Linear block codes and group codes

Definition: A *linear block code* over a field F is a linear subspace of F^n, where n is the blocklength of the code.

Definition: A *group code* is a subgroup of the n-tuples over an additive group.

Facts about linear block codes and group codes.

- In a group code the sum and difference of codewords are codewords.
- In a linear block code, also scalar multiples of codewords are codewords.
- Every linear block code is a group code, but not conversely.
- A *binary* group code is a linear block code because the only scalars are 0 and 1.
- Parity-check codes are linear block codes over $\mathrm{GF}(2)$.
  Every PC code is defined by a set of homogeneous binary equations.
- If $\mathcal{C}$ is a LBC over $\mathrm{GF}(Q)$ of dimension k, then its rate is $R = \dfrac{\log_Q Q^k}{n} = \dfrac{k}{n}$.

Minimum weight

The *Hamming weight* $w_H(v)$ is the number of nonzero components of v.

Obvious facts:

- $w_H(v) = d_H(0, v)$
- $d_H(v_1, v_2) = w_H(v_1 - v_2) = w_H(v_2 - v_1)$
- $w_H(v) = 0$ if and only if $v = 0$

Definition: The *minimum (Hamming) weight* of a block code is the weight of the nonzero codeword with smallest weight:

$$w_{\text{min}} = w^* = \min\{w_H(c) : \, c \in \mathcal{C}, \, c \neq 0\}$$

Examples of minimum weight:

- Simple parity-check codes: $w^* = 2$.
- Repetition codes: $w^* = n$.
- (7,4) Hamming code: $w^* = 3$. (There are 7 codewords of weight 3.)
  Weight enumerator: $A(x) = 1 + 7x^3 + 7x^4 + x^7$.
- Simple product code: $w^* = 4$.

Minimum distance = minimum weight

Theorem: For every linear block code, $d^* = w^*$.

Proof: We show that $w^* \geq d^*$ and $w^* \leq d^*$.

($\geq$) Let c_0 be a nonzero minimum-weight codeword. the 0 vector is a codeword, so

$$w^* = w_H(c_0) = d_H(0, c_0) \;\geq\; d^* \,.$$

($\leq$) Let $c_1 \neq c_2$ be two closest codewords. Then $c_1 - c_2$ is a nonzero codeword, s

$$d^* = d_H(c_1, c_2) = w_H(c_1 - c_2) \;\geq\; w^* \,.$$

Combining these two inequalities, we obtain $d^* = w^*$.

It is easier to find minimum weight than minimum distance because the weight minimization considers only a single parameter.

Computer search: test all vectors of weight $1, 2, 3, \ldots$ until codeword is found.

It is also easier to determine the weight distribution of a linear code than the distance distribution of a general code.

The result $d^* = w^*$ holds for group codes, since the proof used only subtraction.

Generator matrix

Definition: A *generator matrix* for a linear block code $\mathcal{C}$ of blocklength n and dimension k is any $k \times n$ matrix G whose rows form a basis for $\mathcal{C}$.

Every codeword is a linear combination of the *rows* of a generator matrix G:

$$\mathbf{c} = \mathbf{mG} \;=\; [\, m_0\, m_1\, \ldots\, m_{k-1}\,] \begin{bmatrix} \mathbf{g_0} \\ \mathbf{g_1} \\ \vdots \\ \mathbf{g_{k-1}} \end{bmatrix}$$

$$= m_0\mathbf{g_0} + m_1\mathbf{g_1} + \cdots + m_{k-1}\mathbf{g_{k-1}} \,.$$

Since G has rank k, the representation of $\mathbf{c}$ is unique.

Each component of $\mathbf{c}$ is inner product of $\mathbf{m}$ with corresponding *column* of G:

$$c_j \;=\; m_0 g_{0,j} \;+\; m_1 g_{1,j} \;+\; \cdots \;+\; m_{k-1} g_{k-1,j} \,.$$

Both sets of equations can be used for encoding. In either case, each codeword symbol requires k multiplications by constants and $k-1$ additions.

Parity-check matrix

Definition: The *dual code* of $\mathcal{C}$ is the orthogonal complement $C^{\perp}$.

Definition: A *parity-check matrix* for a linear block code $\mathcal{C}$ is any $r \times n$ matrix H whose rows span the orthogonal complement $C^{\perp}$. Obviously, $r \geq n-k$.

Example: G and H for $(5,4)$ simple parity-check code.

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

(H is generator matrix for the $(5,1)$ repetition code — the dual code.)

Example: G and H for $(7,4)$ *cyclic* Hamming code.

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

A *cyclic* code is a linear block code such that the cyclic shift of every codeword is also a codeword. It is *not* obvious by inspection that this property holds for the code generated by G.

Codewords are defined by H

Theorem: If $\mathcal{C}$ is an (n,k) linear block code with parity-check matrix H, then an n-tuple $\mathbf{c}$ is a codeword if and only if $\mathbf{c}H^T = 0$.

Proof:

($\Rightarrow$) Suppose $\mathbf{c}$ is a codeword.

Each component of $\mathbf{c}H^T$ is the inner product of $\mathbf{c}$ and a column of H^T, which is a row of H.

Since every row of H is in $C^{\perp}$, each row is $\perp$ to $\mathbf{c}$.

Thus each component of $\mathbf{c}H^T$ is $\mathbf{c} \cdot \mathbf{h}_i = 0$.

($\Leftarrow$) Since the rows of H span $C^{\perp}$, any n-tuple satisfying $\mathbf{c}H^T = 0$ belongs to the orthogonal complement of $C^{\perp}$.

By the Dimension Theorem (Blahut Theorem 2.5.10), $\mathcal{C}^{\perp\perp} = \mathcal{C}$.

Therefore if $\mathbf{c}H^T = 0$ then $\mathbf{c}$ belongs to $\mathcal{C}$.

($\Leftrightarrow$) Thus $\mathcal{C}$ consists of vectors satisfying the check equations $\mathbf{c}H^T = 0$.

Generator vs. parity-check matrices

Usually we choose H to consist of $n-k$ independent rows, so H is $(n-k)\times n$.

Sometimes it is convenient or elegant to use a parity-check matrix with redundant rows (for example, binary BCH codes, to be discussed later).

Each row of H corresponds to an equation satisfied by all codewords.

Since each row of G is a codeword, for any parity-check matrix H,

$$G_{k\times n}\cdot H^T_{r\times n} \;=\; \mathbf{0}_{k\times r} \quad (r \ge n-k)$$

Each 0 is $\mathbf{0}_{k\times r}$ corresponds to one codeword and one equation.

Conversely, if $GH^T = 0$ and $\operatorname{rank} H = n-k$ then H is a parity-check matrix.

How do we find H from G?

We could find H from G by finding $n-k$ linearly independent solutions of the linear equation $GH^T = 0$.

The equations $GH^T = 0$ are easy to solve when G is *systematic*.

Systematic generator matrices

Definition: A *systematic generator matrix* is of the form

$$G = [\,P\,|\,I\,] = \begin{bmatrix} p_{0,0} & \cdots & p_{0,n-k-1} & 1 & 0 & \cdots & 0 \\ p_{1,0} & \cdots & p_{1,n-k-1} & 0 & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{k-1,0} & \cdots & p_{k-1,n-k-1} & 0 & 0 & \cdots & 1 \end{bmatrix}$$

Advantages of systematic generator matrices:

- Message symbols appear unscrambled in each codeword in the rightmost positions $n-k,\dots,n-1$.
- Encoder complexity is reduced; only check symbols need be computed:
$$c_j \;=\; m_0 g_{0,j} \;+\; m_1 g_{1,j} \;+\cdots+\; m_{k-1} g_{k-1,j} \quad (j = 0,\dots,n-k-1)$$
- Check symbol encoder equations easily yield parity-check equations:
$$c_j - c_{n-k} g_{0,j} - c_{n-k+1} g_{1,j} - \cdots - c_{n-1} g_{k-1,j} \;=\; 0 \quad (m_i = c_{n-k+i})$$
- Systematic parity-check matrix is easy to find: $H = [\,I\,|\,{-P^T}\,]$.

Systematic parity-check matrix

Given a $k \times n$ systematic generator matrix

$$G = [\,P \,|\, I_k\,] = \begin{bmatrix} p_{0,0} & \cdots & p_{0,n-k-1} & 1 & 0 & \cdots & 0 \\ p_{1,0} & \cdots & p_{1,n-k-1} & 0 & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{k-1,0} & \cdots & p_{k-1,n-k-1} & 0 & 0 & \cdots & 1 \end{bmatrix},$$

The corresponding $(n-k) \times n$ systematic parity-check matrix is

$$H = [\,I_{n-k} \,|\, -P^T\,] = \begin{bmatrix} 1 & 0 & \cdots & 0 & -p_{0,0} & \cdots & -p_{k-1,0} \\ 0 & 1 & \cdots & 0 & -p_{0,1} & \cdots & -p_{k-1,1} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & -p_{0,n-k} & \cdots & -p_{k-1,n-k} \end{bmatrix}$$

(The minus signs are not needed for fields of characteristic 2, i.e., $\mathrm{GF}(2^m)$.)

Each row of H is corresponds to an equation satisfied by all codewords.

These equations simply tell how to compute the check symbols $c_0, \ldots, c_{n-k-1}$ in terms of the information symbols $c_{n-k}, \ldots, c_{n-1}$.

Minimum weight and columns of H

Since $\mathbf{c}H^T = 0$ for every codeword $\mathbf{c} = (c_0, c_1, \ldots, c_{n-1})$, every *nonzero* codeword determines a linear dependence among a subset of the *rows* of H^T. Thus

$$(\mathbf{c}H^T)^T = H\mathbf{c}^T = c_0 h^0 + c_1 h^1 + \cdots + c_{n-1} h^{n-1} = 0$$

is a linear dependence among a subset of the *columns* of H.

$Theorem$: The minimum weight of a linear block code is the smallest number of linearly dependent columns of any parity-check matrix.

$Proof$: Each linearly dependent subset of w columns corresponds to a codeword of weight w.

A set of columns of H is linearly dependent if one column is a linear combination of the other columns.

- A LBC has $w^* \le 2$ iff one column of H is a multiple of another column.
- Binary Hamming codes have $w^* = 3$ because no columns of H are equal.

The Big Question: how to find H such that no 5 (or 7 or more) columns are LI?

Computing minimum weight

The *rank* of H is the maximum number of linearly independent columns.

It can be determined in time $O(n^3)$ using linear operations — Gaussian elimination.

The minimum distance is the smallest number of linearly dependent columns.

Finding the minimum distance is difficult (NP-hard). We might have to look at large numbers of subsets of columns.

Solution: design codes whose minimum distance can be proven to have desired lower bounds.

The dimension of the column space of H is $n-k$. Thus *any* $n-k+1$ columns are linearly dependent. Therefore

$$d^* = w^* \le n-k+1$$

for any linear block code. This is known as the *Singleton bound*.

Exercise: Show that the Singleton bound holds for all (n,k) block codes, not just linear codes.

Maximum distance separable codes

Codes that achieve Singleton bound are called *maximum-distance separable* (MDS) codes.

Repetition code satisfies the Singleton bound with equality:

$$d^* = n = (n-1)+1 = (n-k)+1$$

Another class of MDS codes are the simple parity-check codes:

$$d^* = 2 = 1+1 = (n-k)+1$$

The best known *nonbinary* MDS codes are the Reed-Solomon codes over $\mathrm{GF}(Q)$. The RS code parameters are

$$(n,k,d^*) = (Q-1,\ Q-d^*,\ d^*) \ \Rightarrow\ n-k = d^*-1\,.$$

Exercise: Show that the repetition codes and the simple parity-check codes are the only nontrivial *binary* MDS codes.

Linear block codes: review

- An (n,k) linear block code is a k-dimensional subspace of a finite field F^n. Sums, differences, and scalar multiples of codewords are also codewords.
- A group code over an additive group G is closed under sum and difference.
- An (n,k) LBC over $F = \mathrm{GF}(q)$ has $M = q^k$ codewords and rate k/n.
- A linear block code $\mathcal{C}$ can be defined by two matrices.
  - Generator matrix G: rows of G are basis for $\mathcal{C}$, i.e., $\mathcal{C} = \{\mathbf{m}G : \mathbf{m} \in F^k\}$
  - Parity-check matrix H span $C^\perp$, hence $\mathcal{C} = \{\mathbf{c} \in F^n : \mathbf{c}H^T = 0\}$
- The Hamming weight of an n-tuple is the number of nonzero components.
- The minimum weight w^* of a block code is the Hamming weight of the nonzero codeword of minimum weight.
- The minimum distance of every LBC equals the minimum weight: $d^* = w^*$.
- The minimum weight of a linear block code is the smallest number of linearly dependent columns of any parity-check matrix.

Syndrome decoding

Linear block codes are much simpler than general block codes:

- Encoding is vector-matrix multiplication.

  (Cyclic codes are even simpler — polynomial multiplication/division is used.)
- Decoding is inherently nonlinear. Fact: linear decoders are very weak.

  However, several steps in the decoding process are linear:
  - syndrome computation
  - final correction after error pattern and location have been found
  - extracting estmated message from estimated codeword

Definition: The *error vector* or *error pattern* $\mathbf{e}$ is the difference between the received n-tuple $\mathbf{r}$ and the transmitted codeword $\mathbf{c}$:

$$\mathbf{e} \overset{\Delta}{=} \mathbf{r} - \mathbf{c} \Rightarrow \mathbf{r} = \mathbf{c} + \mathbf{e}$$

Note: The physical noise model may not be additive noise, and the probability distribution for the error $\mathbf{e}$ may depend on the data $\mathbf{c}$. We assume a channel error model determined by $\mathrm{P}(\mathbf{e})$.

Syndrome decoding (2)

Multiply both sides of the equation $\mathbf{r} = \mathbf{c} + \mathbf{e}$ by H:

$$\mathbf{s} \stackrel{\Delta}{=} \mathbf{r}H^T = (\mathbf{c} + \mathbf{e})H^T = \mathbf{c}H^T + \mathbf{e}H^T = \mathbf{0} + \mathbf{e}H^T = \mathbf{e}H^T .$$

The *syndrome* of the senseword $\mathbf{r}$ is defined to be $\mathbf{s} = \mathbf{r}H^T$.

The syndrome of $\mathbf{r}$ (known to receiver) equals the syndrome of the error pattern $\mathbf{e}$ (not known to receiver but must be estimated).

Decoding consists of finding the most plausible error pattern $\mathbf{e}$ such that

$$\mathbf{e}H^T = \mathbf{s} = \mathbf{r}H^T .$$

"Plausible" depends on the error characteristics:

- For binary symmetric channel, most plausible means smallest number of bit errors. Decoder picks error pattern of smallest weight satisfying $\mathbf{e}H^T = \mathbf{s}$.
- For bursty channels, error patterns are plausible if the symbol errors are close together.

Syndrome decoding (3)

Syndrome table decoding consists of these steps:

1. Calculate syndrome $\mathbf{s} = \mathbf{r}H^T$ of received n-tuple.
2. Find most plausible error pattern $\mathbf{e}$ with $\mathbf{e}H^T = \mathbf{s}$.
3. Estimate transmitted codeword: $\hat{\mathbf{c}} = \mathbf{r} - \mathbf{e}$.
4. Determine message $\hat{\mathbf{m}}$ from the encoding equation $\hat{\mathbf{c}} = \hat{\mathbf{m}}G$.

Only step 2 requires nonlinear operations.

For small values of $n - k$, lookup tables can be used for step 2.

For BCH and Reed-Solomon codes, the error locations are the zeroes of certain polynomials over the channel alphabet.

These *error locator polynomials* are linear functions of the syndrome.

Challenge: find, then solve, the polynomials.

Step 4 is not needed for systematic encoders, since $\mathbf{m} = \hat{\mathbf{c}}[n{-}k : n{-}1]$.

Syndrome decoding: example

An $(8,4)$ binary linear block code $\mathcal{C}$ is defined by systematic matrices:

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

Consider two possible messages:

$$\mathbf{m}_1 = [\,0\,1\,1\,0\,] \qquad \mathbf{m}_2 = [\,1\,0\,1\,1\,]$$

$$\mathbf{c}_1 = [\,0\,1\,1\,0\,0\,1\,1\,0\,] \qquad \mathbf{c}_2 = [\,0\,1\,0\,0\,1\,0\,1\,1\,]$$

Suppose error pattern $\mathbf{e} = [\,0\,0\,0\,0\,0\,1\,0\,0\,]$ is added to both codewords.

$$\mathbf{r}_1 = [\,0\,1\,1\,0\,0\,0\,1\,0\,] \qquad \mathbf{r}_2 = [\,0\,1\,0\,0\,1\,1\,1\,1\,]$$

$$\mathbf{s}_1 = [\,1\,0\,1\,1\,] \qquad \mathbf{s}_2 = [\,1\,0\,1\,1\,]$$

The syndromes are the same and equal column 6 of H, so decoder corrects bit 6.

$\mathcal{C}$ is an expanded Hamming code with weight enumerator $A(x) = 1 + 14x^4 + x^8$.

Standard array

Syndrome table decoding can also be described using the standard array.

The *standard array* of a group code $\mathcal{C}$ is the coset decomposition of F^n with respect to the subgroup $\mathcal{C}$.

0	c_2	c_3	$\cdots$	c_M
e_2	$c_2 + e_2$	$c_3 + e_2$	$\cdots$	$c_M + e_2$
e_3	$c_2 + e_3$	$c_3 + e_3$	$\cdots$	$c_M + e_3$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
e_N	$c_2 + e_N$	$c_3 + e_N$	$\cdots$	$c_M + e_N$

1. The first row is the code $\mathcal{C}$, with the zero vector in the first column.
2. Every other row is a coset.
3. The n-tuple in the first column of a row is called the *coset leader*

   We usually choose the coset leader to be the most plausible error pattern, e.g., the error pattern of smallest weight.

Standard array: decoding

An (n, k) LBC over $\mathrm{GF}(Q)$ has $M = Q^k$ codewords.

Every n-tuple appears exactly once in the standard array. Therefore the number of rows N satisfies

$$MN = Q^n \;\Rightarrow\; N = Q^{n-k}\,.$$

All vectors in a row of the standard array have the same syndrome.

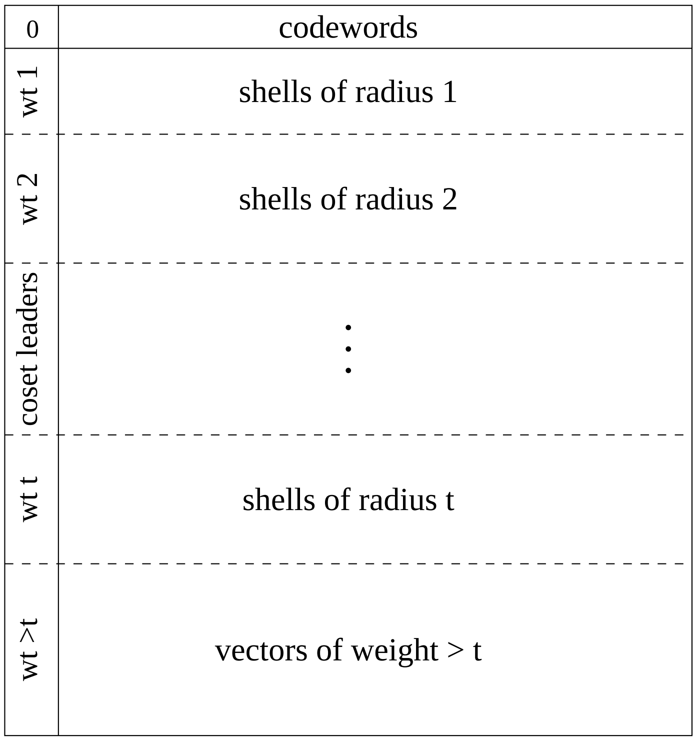
Thus there is a one-to-one correspondence between the rows of the standard array and the Q^{n-k} syndrome values.

Decoding using the standard array is simple: decode senseword $\mathbf{r}$ to the codeword at the top of the column that contains $\mathbf{r}$.

The *decoding region* for a codeword is the column headed by that codeword.

The decoder subtracts the coset leader from the received vector to obtain the estimated codeword.

Standard array and decoding regions



Standard array: example

The systematic generator and parity-check matrices for a $(6,3)$ LBC are

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

The standard array has six coset leaders of weight 1 and one of weight 2.

000000	001110	010101	011011	100011	101101	110110	111000
000001	001111	010100	011010	100010	101100	110111	111001
000010	001100	010111	011001	100001	101111	110100	111010
000100	001010	010001	011111	100111	101001	110010	111100
001000	000110	011101	010011	101011	100101	111110	110000
010000	011110	000101	001011	110011	111101	100110	101000
100000	101110	110101	111011	000011	001101	010110	011000
001001	000111	011100	010010	101010	100100	111111	110001

See http://www.stanford.edu/class/ee387/src/stdarray.pl for the short Perl script that generates the above standard array. This code is a *shortened* Hamming code.

Standard array: summary

The standard array is a conceptional arrangement of all n-tuples.

0	c_2	c_3	$\cdots$	c_M
e_2	$c_2 + e_2$	$c_3 + e_2$	$\cdots$	$c_M + e_2$
e_3	$c_2 + e_3$	$c_3 + e_3$	$\cdots$	$c_M + e_3$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
e_N	$c_2 + e_N$	$c_3 + e_N$	$\cdots$	$c_M + e_N$

- The first row is the code $\mathcal{C}$, with the zero vector in the first column.
- Every other row is a coset.
- The n-tuple in the first column of a row is called the *coset leader*.
- Senseword $\mathbf{r}$ is decoded to codeword at top of column that contains $\mathbf{r}$.
- The *decoding region* for codeword is column headed by that codeword.
- The decoder subtracts coset leader from $\mathbf{r}$ to obtain the estimated codeword.

Syndrome decoding: summary

Syndrome decoding is closely connected to standard array decoding.

1. Calculate syndrome $\mathbf{s} = \mathbf{r}H^T$ of received n-tuple.
2. Find most plausible error pattern $\mathbf{e}$ with $\mathbf{e}H^T = \mathbf{s}$.

   This error pattern is the coset leader of the coset containing $\mathbf{r}$.
3. Estimate transmitted codeword: $\hat{\mathbf{c}} = \mathbf{r} - \mathbf{e}$.

   The estimated codeword $\hat{\mathbf{c}}$ is the entry at the top of the column containing $\mathbf{r}$ in the standard array.
4. Determine message $\mathbf{m}$ from the encoding equation $\mathbf{c} = \mathbf{m}G$.

   In general, $\mathbf{m} = \mathbf{c}R$, where R is an $n \times k$ pseudoinverse of G. If the code is systematic, then $R = [\,0_{(n-k)\times k} \mid I_{k\times k}\,]^T$.

Only step 2 requires nonlinear operations and is the conceptually the most difficult.

Surprisingly, most computational effort is spent on syndrome computation.

Bounds on minimum distance

The minimum distance of a block code is a *conservative* measure of the quality of an error control code.

- A large minimum distance guarantees reliability against random errors.
- However, a code with small minimum distance *may* be reliable — provided the probability of sending codewords with nearby codewords is small.

We use minimum distance as the measure of a code's reliability because:

- A single number is easier to understand than a weight/distance distribution.
- The guaranteed error detection and correction ability are
  - detection: $e = d^* - 1$
  - correction: $t = \lfloor \frac{1}{2}(d^* - 1) \rfloor$
- Algebraic codes covered in the course are limited by minimum distance — these codes cannot correct more than t errors even if there is only one closest codeword.

Hamming (sphere-packing) bound

The Hamming bound for a (n,k) block code over Q-ary channel alphabet:

- A code corrects t errors iff spheres of radius t around codewords do not overlap.
- Therefore
$$Q^k = \text{number of codewords} \le \frac{\text{volume of space}}{\text{volume of sphere of radius } t} = \frac{Q^n}{V(Q,n,t)},$$
where $V(Q,n,t)$ is the "volume" (number of elements) of a sphere of radius t in Hamming space of n-tuples over a channel alphabet with Q symbols:
$$V(Q,n,t) = 1 + \binom{n}{1}(Q-1) + \binom{n}{2}(Q-1)^2 + \cdots + \binom{n}{t}(Q-1)^t$$
- Rearranging the inequality gives a lower bound on $n-k$,
$$Q^{n-k} \ \ge\ V(Q,n,t) \ \Rightarrow\ n-k \ \ge\ \log_Q V(Q,n,t)\,,$$
and thus an upper bound on rate R,
$$R \le 1 - \frac{1}{n}\log_Q\left(1 + \binom{n}{1}(Q-1) + \binom{n}{2}(Q-1)^2 + \cdots + \binom{n}{t}(Q-1)^t\right).$$

Hamming bound: example

A wireless data packet contains 192 audio samples, 16 bits for two channels. The number of the information bits is $192 \cdot 2 \cdot 16 = 6144$.

The communications link is a binary symmetric channel with raw error rate 10^{-3}. How many check bits are needed for reliable communication?

t	$n-k$	n	Rate	$\mathrm{P}\{> t$ errors$\}$
10	105	6249	0.983	5.4×10^{-02}
12	123	6267	0.980	1.2×10^{-02}
14	141	6285	0.978	2.2×10^{-03}
16	158	6302	0.975	3.0×10^{-04}
18	175	6319	0.972	3.5×10^{-05}
20	192	6336	0.970	3.3×10^{-06}
22	208	6352	0.967	2.6×10^{-07}
24	225	6369	0.965	1.8×10^{-08}
26	241	6385	0.962	1.1×10^{-09}
28	257	6401	0.960	5.5×10^{-11}
30	272	6416	0.958	2.5×10^{-12}
32	288	6432	0.955	1.0×10^{-13}

The Hamming bound shows that more than 4% redundancy is needed to achieve a reasonable bit error rate.

Other bounds on minimum distance

The following bounds show tradeoffs between rate R and minimum distance d^*.

- McEliece-Rodemich-Rumsey-Welch (MRRW) upper bound.

  $$R \le H\left(\tfrac{1}{2} - \sqrt{\delta(1-\delta)}\right).$$

  H is binary entropy function and $\delta = d^*/n$ is *normalized* minimum distance

- Plotkin upper bound for binary linear block codes (homework exercise):

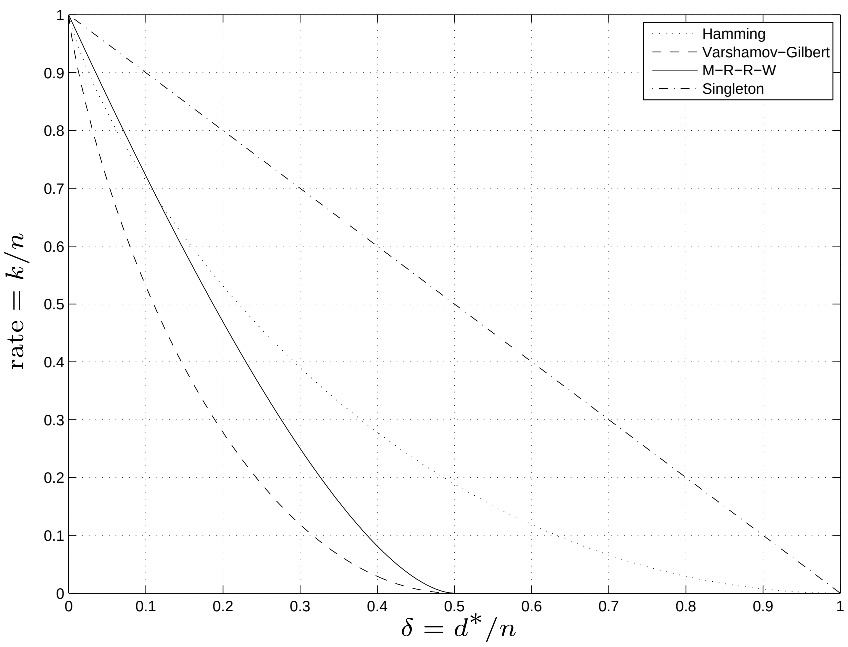
  $$d^* \le \frac{n \cdot 2^{k-1}}{2^k - 1} \Rightarrow \delta = \frac{d^*}{n} \le \tfrac{1}{2} \text{ for large } k.$$

- Varshamov-Gilbert *lower* bound for binary block codes. If $d^* < n/2$ there then *exists* a code with minimum distance d^* and rate R satisfying

  $$R \ge 1 - \log_2\left(\sum_{i=0}^{d-1}\binom{n}{i}\right) \approx 1 - H(d^*/n) = 1 - H(\delta).$$

  For comparison, the Hamming bound is $R \le 1 - H(\delta/2)$.

Plots of rate vs. normalized minimum distance



The MRRW bound is stronger than the Hamming bound except for high rates.

The Hamming bound is fairly tight for high rates. For example, to correct 10 errors in 1000 bits, the Hamming bound requires 78 check bits, but there exists a BCH code that has 100 check bits.

Perfect codes

Definition: A block code is called *perfect* if every senseword is within distance t of exactly one codeword.

Other definitions of perfect codes:

- decoding spheres pack perfectly
- have complete bounded-distance decoders
- satisfy the Hamming bound with equality

There are only finitely many classes of perfect codes:

- Codes with no redundancy ($k = n$)
- Repetition codes with odd blocklength: $n = 2m + 1,\ k = 2m,\ t = m$
- Binary Hamming codes: $n = 2^m - 1,\ n - k = m$
- Nonbinary Hamming codes: $n = (q^m - 1)/(q - 1),\ n - k = m,\ q > 2$
- Binary Golay code: $q = 2,\ n = 23,\ k = 12,\ t = 3$
- Ternary Golay code: $q = 3,\ n = 11,\ k = 6,\ t = 2$

Golay discovered the perfect Golay codes in 1949 — a very good year for Golay.

Quasi-perfect codes

Definition: A code is *quasi-perfect* if every n-tuple is

- within distance t of *at most* one codeword, and is
- within distance $t + 1$ of *at least* one codeword.

Equivalently, a code is quasi-perfect if spheres of radius t surrounding codewords do not overlap, while spheres of radius $t + 1$ cover the space of n-tuples.

Examples of quasi-perfect codes:

- Repetition codes with even blocklength
- Expanded Hamming and Golay codes with overall parity-check bit

Exercise: Show that expurgated Hamming codes (obtained by adding an overall parity-check equation) are *not* quasi-perfect.

Modified linear codes

The *design blocklength* of a linear block code is determined by algebraic and combinatorial properties of matrices or polynomials.

The *desired blocklength* of a linear block code is often different from the design blocklength.

Example:

- Design blocklength of binary Hamming code is $2^m - 1$ $(7, 15, 31, \ldots)$
- Number of information symbols may not be $k = 2^m - 1 - m$ $(4, 11, 26, \ldots)$

There are six ways to modify parameters of a linear block code (n, k, $n-k$) by increasing one, decreasing another, and leaving the third unchanged.

The most common modification is to *shorten* the code by dropping information symbols.

The other modifications are *lengthen*, *expurgate*, *augment*, *puncture*, *expand*.

Shortened codes

Shorten: Fix $n-k$, decrease k and therefore n.

Information symbols are deleted to obtain a desired blocklength smaller than the *design blocklength*.

The missing information symbols are usually imagined to be at the beginning of the codeword and are considered to be 0.

Example: Ethernet frames are variable-length packets. Maximum packet size is about 1500 data octets or 12000 bits.

The 32-bit ethernet checksum comes from a Hamming code with design blocklength $2^{32} - 1 = 4294967295$ bits, or 536870907 octets.

Encoder/decoder cost can be reduced by deleting carefully chosen symbols.

Shortened code example

The systematic parity-check matrix for a $(15,11)$ binary Hamming code is

$$H = \left[\begin{array}{cccc|ccccccccccc} 1&0&0&0&1&0&0&1&1&0&1&0&1&1&1 \\ 0&1&0&0&1&1&0&1&0&1&1&1&1&0&0 \\ 0&0&1&0&0&1&1&0&1&0&1&1&1&1&0 \\ 0&0&0&1&0&0&1&1&0&1&0&1&1&1&1 \end{array}\right].$$

We can shorten to $(12,8)$ code by deleting maximum-weight columns 12 to 14:

$$H' = \left[\begin{array}{cccc|cccccccc} 1&0&0&0&1&0&0&1&1&0&1&1 \\ 0&1&0&0&1&1&0&1&0&1&1&0 \\ 0&0&1&0&0&1&1&0&1&0&1&0 \\ 0&0&0&1&0&0&1&1&0&1&0&1 \end{array}\right].$$

The shortened code can correct single bit errors in an 8-bit data byte.

Each check equation is the exclusive-or of 5 or 6 input bits, compared to 8 inputs in original code.

Lengthened codes

Lengthen: Fix $n-k$, increase k and therefore n.

New information symbols are introduced and included in check equations.

Usually difficult to do without reducing the minimum distance of the code.

Example: Extended Reed-Solomon codes, obtained by lengthening the $(Q-1,k)$ R-S codes to $(Q+1,k+2)$ by adding two columns at the left of H:

$$H = \begin{bmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{Q-2} \\ 1 & \alpha^2 & \alpha^4 & \cdots & \alpha^{2(Q-2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^d & \alpha^{2d} & \cdots & \alpha^{d(Q-2)} \end{bmatrix} \Rightarrow$$

$$H' = \begin{bmatrix} 1 & 0 & 1 & \alpha & \alpha^2 & \cdots & \alpha^{Q-2} \\ 0 & 0 & 1 & \alpha^2 & \alpha^4 & \cdots & \alpha^{2(Q-2)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & 1 & \alpha^d & \alpha^{2d} & \cdots & \alpha^{d(Q-2)} \end{bmatrix}$$

Expurgated codes

Expurgate: Fix n, decrease k and increase $n-k$.

Codewords are deleted by adding check equations, reducing the dimension of the code. The goal is to increase error protecting ability

Example: The $(7,3)$ expurgated Hamming code.

Example: (15,7) double error correcting binary BCH code is obtained from the (15,11) Hamming code by adding four more rows to H:

$$H^+ = \begin{bmatrix} 1&0&0&0&1&1&0&0&0&1&1&0&0&0&1 \\ 0&0&0&1&1&0&0&0&1&1&0&0&0&1&1 \\ 0&0&1&0&1&0&0&1&0&1&0&0&1&0&1 \\ 0&1&1&1&1&0&1&1&1&1&0&1&1&1&1 \end{bmatrix}$$

The combined parity-check matrix is (α is a primitive element in $\mathrm{GF}(2^4)$):

$$\begin{bmatrix} H \\ H^+ \end{bmatrix} = \begin{bmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \cdots & \alpha^{12} & \alpha^{13} & \alpha^{14} \\ 1 & \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} & \cdots & \alpha^{36} & \alpha^{39} & \alpha^{42} \end{bmatrix}$$

We will see that no 4 columns are linearly dependent over $\mathrm{GF}(2)$, so $d^* \geq 5$.

Augmented codes

Augment: Fix n, increase k and decrease $n-k$.

Add codewords by adding new basis vectors — new rows of generator matrix.

This increases rate of code while possibly decreasing the minimum distance.

Example: The generator matrix of Reed-Muller code $\mathcal{R}(r,m)$ is defined by augmentation:

$$G = \begin{bmatrix} G_0 \\ G_1 \\ \vdots \\ G_r \end{bmatrix}$$

Submatrix G_i has $\binom{m}{i}$ rows and $n = 2^m$ columns. Number of information bits is

$$k \;=\; \binom{m}{0} + \binom{m}{1} + \cdots + \binom{m}{r}$$

It can be shown that the minimum weight is 2^{m-r}.

The Reed-Muller codes have a wide range of minimum distances and corresponding rates. The rate $1/2$ codes have $d^* = \sqrt{n}$, which is essentially optimal.

Expanded codes

Expand: Fix k, increase $n - k$ and n.

Add new check symbols and corresponding equations.

Example: The extended Hamming code is obtained by adding an overall parity-check bit, thereby increasing the minimum distance from 3 to 4.

Fact: When the minimum distance of a *binary* linear block code is odd, overall parity-check bit increases the miminum distance by 1 to the next even number.

Example: The binary Golay code is a $(23, 12)$ code with minimum distance 7 — a perfect three-error correcting code.

Thus an overall parity-check equation increases the minimum distance to 8.

The extended Golay code, with parameters $(24, 12, 8)$, was used for error protection in the Voyager I and II spacecraft.

Robert Gallager's tribute: Marcel Golay's one-page paper, "Notes on Digital Coding" (Proc. IRE, vol. 37, p. 657, 1949) is surely the most remarkable paper on coding theory ever written. Not only did it present the two perfect "Golay codes", the $(n = 23, k = 12, d = 7)$ binary code and the $(n = 11, k = 6, d = 5)$ ternary code, but it also gave the non-binary generalization of the perfect binary Hamming codes and the first publication of a parity-check matrix.

Punctured codes

Puncture: Fix k, decrease $n - k$ and therefore n.

Deleting check symbols may reduce minimum distance.

However, punctured codes may correct the large majority of errors up to the minimum distance of the original code.

Puncturing may reduce minimum distance but not significantly reduce reliability.

Punctured codes may be obtained from simple codes that have too much redundancy.

Example: We can puncture a $(9, 4)$ simple product code with $d^* = 4$ to obtain a $(8, 4)$ code with $d^* = 3$. If we expand the punctured code by adding an overall parity-check bit, we recover the simple product code.

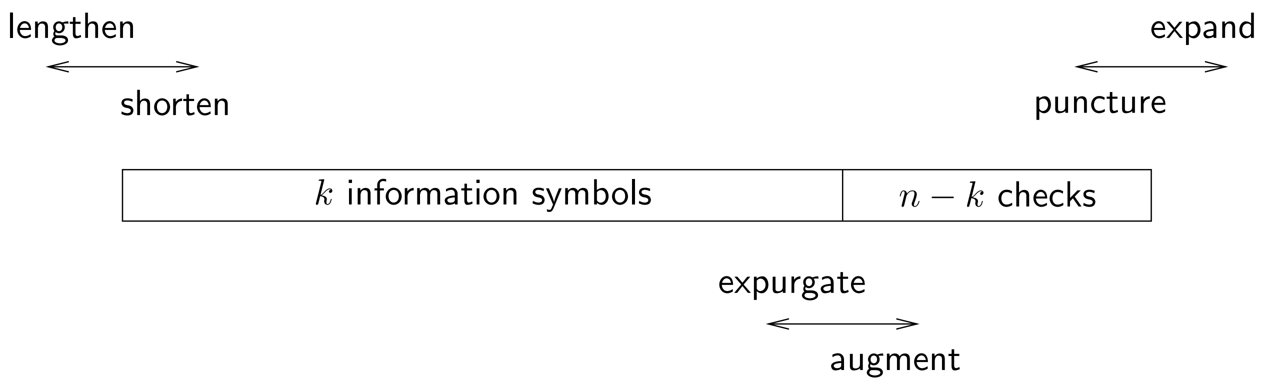
Soft-decision decoders or error-and-erasure decoders can treat the missing check symbols as unreliable.

Linear block code modifications: summary

Change any two of the block code parameters $n,\ k,\ ,n-k$:

- *Shorten*: delete message symbols:
  $n-k$ fixed, $k\downarrow\ \Rightarrow\ n\downarrow$
- *Lengthen*: add message symbols:
  $n-k$ fixed, $k\uparrow\ \Rightarrow\ n\uparrow$
- *Puncture*: delete check symbols:
  k fixed, $n-k\downarrow\ \Rightarrow\ n\downarrow$
- *Extend (expand)*: add check symbols:
  k fixed, $n-k\uparrow\ \Rightarrow\ n\uparrow$
- *Expurgate*: delete codewords, add check equations:
  n fixed, $k\downarrow\ \Rightarrow\ n-k\uparrow$
- *Augment*: add codewords, delete check equations:
  n fixed, $k\uparrow\ \Rightarrow\ n-k\downarrow$

Linear block code modifications: picture


lengthen
shorten
expand
puncture
k information symbols
$n-k$ checks
expurgate
augment