2.0 Block Codes

2.1 Definitions and Examples

Definition 1 *A* **block code** $\mathcal{C}$ *is a set of* M $n-$*tuples drawn from some specified alphabet.*[a]

□

- Each codeword represents $\log_2 M$ bits of information.

Definition 2 *The rate* R *of code* $\mathcal{C}$ *over an alphabet of size* q *is*

$$R = \frac{\log_2 M}{n}.$$

□

- R is expressed in bits/symbol.

[a] The alphabet will be defined more precisely later.

2.2 Characterization of Errors

Abstract channel model:

- Codeword $\mathbf{c} = (c_0, c_1, \ldots, c_{n-1})$ is transmitted over a noisy channel.
- $\mathbf{y} = (y_0, y_1, \ldots, y_{n-1})$ is received.

$$\mathbf{y} = \mathbf{c} + \mathbf{e}$$

- "$+$" is defined in the *symbol alphabet*.
- $\mathbf{e} = (e_0, \ldots, e_{n-1})$ is the *error pattern* or *error vector*.
- *Error detection:* did any errors occur?
- *Error correction:* where are the errors; what are their values?

2.3 Weights and Distances

- We need a measure of *distance* or *difference* between codewords.
- Properties of distance measures.
  - $d(\mathbf{x}, \mathbf{y}) \geq 0$.
  - $d(\mathbf{x}, \mathbf{y}) = 0 \iff \mathbf{x} = \mathbf{y}$.
  - $d(\mathbf{x}, \mathbf{z}) + d(\mathbf{z}, \mathbf{y}) \geq d(\mathbf{x}, \mathbf{y})$ (triangle inequality).
  - $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$.

Definition 3 *The* **Hamming distance** *between two vectors of the same length is the number of positions in which they differ.* $\square$

We will also need the following.

Definition 4 *The* **Hamming weight** $w_H(\mathbf{v})$ *of an* $n-$*tuple is the number of nonzero components in the vector.* $\square$

2.4 Decoding

2.4.1 Distance Measures and Error Correction

Definition 5 *The* **minimum distance** *of a code is*

$$d_{min} = \min_{c_i \neq c_j \in \mathcal{C}} d_H(c_i, c_j)$$

$\square$

Let us return to our example:

$$\mathbf{y} = \mathbf{c} + \mathbf{e}$$

and let

$$w_H(\mathbf{e}) = t'$$

i.e., t' errors have occurred in the transmission of $\mathbf{c}$.

Definition 6 *The process of estimating* $\mathbf{c}$ *(equivalent to finding* $\mathbf{e}$*) from* $\mathbf{y}$ *is called* **decoding**.

□

Suppose decoder uses a *minimum distance decoding rule*:

$$\hat{\mathbf{c}} = \arg\min_{\mathbf{c}\in\mathcal{C}} d_H(\mathbf{y}, \mathbf{c}).$$

Then, $t < d_{min}/2 \;\Rightarrow\; \hat{\mathbf{c}}$ is the transmitted word.

Note: "Decoding" includes the process of *error correction*.

Formally...

Theorem 1 *A code with minimum distance* $d_{min} = 2t + 1$ *can, with suitable decoding, correct any error pattern* $\mathbf{e}$ *if*

$$w_H(\mathbf{e}) \le t$$

where

$$t = \left\lfloor \frac{d_{min} - 1}{2} \right\rfloor$$

Proof:

- Construct sphere of "radius" $(d_{min} - 1)/2$ about every codeword.
- These **nonoverlapping** spheres are **decoding regions** of $\mathcal{C}$.
  - Suppose $\mathbf{y}$ $\in$ a sphere about word $\mathbf{c}_i$.
    * Then $d_H(\mathbf{y}, \mathbf{c}_i) \leq t$.
    * But $d_h(\mathbf{c}_i, \mathbf{c}_j) > 2t$ for every $j \neq i$ such that $c_i \in \mathcal{C}$.
  - So, $\mathbf{y}$ is nearer to $\mathbf{c}_i$ than to any other codeword. (see below).

$$\mathbf{c}_i \quad < - - - - > \quad \mathbf{y} \quad < - - - - - - - - - - - > \quad \mathbf{c}_j$$

$$\leq t \qquad\qquad\qquad\qquad \geq t + 1$$

- Hence, every other codeword is farther from $\mathbf{y}$ than $\mathbf{c}_i$

$\square$

How do we use distance measures? (See also Appendix 2-A.)

Definition 7 *A channel with input symbols from an M-ary alphabet and output symbols from a Q-ary alphabet, where M and Q are finite integers is said to be a* **discrete** *channel.* □

Definition 8 *A discrete channel whose output during a symbol interval is determined only by the input symbol during that interval (and on no previous symbol) is called a* **discrete memoryless channel** *(DMC).* □

The *binary symmetric channel* (BSC) is a special case of the DMC.

- On the BSC with error prob. $p < 1/2$,

$$(1-p)^n > p \cdot (1-p)^{n-1} > p^2 \cdot (1-p)^{n-2} > \cdots > p^n$$

so

- receiving the block with no errors is more likely than receiving of any other block;
- receiving a block with one error is more likely than receiving a block with two (or more) errors;
- etc.

Thus, the best strategy is to decode into the codeword that is *closest* to the received word.

Exercise:

For a block length of $n = 7$, for what values of p does the probability of receiving an n-tuple correctly exceed the probability of receiving the n-tuple with a single error?

(*Hint:* the probability of j errors in an n-tuple is given by the *binomial probability distribution*.

2.4.2 Decoder Performance Measures

Definition 9 *The event that the decoder chooses other than the transmitted codeword is called a* **decoding error**. □

Definition 10 *The event that the decoder is unable to choose any codeword is called a* **decoding failure.** □

Definition 11 *A decoder which finds the codeword* **nearest** *the received vector is called a* **complete** *(or* **nearest neighbor***) decoder.* □

$$\hat{\mathbf{c}}_i = \arg\min_{\mathbf{c}\in\mathcal{C}} \; d(\mathbf{y}, \mathbf{c})$$

Definition 12 *A decoder which decodes correctly* **only** *when $t' \leq t$ called a* **bounded distance** *decoder (BDD).* □

i.e.,

$$\hat{\mathbf{c}}_i = \arg\min_{\mathbf{c}} \ d(\mathbf{y}, \mathbf{c})$$

only if $d(\mathbf{y}, \mathbf{c}) \leq t$ where $d_{min} \geq 2t + 1$.

For a BDD,

- if $d_H(\mathbf{y}, \mathbf{c}_j) \leq t$ where $\mathbf{c}_j$ is **not** the transmitted codeword, the decoder outputs an incorrect word and suffers a *decoding error*.

- if $d_H(\mathbf{y}, \mathbf{c}) > t, \ \forall \mathbf{c} \in \mathcal{C}$, the BDD can make no selection and suffers a *decoding failure*.

2.4.3 Optimal Decoders

One must define the criterion for optimality before identifying the characteristics of an "optimal" decoder.

Let $\Pr(y_i|c_i)$ be the (conditional) probability that the DMC output symbol is y_i, given that the input symbol is c_i.

Lemma: *The conditional probability distribution of the channel output word is given by*

$$P(\mathbf{y}|\mathbf{c}) = \prod_{i=0}^{n-1} \Pr(y_i|c_i)$$

$\square$

Proof: Exercise.

Definition 13 *The* **maximum likelihood decoder** *produces codeword* $\hat{\mathbf{c}}$ *given by*

$$\hat{\mathbf{c}} = \arg\max_{\mathbf{c}\in\mathcal{C}} P(\mathbf{y}|\mathbf{c}).$$

$\square$

Now we apply Bayes's rule to compute

$$P(\mathbf{c}|\mathbf{y}) = \frac{P(\mathbf{y}|\mathbf{c})p(\mathbf{c})}{p(\mathbf{y})}$$

where $p(\mathbf{c})$ is the **prior probability** of codeword $\mathbf{c}$ and $p(\mathbf{y})$ is the unconditional probability of channel output $\mathbf{y}$. This gives us

Definition 14 *The* **maximum a posteriori (MAP)** *decoder is given by*

$$\hat{\mathbf{c}} = \arg\max_{\mathbf{c} \in \mathcal{C}} P(\mathbf{c}|\mathbf{y})$$

$\square$

Lemma: *The MAP and ML decoders are identical for the DMC when codewords are equiprobable.* $\square$

2.5 Some Useful Bounds on Block Codes

2.5.1 The Hamming Bound

- Consider $n-$tuples as points in $n - space$.
  - This is a *discrete* space.
  - Distance measure is d_H.

- Place codeword $\mathbf{c}_1$ at *center* of "sphere" of radius $t = \lfloor (d-1)/2 \rfloor$.

- If $\mathbf{y}$ $(channeloutput)$ $\in$ the sphere, then $\mathbf{y}$ is decoded as $\mathbf{c}_1$ and
  1. fewer than t errors occurred, decoding is correct, or
  2. more than t errors occurred, and the decoder output is incorrect.

- d_{min} constraint: max number of spheres in n-space separated by at least d_{min} is the max number M of codewords.
- *volume* (number of points) of sphere is found by summing:

$$
\begin{array}{rl}
1 & \text{@ center} \\
n(q-1) & \text{@ } d = 1 \text{ from center} \\
\binom{n}{2}(q-1)^2 & \text{@ } d = 2 \text{ from center} \\
\vdots & \\
\binom{n}{t}(q-1)^t & \text{@ } d = t \text{ from center.}
\end{array}
$$

We sum these to get the total volume occupied by code words.

$$V_q(n,t) = \sum_{j=0}^{t} \binom{n}{j} (q-1)^j$$

- Number of points in the space $= q^n$.
- If there are M spheres (codewords),

$$\begin{aligned} M \cdot V_q(n,t) &\leq q^n \\ \log_q M + \log_q V_q(n,t) &\leq n \\ n - \log_q M &\geq \log_q V_q(n,t) \end{aligned}$$

- Let $r = n - \log_q M =$ the block code *redundancy* (Why?). Then

$$r \geq \log_q V_q(n,t)$$

- This is the *Hamming lower bound* on r for *any* block code.

2.5.2 The Gilbert Bound

A random code design method:

1. Randomly select the first codeword $\mathbf{c}_1$.
2. Delete all $\mathbf{x}$ $s.t.$ $d(\mathbf{c}_1, \mathbf{x}) \leq 2t$ (as many as $V_q(n, 2t)$ points.)
3. Select a remaining point and repeat.
4. Stop when points are exhausted.

By this procedure, M codewords have been chosen, where

$$\begin{aligned} M &= \left\lceil \frac{q^n}{V_q(n, 2t)} \right\rceil \\ &\geq \frac{q^n}{V_q(n, 2t)} \end{aligned}$$

Taking logs and rearranging gives

$$r \leq \log_q V_q(n, 2t)$$

This is the *Gilbert Bound.* Note that, for spheres of radius $2t$, the Hamming bound gives a lower bound of $\log_q V_q(n, 2t) \leq r$. However, this lower bound is subsumed by that for smaller radius:

$$\log_q V_q(n, t) \leq r \leq \log_q V_q(n, 2t),$$

where

- the first inequality is a *bound*;
- the second inequality shows *existence*.

2.5.3 Perfect Codes

Definition 15 *A* **perfect code** *is one that satisfies the Hamming bound with equality.* □

$$r = \log_q \sum_{j=0}^{t} \binom{n}{j}$$

Thus, every point in the space is within distance $(d_{min} - 1)/2$ of a code word and within a sphere.

Definition 16 *In a* **quasi-perfect** *code, all points not in a sphere about a codeword lie at distance $t + 1$ from at least one codeword.* □

2.5.4 Varsharmov-Gilbert Bound

Theorem 2 *For each R, $d(R) \geq \delta$ for all δ that satisfy*

$$R \geq 1 - H_q(\delta)$$

where H_q is the entropy function,

$$H_q(x) = x \log_q(q-1) - x \log_q x - (1-x) \log_q(1-x)$$

and

$$\begin{aligned} d(R) &= \lim_{n \to \infty} \frac{1}{n} d(n, R) \\ d(n, R) &= \max_{\mathcal{C}} d_{min}(\mathcal{C}) \end{aligned}$$