)^i}$$

I leave it to the interested reader to derive this result.

Linearity: parity and generator matrices

We have approached error correcting codes through the specific example of the Hamming code. It is an example of a linear code. We explore linearity in this section and present matrix descriptions of encoding a decoding.

Consider M_n as a subset of the space of n vectors over the field F_2, $M_n \subseteq F_2^n$.

Theorem 2 (M_n is a subspace) *If $a, b \in M_n$ then $a + b \in M_n$.*

Proof: Since a and b are in M_n,

$$\oplus_i \, a_i \cdot \langle i \rangle = 0, \; \oplus_i \, b_i \cdot \langle i \rangle = 0,$$

therefore,

$$\oplus_i \, (a_i \oplus b_i) \cdot \langle i \rangle = 0.$$

The dimension of M_n is k, since given any k bits, we can choose the r parity bits to achieve parity zero. So M_n is a k dimensional subspace of F_2^n.

It is in fact the space of vectors $v \in F_2^n$ such that $P\,v = 0$ for the *parity check matrix* P. The parity check matrix for the $[7, 4]$ Hamming code is,

$$P = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

The array is $r \times n$ and has rank r. For m to be a codeword, it must be in the null space of this matrix, that is, $P\,m = 0$. This is just a rewording of our ad hoc approach, this time using the language of matrices.

Since we can set k for the bits anyway we like, and then get zero parity by chosing the parity bits correctly, the space M_n is k dimensional, an linear. So it can be described by a $k \times n$ dimensional matrix G of rank k.

We work out G for the specific example of the $[7, 4]$ Hamming code. The first row of G should be the n bit encoding of the k bit information vector [1 0 0 0]. This sets the third bit and to recover zero parity the first and second bit must be set,

$$[\, 1\, 0\, 0\, 0\,]G = [\, 1\, 1\, 1\, 0\, 0\, 0\, 0\,]$$

The vector [0 1 0 0] sets the fifth bit, and hence bits one and four are set to recover zero parity. Continuing,

$$G = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Theorem 3 (Parity Check) *Let P be a parity check matrix, and G a generator matrix for a linear code M_n. Then G maps the k information bits into the n code bits with $r = n - k$ bits of correction information, i.e. parity. We summarize the linear maps as follows:*

$$0 \to N \stackrel{G}{\to} M_n \stackrel{P^T}{\to} 0$$

(See appendix for explanation of exact sequences*.) Where N is the space F_2^k, k bits of information, and $M_n \subseteq F_2^n$. The map G uniquely encodes C into M_n. It is therefore a $k \times n$ matrix of rank k. The parity check matrix P is a $r \times n$ matrix of rank r which is zero on all codewords.*

Systematic codes

The defining relationship between the generator matrix G and the parity matrix P is that $G\,P^T = 0$, and both are of full rank. We have denoted the dimension of the parity space by r, so P is $r \times n$ and has rank r, for $r < n$. The remaining k degrees of freedom, $k = n - r$, is the dimension spanned by G. Hence G is a $k \times n$ matrix of rank k, for $k < n$.

The code M_n is the span of G, that is,

$$M_n = \{\, v\,G \,|\, v \in N \,\}$$

This space is not changed by modifications to G which bring it into a standard form. However, the exact associations between vectors in N and codewords will change. Ignoring this, use Gaussian elimination to bring G into the form,

$$G = \left[I_{k\times k}\ \widehat{P}_{k\times r} \right]$$

(This is a block form of a matrix. $I_{k\times k}$ is the k by k identity matrix, and it is followed on the right by $\widehat{P}$, a k by r matrix, the concatentation of the two submatrices forming a k by n matrix.) In this form the code is called *systematic*. It is clear that the first k bits of the code are the information bits and the last r bits are checksum bits, combinations of rows of $\widehat{P}$ which confirm the integrity of the information bits.

In our initial approach to the Hamming code, we handcrafted the parity matrix and then derived the corresponding generator matrix. Given the generator matrix for a systematic code, the parity matrix is mechanically derived,

$$P = \left[-(\widehat{P}_{k\times r})^T\ I_{r\times r} \right]$$

Theorem 4 *Matrices G and H defined as above, $G\,P^T = 0$, and each is of appropriate rank.*

Proof: We use the rules of block matrix multiplication (a userful thing to know):

$$\begin{aligned} G\,P^T &= \left[I_{k\times k}\ \widehat{P}_{k\times r} \right] \left[-(\widehat{P}_{k\times r})^T\ I_{r\times r} \right]^T \\ &= I_{k\times k}\,(-\widehat{P}_{k\times r}) + \widehat{P}_{k\times r}\,I_{r\times r} \\ &= -\widehat{P}_{k\times r} + \widehat{P}_{k\times r} \\ &= 0_{k\times r} \end{aligned}$$

Syndromes and coset leaders

We could have put our Hamming matrix G into the systematic form, but that would necessarily have disturbed our very simple error correction procedure. We now discuss the error correction procedure for a systematic code.

Our previous procedure was to form the parity sum and interpret the result as the index of a column. If the index was zero, no bit errors occured; else flip the bit in the column given by the index. The method generalizes:

1. Calculate the *syndrome* of the received bit string, $\mathcal{S}(b) = P\,b^T$. This is equivalent to the parity sum for the Hamming code.
2. Map by some rule the syndrome to the *coset leader*, the least weight element whose disturbance of a codeword would cause this syndrome. For the Hamming code, we considered the syndrome to be an integer, say i, and formed the bit vector zero everywhere except one in the i-th position.
3. Subtract the coset leader from the received bit string to recover the codeword.

The name coset leader comes from the mathematical term coset. The matrix P sends to zero any codeword in M_n. M_n is a subspace of the space M of n vectors, and vectors in this space but not in M_n are not mapped to zero by P. In fact, P categorizes the space M into a collection of disjoint subsets, one of which is M_n, and the other are translates of M_n. The collection of all translates are called *cosets.*

Let e be any vector in M. The coset $M_n(e)$ of e is the set,

$$M_n(e) = \{\, e + c \,|\, c \in M_n \,\}$$

We use the letter e because we will think of this vector as an error which is modifying the correct codeword. That is, every element in $M_n(e)$ is of the form $e + c$ where $c \in M_n$ is a valid codeword and e is the transmission error.

Since M_n is linear, $M_n(e) = M_n$ if $e \in M_n$. Since $0 \in M_n$, every $m \in M$ is in a coset, $m \in M_n(m)$. For two errors $e_1, e_2 \in M$, either their cosets are the same, $M_n(e_1) = M_n(e_2)$, or they are completely disjoint, $M_n(e_1) \cap M_n(e_2) = \emptyset$.

Proof: Suppose they touch at all, $x \in M_n(e_1) \cap M_n(e_2)$. *Then* $x = e_1 + c_1 = e_2 + c_2$ *so* $e_1 = e_2 + c_2 - c_1 = e_2 + c_3$, *where* $c_i \in M_n$. *Let* $y \in M_n(e_1)$. *Then* $y = e_1 + c_4 = e_2 + c_3 + c_4 = e_2 + c_5$, *so* $y \in M_n(e_2)$.

Furthermore, if two vectors are in the same coset, $m_1, m_2 \in M_n(e)$, they share syndromes,

$$\mathcal{S}(m_i) = \mathcal{S}(e + c_i) = P\,(e + c_i)^T = P\,e^T + P\,{c_i}^T = P\,e^T, \;\; i = 1, 2$$

for appropriate $c_i \in M_n$ which have syndrome 0.

The full picture extends our Parity Check theorem,

$$0 \to N \xrightarrow{G} M \xrightarrow{P^T} S \to 0$$

where S is the r dimensional space of syndromes, and P^T is also the syndrome map $\mathcal{S}$. (See appendix for explanation of *exact sequences*.)

A coset can be given by different translations. In the context of error correction, it is the vector of least weight which generates the coset that is the most likely way a codeword was modified into a member of the coset. This vector is called the *coset leader*. The coset is identified by taking the syndrome, and then the coset leader of this coset is understood to be the error vector that modified the codeword. We then subtract off the coset leader and recover the most likely codeword. This justifies our stated error correction procedure.

Hamming, revisted

The Hamming code was found to be a type of linear code. The simple development of the Hamming code was to assignment of parity values $1, 2, \ldots$ to the corresponding bit positions in a block of bits, and to select as codewords only those bit blocks which sum to zero parity. The theory then introduced systematic codes, a formula for the parity matrix given the generator matrix, and a decoding rule based on syndromes and coset leaders. We will redevelope Hamming using these concepts.

The generator matrix for the $[7, 4]$ Hamming code is,

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

We apply Gaussian elimination. Subtracting row one from rows two and four, then row two from row three, finally row three from row four,

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Now we clean up the first four colums by subtracting row four from row two, then row two from row one, then row three from row two,

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Therefore $\widehat{P}$ is,

$$\widehat{P} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

and the parity matrix,

$$P = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Check that $G\,P^T = 0$. Therefore M_n is the space of the 16 linear combinations of the rows of G, also the space of all vectors with syndrome 0. The first four bits of the codeword are the information bits, and the last three bits are parity. For example, the number 10 is encoded,

$$\begin{bmatrix} 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

The information bits are 0101 (10 base 2) and the parity bits are 010.

Suppose there is an error in the first bit position, so that 1101010 is received. We calculate the syndrome,

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

Hence we associate syndrome 011 with coset leader 1000000. In general, the i-th column of P is the syndrome for the coset $M_n(e_i)$ where e_i is the vector of all zeros except a one in index i.

In the original Hamming development, the presentation of P made the mapping from syndrome to coset leader more direct. However in this section we have demonstrated this mapping for systematic linear codes in a general manner.

Conclusion

By selecting codewords carefully it is possible to correct or detect errors in communication. As an example, the Hamming code corects one error using about $\log n$ extra bits for parity. Certain error correcting codes have algebraic structure. Hamming codes are linear, and can be manipulated using the rules of linear algebra.

Exact sequences

Algebra is concerned with mathematical objects and the mappings between objects. When the object in question has structure, the map should naturally respects that structure. For instance, linear spaces are related by linear maps. In a linear space, such as a vector space, vectors x and y

can sum to a third vector z. Therefore the map $L(x)$ and $L(y)$ should sum to $L(x+y)$,

$$L(x) + L(y) = L(x+y).$$

Linear spaces also have a special zero vector for which $x + 0 = x$. It follows that a linear map must take zero to zero, $L(0) = 0$.

The collection of all vectors which map to zero by a linear map L is the *kernel* of the map L. The kernel is a supspace: it contains 0 and is closed by addition, if $L(x) = L(y) = 0$ then $L(x+y) = 0$. A map also has an *image* or *range*. The range of L is also possible "outputs" of the function L,

$$\{\, L(x) \,|\, x \in X \,\}.$$

An *exact sequence* is a collection of three linear spaces connected by two maps such that the image of the first map is exactly the kernel of the second map:

$$X \xrightarrow{f} Y \xrightarrow{g} Z$$

This means that for all $x \in X$, $g(f(x)) = 0$, so that the image of f is contained in the kernel of g. Also, that for any $y \in Y$ for which $g(y) = 0$, there is an $x \in X$ such that $f(x) = y$, so that the image of f contains the kernel of g.

A *short exact sequence* is a collection of five linear spaces, beginning and ending with the the trivial vector space consisting of only the zero vector, connected by linear maps which are *exact*,

$$0 \to X \xrightarrow{f} Y \xrightarrow{g} Z \to 0$$

The point of writing down an exact sequence is to summarize the following information,

1. That the map f is injective, distinct x give distinct $f(x)$. Else the map $0 \to X$ would not be exact, as only 0 is in its image, so only 0 is in the kernel of f.
2. That the map g is surjective, the image of g is all of Z. Since the map $Z \to 0$ makes all of Z the kernel, all of Z must be the range of g.
3. Z is the set of cosets (translates) of $f(X)$ in Y. More generally, $Z = Y/X$.

More specifically, we called Z the space of syndromes, and the image of f was the set of codewords. The short exact sequence summarizes that information words are mapped to unique codewords whose syndrome is zero, and all zero syndrome words are codewords for some unique information word. Furthermore, communication words with non-zero syndromes are definitively attached to some correcting vector e which moves every word in the coset of that syndrome back into a codeword.