Hamming Codes

James Fiedler

Fall, 2004

In the late 1940s Richard Hamming recognized that the further evolution of computers required greater reliability, in particular the ability to detect and correct errors. (At the time, parity checking was being used to detect errors, but was unable to correct any errors.) He created the, Hamming Codes, perfect 1-error correcting codes, and the extended Hamming Codes, 1-error correcting and 2-error detecting codes. Hamming Codes are still widely used in computing, telecommunication, and other applications including data compression, popular puzzles, and turbo codes.

Definition A *code*, C is a subset of A^n, for some set or *alphabet* A.

Definition A code with a field as alphabet and which is a linear space over the field is a *linear code*.

Definition A *word* is an element of A^n. A *codeword* is an element of C.

Here the alphabets will be finite fields. Linear codes with length n and dimension k will be described as [n,k] codes. Hamming Codes are linear codes, and a Hamming Code will be described as a [n,k] q-ary Hamming Code, where q is the size of the base field, F_q. In other words an [n,k] q-ary Hamming Code is a linear subspace of the n-dimensional vector space over F_q. As an introduction, here is a concrete construction of a [7,4] binary Hamming Code.

Begin with 4 ordered bits (base field F_2) of information, and rename these as x_3, x_5, x_6, x_7, respectively. Choose

$$\begin{aligned} x_4 &= x_5 + x_6 + x_7 \ (\in F_2) \\ x_2 &= x_3 + x_6 + x_7 \ (\in F_2) \\ x_1 &= x_3 + x_5 + x_7 \ (\in F_2). \end{aligned}$$

Our Hamming codeword is then $(x_1\ x_2\ x_3\ x_4\ x_5\ x_6\ x_7)$ So, for instance, if we started with (1 0 1 0) we would get, step-by-step,

$$(x_1\ x_2\ 1\ x_4\ 0\ 1\ 1)$$
$$(x_1\ x_2\ 1\ 0\ 0\ 1\ 1)$$
$$(x_1\ 1\ 1\ 0\ 0\ 1\ 1)$$

$$(0\ 1\ 1\ 0\ 0\ 1\ 1).$$

Now, for the error correction we do the following. Let

$$\begin{aligned} a &= x_4 + x_5 + x_6 + x_7 \\ b &= x_2 + x_3 + x_6 + x_7 \\ c &= x_1 + x_3 + x_5 + x_7, \end{aligned}$$

and suppose that one error occurred during transmission of the codeword ($x_1\ x_2\ x_3\ x_4\ x_5\ x_6\ x_7$). Then abc, interpreted as a binary number, will give the subscript of the bit that is in error. For instance suppose we sent (0 1 1 0 0 1 1) and (0 1 1 0 1 1 1) was received. In this case we compute $a = 1$, $b = 0$, $c = 1$, and abc=101, which is the binary representation of 5 so we know that the 5^{th} bit is wrong. (If there is no error then $abc = 000$ indicates no error occurred.)

For a more general construction of [n,k] binary codes we need the definitions of generator and check matrices.

Definition A *generator matrix* **G** for an [n,k] linear code C (over any field F_q) is a k-by-n matrix for which the row space is the given code. In other words $C = \{\mathbf{xG}\ \ \mathbf{x} \in F_q^k\}$.

Definition The *dual* of a code C is the orthogonal complement, $C^\perp$.

Definition A *check matrix* for an [n, k] linear code is a generator matrix for the dual code.

If C is an [n,k] linear code, the dual to it is an [n, n-k] linear code. If **M** is the check matrix for C, **M** is an $(n-k) \times k$ matrix the rows of which are orthogonal to C and $\{\mathbf{x} \mid \mathbf{Mx}^\mathsf{T} = 0\} = C$.

Our general construction of a binary Hamming Code is actually a construction of a matrix, from which we'll define the Hamming Code as the linear code for which this matrix is the check matrix. For a given positive integer r form an $r \times (2^r - 1)$ matrix **M** by making the columns the binary representations of $1, \ldots, 2^r - 1$ (not necessary in increasing order, though this is often most convenient). Now $\{\mathbf{x} \mid \mathbf{Mx}^\mathsf{T} = 0\}$ is a linear code of dimension $2^r - 1 - r$, which we define to be a $[2^r - 1, 2^r - 1 - r]$ binary Hamming Code. Call $\{\mathbf{x} \mid \mathbf{Mx}^\mathsf{T} = 0\}$ a $[2^r - 1,\ 2^r - 1 - r]$ binary Hamming Code.

The [7,4] binary Hamming Code ($r = 3$) first introduced has check matrix

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

the code is

$$\begin{aligned} C = \{\ & \begin{pmatrix}0\,0\,0\,0\,0\,0\,0\end{pmatrix}, \begin{pmatrix}1\,0\,0\,0\,0\,1\,1\end{pmatrix}, \begin{pmatrix}0\,1\,0\,0\,1\,0\,1\end{pmatrix}, \begin{pmatrix}0\,0\,0\,1\,1\,1\,1\end{pmatrix}, \\ & \begin{pmatrix}0\,0\,1\,1\,0\,0\,1\end{pmatrix}, \begin{pmatrix}0\,0\,1\,0\,0\,1\,1\end{pmatrix}, \begin{pmatrix}0\,0\,1\,0\,1\,1\,0\end{pmatrix}, \begin{pmatrix}0\,1\,1\,0\,0\,1\,1\end{pmatrix}, \\ & \begin{pmatrix}0\,1\,1\,0\,1\,1\,1\end{pmatrix}, \begin{pmatrix}0\,1\,0\,1\,1\,1\,1\end{pmatrix}, \begin{pmatrix}1\,0\,0\,0\,0\,1\,1\end{pmatrix}, \begin{pmatrix}1\,0\,1\,1\,0\,1\,0\end{pmatrix}, \\ & \begin{pmatrix}1\,1\,1\,0\,0\,0\,0\end{pmatrix}, \begin{pmatrix}1\,0\,0\,1\,1\,0\,0\end{pmatrix}, \begin{pmatrix}1\,1\,0\,0\,1\,1\,0\end{pmatrix}, \begin{pmatrix}1\,1\,1\,1\,1\,1\,1\end{pmatrix}\ \}. \end{aligned}$$

The construction also produces, for $r = 3$, the check matrix

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

which gives a different set of Hamming codewords, and thus a different [7,4] binary Hamming Code. The word (1 0 0 0 1 1 1) is in this latter code, but does not appear in the list for the former.

Definition The *Hamming distance* d_H between any two words of the same length is defined as the number of coordinates in which they differ.

Any mention of distance herein refers to Hamming distance.

Proposition The minimum distance between binary Hamming codewords is 3.

Proof: Suppose $\mathbf{x}$ and $\mathbf{y}$ are two codewords from a Hamming Code, C with check matrix $\mathbf{M}$. Then $\mathbf{x} - \mathbf{y} \in C$, since C is a linear code. If $d_H(\mathbf{x}, \mathbf{y}) = 1$, then $\mathbf{M}$(**x**-**y**) is a column of $\mathbf{M}$. All columns of $\mathbf{M}$ are nonzero, but if (**x**-**y**) is a Hamming codeword, then $\mathbf{M}$(**x**-**y**) $= 0$. Contradiction. If $d_H(\mathbf{x}, \mathbf{y}) = 2$ then $\mathbf{M}$(**x**-**y**) $= 0$ iff there are two columns of $\mathbf{M}$ which are linearly dependent. This is not the case, hence $d_H(\mathbf{x}, \mathbf{y}) \geq 3$ for all codewords $\mathbf{x}$, $\mathbf{y}$. Every check matrix for a binary Hamming Code will have three columns that are linearly dependent, so in fact some codewords are of distance 3.

It is easy to see the Hamming distance is a metric. Then any word within distance 1 to a codeword is, in fact, within distance 1 to a unique codeword. Thus if any Hamming codeword is transmitted and at most 1 error occurs then the original codeword is the unique codeword within distance 1 to the received word. Thus it is true that the minimum distance between any two Hamming codewords is ≥ 3, then it is true that Hamming Codes are 1-error correcting. Decoding any received word to this nearest Hamming codeword corrects any single error.

In the general case error correction is even easier than in the introductory concrete construction. Again, we are assuming no more than one error during transmission. The method is called syndrome decoding. In the case of binary Hamming Codes syndrome decoding takes the following form. Both the sender and receiver of a word have agreed on an [n,k] binary Hamming Code with check matrix $\mathbf{M}$. Upon receiving a word $\mathbf{y}$, the receiver will compute $\mathbf{My}^{\mathsf{T}}$, which will be a binary r-tuple and thus be either the zero r-tuple, in which case there is no error, or be identical to one column of **M** (since **M** cointains every possible nonzero binaryr-tuple as a column), say $\mathbf{m}_i$. This tells the receiver that an error has occurred in the i^{th} coordinate of y.

Consider the example from the concrete construction above where we sent (0 1 1 0 0 1 1) and (0 1 1 0 1 1 1) was received. The syndrome is

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

whic matches the 5^{th} column of the check matrix. We take this to mean that the 5^{th} bit of the received word is in error.

Hamming Codes run into problems if more than one error has occurred. If in the last example the word (1 1 1 0 1 1 1) is received (errors in places 1 and 5) then the syndrome will be

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

telling us to switch the 4^{th} bit, which will give the word (0 1 1 1 1 1 1). Thus more than one error can not be corrected, nor even can we know that more than one error has occurred. The extended Hamming Codes are slightly more powerful, able to detect when 2 errors have occurred as well as able to correct any single error.

Returning to the introductory construction of a [7,4] binary Hamming Code, we include a new "parity check" bit, x_0, with

$$x_0 = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7,$$

so that all eight digits sum to 0. The code now has length 8 and is still a linear code of dimension 4. We call this code an [8,4] extended binary Hamming Code. The construction of an extended binary Hamming Code which corrects one error and detects two follows the same procedure for any length n: just add a parity check bit.

Check matrices can easily be constructed for the extended binary Hamming Codes from the check matrix for a Hamming Code: add a zero column on the left, then a row of all 1's on the bottom. For example the Hamming Code with check matrix

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

becomes the extended Hamming Code with check matrix

$$\left[\begin{array}{c|ccccccc} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right].$$

Suppose that $\mathbf{x}$ and $\mathbf{y}$ are binary Hamming codewords of distance 3. Then one of $\mathbf{x}$ or $\mathbf{y}$ has even parity and the other odd, say $\mathbf{x}$ has even parity. If $\mathbf{x}'$ and $\mathbf{y}'$ are the extended Hamming codewords obtained from adding a check digit. Then $x_0 = 0$ since $\mathbf{x}$ has even parity and $y_0 = 1$ since $\mathbf{y}$ has odd parity. The distance between $\mathbf{x}'$ and $\mathbf{y}'$ is one more than the distance between $\mathbf{x}$ $\mathbf{y}$, so the minimum distance between codewords of an extended Hamming Code is 4. Now any received word with one error is distance 1 from a unique codeword, and a received word with 2 errors is not within distance 1 from any codeword. If a word is within distance 1 to a codeword then we decode the word to that codeword as before. If a word is not within distance 1 to any codeword, then we recognize that 2 errors have occurred and report that two errors have occurred.

Decoding with an extended Hamming Code is a little more complicated. Let $\mathbf{M}$ be a check matrix for an extended Hamming Code, let $\mathbf{M}$' be the (regular) Hamming Code matrix from which is was derived, and let $\mathbf{y} = (y_0\ y_1\ \ldots\ y_n)$ be a received word. Suppose only one error has occurred, and suppose first that it has occurred in the last n bits. Computing the first n rows of the syndrome $\mathbf{M}\mathbf{y}^{\mathsf{T}}$ is the same as computing the syndrome $\mathbf{M}'(y_1\ \ldots\ y_n)^{\mathsf{T}}$. This last syndrome will be nonzero (since there is an error in one of these bits). The last row of the syndrome $\mathbf{M}\mathbf{y}^{\mathsf{T}}$ will be 1, since the parity is off with only one error. Thus the syndrome matches a column of $\mathbf{M}$.

Suppose now that only the parity bit is in error. Then $\mathbf{M}'(y_1\ \ldots\ y_n)^{\mathsf{T}}$ will be zero, so the first n columns of $\mathbf{M}\mathbf{y}^{\mathsf{T}}$ will be zero, but the last column will be 1 since the parity is off again, thus $\mathbf{M}\mathbf{y}^{\mathsf{T}}$ will match the first column of $\mathbf{M}$.

Thus as long as the syndrome matches a column of the check matrix then we can assume 1 error has occurred and switch the bit of the word corresponding to that column.

Now suppose 2 errors have occurred. Wherever they occur the parity of the entire word will be correct, thus the syndrome will have a 0 in the last row and will not be a column of the check matrix. The syndrome will not be zero either since codewords of the extended Hamming Code are distance at least 4 apart. Thus a nonzero, non-column syndrome indicates 2 errors (assuming that no more than two errors occurred).

We can generalize our construction of binary Hamming Codes to q-ary Hamming Codes, where an [n,k] Hamming Code is now a linear space over a field of order q, prime. For a given r, choose a nonzero r-tuple of elements of F_q. Choose another r-tuple from F_q linearly independent from the first. Make this column 2. Continue choosing r-tuples such that any two are linearly independent until it is no longer possible to do so. Arrange these r-tuples as the columns of a matrix $\mathbf{M}$. There are $q^r - 1$ possible nonzero r-tuples, and each choice eliminates its (q-1) nonzero multiples from further consideration. Thus the number of columns of our matrix is $(q^r - 1)/(q - 1)$. Define the (linear) code for which this is a check matrix to be a $[(q^r - 1)/(q - 1), (q^r - 1)/(q - 1) - r]$ q-ary Hamming Code. Recall that a check matrix is the generator matrix of the dual code, which in this case must have dimension r, hence the dimension of our Hamming Code must be $n - r = (q^r - 1)/(q - 1) - r$.

A [4,2] ternary (3-ary) Hamming Code can be given by the check matrix

$$\begin{bmatrix} 1 & 1 & 2 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

which has Hamming Code

$$\begin{array}{l} \{\ \begin{pmatrix} 0\,0\,0\,0 \end{pmatrix}, \begin{pmatrix} 0\,1\,1\,1 \end{pmatrix}, \begin{pmatrix} 0\,2\,2\,2 \end{pmatrix}, \\ \quad \begin{pmatrix} 1\,0\,1\,2 \end{pmatrix}, \begin{pmatrix} 2\,0\,2\,1 \end{pmatrix}, \begin{pmatrix} 1\,1\,2\,0 \end{pmatrix}, \\ \quad \begin{pmatrix} 2\,2\,1\,0 \end{pmatrix}, \ \begin{pmatrix} 1\,2\,0\,1 \end{pmatrix} \ \begin{pmatrix} 2\,1\,0\,2 \end{pmatrix}\ \}. \end{array}$$

The same Hamming Code is produced with the check matrix

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 2 \end{bmatrix}.$$

A different [4,2] Hamming Code results from the check matrix

$$\begin{bmatrix} 0 & 2 & 1 & 2 \\ 1 & 0 & 1 & 1 \end{bmatrix}.$$

Hamming Code:

$$\begin{array}{l} \{\ \begin{pmatrix} 0\,0\,0\,0 \end{pmatrix}, \begin{pmatrix} 0\,1\,2\,1 \end{pmatrix}, \begin{pmatrix} 0\,2\,1\,2 \end{pmatrix}, \\ \quad \begin{pmatrix} 1\,0\,1\,1 \end{pmatrix}, \begin{pmatrix} 2\,0\,2\,2 \end{pmatrix}, \begin{pmatrix} 1\,1\,0\,2 \end{pmatrix}, \\ \quad \begin{pmatrix} 2\,2\,0\,1 \end{pmatrix}, \ \begin{pmatrix} 1\,2\,2\,0 \end{pmatrix} \ \begin{pmatrix} 2\,1\,1\,0 \end{pmatrix}\ \}. \end{array}$$

These q-ary codes are again 1-error correcting, relying on that fact that each codeword is at a distance of at least 3 from any other codeword, which in turn relies on the construction of the check matrix. Specifically, the fact that no two columns of the check matrix are linearly dependent means that the minimum distance between any two codewords is at least 3.

Syndrome decoding (of at most one error) easily extends to the q-ary case. Again we compute the syndrome $\mathbf{My}^{\mathsf{T}}$ for check matrix $\mathbf{M}$ and received word $\mathbf{y}$. If the syndrome is the zero vector then we assume no error. If the syndrome is nonzero then it is a multiple of some column. If the syndrome is $\alpha\mathbf{m}_i$ for $\alpha\epsilon F_q$ and $\mathbf{m}_i$ the i^{th} column of $\mathbf{M}$, take this to mean that an error vector was added to the intended codeword during transmission, where the error vector is $(0 \ \ldots \ 0 \ \alpha \ 0 \ \ldots \ 0)$, with α in the i^{th} spot. Then we recover the intended codeword as $\mathbf{y} - (0 \ \ldots \ 0 \ \alpha \ 0 \ \ldots \ 0)$.

Definition A sphere of radius ρ centered at $\mathbf{x}$ is

$$S_\rho(\mathbf{x}) = \{\mathbf{y} \in F_q^n | \ d_H(\mathbf{x}, \mathbf{y}) \leq \rho\}.$$

For any $\mathbf{x} \in F_q^n$, $|S_0(\mathbf{x})| = 1$, that 1 being just $\mathbf{x}$. There are (q-1) ways to change any coordinate of $\mathbf{x}$ and get a new word, and n possible coordinates to change. Hence $|S_1(\mathbf{x})| = 1 + n(q-1)$. There are $\binom{n}{2}$ ways to choose two coordinates to change, and $(q-1)^2$ ways to change the two, so $|S_2(\mathbf{x})| = 1 + n(q-1) + \binom{n}{2}(q-1)^2$. In general,

$$|S_\rho(\mathbf{x})| = \sum_{i=0}^{\rho} \binom{n}{i}(q-1)^i.$$

From discussion above, a code is e-error correcting if $S_e(\mathbf{x}) \cap S_e(\mathbf{y}) = \emptyset$ whenever $\mathbf{x} \neq \mathbf{y}$. Thus, if C is an e-error correcting code, $|C| \cdot |S_e(x)| \leq |F_q|^n$, (for any $\mathbf{x}$ since the cardinality of the sphere does not depend on the choice of $\mathbf{x}$). This inequality is called the sphere-packing bound.

Definition An e-error correcting code is called a perfect e-error correcting code if $|C| \cdot |S_e(x)| \leq |F_q|^n$, or if

$$|C| = |F_q|^n / \sum_{i=0}^{e} \binom{n}{i}(q-1)^i$$

Proposition Hamming Codes are perfect 1-error correcting codes.

Proof: We need to check that

$$|C| \cdot \sum_{i=0}^{1} \binom{n}{i}(q-1)^i = |F_q|^n.$$

The right hand side of this is q^n, where $n = (q^r - 1)/(q-1)$. The left hand side is

$$\begin{aligned} q^{n-r}\left(1 + n(q-1)\right) &= q^{n-r}\left(1 + \frac{(q^r - 1)}{(q-1)}(q-1)\right) \\ &= q^{n-r}\left(1 + (q^r - 1)\right) \\ &= q^{n-r}(q^r) \\ &= q^n. \end{aligned}$$

Thus Hamming Codes are perfect 1-error correcting codes.

Definition A *e-covering code* is a code $C \subset A^n$ for which $A^n = \cup\{S_e(\mathbf{x}| \mathbf{x} \in C\}$.

Proposition If C is an e-covering code then $|C| \cdot |S_e(x)| \geq |F_q|^n$.

Definition If equality holds in the last proposition then C is called a *perfect e-covering code.*

Of course the perfect e-error correcting codes and the perfect e-covering codes coincide. It is known that Hamming Codes are the only perfect 1-error correcting codes.