E2 205: Error-Control Coding
Chapter 2: Block Codes

Navin Kashyap

Indian Institute of Science

- The coding schemes we have seen so far divide the message to be transmitted into fixed-length blocks, each of which is encoded using a codeword of fixed length n

- This is in contrast to coding schemes, such as convolutional codes, which encode a message in a sequential fashion.

Block Codes

Code alphabet: A finite set of q symbols, denoted by $\mathbb{F}$ or $\mathbb{F}_q$.

(Later, we will endow $\mathbb{F}$ with an algebraic structure.)

Notation: $\mathbb{F}^n = \{(x_1, x_2, \ldots, x_n) : x_i \in \mathbb{F}\}$.

Clearly, $|\mathbb{F}^n| = |\mathbb{F}|^n = q^n$.

Definition: An (n, M) block code (or simply, code) is a non-empty subset $\mathcal{C} \subseteq \mathbb{F}^n$, with $|C| = M$.

- The elements of $\mathcal{C}$ are called codewords
- n is the blocklength, or simply length, of the code.
- The rate of the code is $R = \frac{1}{n} \log_q M$.

Examples:

- The 3-fold repetition code $\mathcal{C} = \{000, 111\}$ is a $(3, 2)$ block code over $\mathbb{F}_2$.
- The length-7 Hamming code is a $(7, 16)$ block code over $\mathbb{F}_2$.

Probabilistic Channel Models and Decoding

A (discrete) probabilistic channel is a triple $(\mathbb{F}, \mathcal{Y}, \Pr)$, where

- $\mathbb{F}$ is a finite input alphabet
- $\mathcal{Y}$ is a finite output alphabet
- $\Pr$ is the channel transition probability function

$$\Pr[\mathbf{y} \text{ received} \mid \mathbf{x} \text{ transmitted}]$$

defined for each pair $(\mathbf{x}, \mathbf{y}) \in \mathbb{F}^m \times \mathcal{Y}^m$, as m ranges over the positive integers.

Probabilistic Channel Models and Decoding

A (discrete) probabilistic channel is a triple $(\mathbb{F}, \mathcal{Y}, \mathsf{Pr})$, where

- $\mathbb{F}$ is a finite input alphabet
- $\mathcal{Y}$ is a finite output alphabet
- Pr is the channel transition probability function

$$\mathsf{Pr}[\mathbf{y} \text{ received} \mid \mathbf{x} \text{ transmitted}]$$

  defined for each pair $(\mathbf{x}, \mathbf{y}) \in \mathbb{F}^m \times \mathcal{Y}^m$, as m ranges over the positive integers.

Let $\mathcal{C}$ be an (n, M) code over the alphabet $\mathbb{F}$.

A decoder for $\mathcal{C}$ with respect to the channel $(\mathbb{F}, \mathcal{Y}, \mathsf{Pr})$ is a function

$$\mathcal{D} : \mathcal{Y}^n \longrightarrow \mathcal{C}$$

Probability of Decoding Error

Let $\mathcal{C}$ be a block code with decoder $\mathcal{D}$ wrt the channel $(\mathbb{F}, \mathcal{Y}, \mathrm{Pr})$.

- For $\mathbf{c} \in \mathcal{C}$, define

$$P_{\mathrm{err}}(\mathbf{c}) = \sum_{\mathbf{y}:\mathcal{D}(\mathbf{y}) \neq \mathbf{c}} \mathrm{Pr}[\mathbf{y} \text{ received} \mid \mathbf{c} \text{ transmitted}].$$

  This is the probability of decoding error, given that **c** was transmitted, when using decoder $\mathcal{D}$

- Worst-case prob. of decoding error, when using decoder $\mathcal{D}$:

$$P_{\mathrm{err}} = \max_{\mathbf{c} \in \mathcal{C}} P_{\mathrm{err}}(\mathbf{c})$$

- Average prob. of decoding error, when using decoder $\mathcal{D}$:

$$\overline{P}_{\mathrm{err}} = \sum_{\mathbf{c} \in \mathcal{C}} P_{\mathrm{err}}(\mathbf{c}) \underbrace{P(\mathbf{c} \text{ transmitted})}_{\textit{a priori} \text{ prob. of transmitting } \mathbf{c}}$$

Average Prob. of Decoding Error

$$\overline{P}_{\mathsf{err}} \;=\; \sum_{\mathbf{c}\in\mathcal{C}} P_{\mathsf{err}}(\mathbf{c})\, P(\mathbf{c}\ \text{transmitted})$$

- Often, we do not know the probabilities $P(\mathbf{c}\ \text{transmitted})$; so we assume that all M codewords $\mathbf{c} \in \mathcal{C}$ are equally likely to be transmitted. In this case:

$$\overline{P}_{\mathsf{err}} \;=\; \frac{1}{M} \sum_{\mathbf{c}\in\mathcal{C}} P_{\mathsf{err}}(\mathbf{c}).$$

Average Prob. of Decoding Error

$$\overline{P}_{\mathrm{err}} = \sum_{\mathbf{c}\in\mathcal{C}} P_{\mathrm{err}}(\mathbf{c})\, P(\mathbf{c} \text{ transmitted})$$

An alternate form:

$$\begin{aligned}
\overline{P}_{\mathrm{err}} &= \sum_{\mathbf{c}\in\mathcal{C}} \sum_{y:\mathcal{D}(\mathbf{y})\neq\mathbf{c}} \Pr[\mathbf{y} \mid \mathbf{c}]\, P(\mathbf{c}) \\
&= \sum_{\mathbf{y}} \sum_{\mathbf{c}\in\mathcal{C}:\mathcal{D}(\mathbf{y})\neq\mathbf{c}} \Pr[\mathbf{y} \mid \mathbf{c}]\, P(\mathbf{c}) \\
&= \sum_{\mathbf{y}} \sum_{\mathbf{c}\in\mathcal{C}:\mathcal{D}(\mathbf{y})\neq\mathbf{c}} P(\mathbf{c} \mid \mathbf{y}) P(\mathbf{y}) \\
&= \sum_{\mathbf{y}} P(\mathbf{y}) \underbrace{\sum_{\mathbf{c}\in\mathcal{C}:\mathcal{D}(\mathbf{y})\neq\mathbf{c}} P(\mathbf{c} \mid \mathbf{y})}_{\text{prob. of decoding error, given } \mathbf{y} \text{ rcvd.}}
\end{aligned}$$

MAP Decoding

Consider the prob. of decoding error given $\mathbf{y}$ received, when using decoder $\mathcal{D}$: Assuming that $\mathcal{D}(\mathbf{y}) = \hat{\mathbf{c}}$, we have

$$\begin{aligned}
\sum_{\mathbf{c}\in\mathcal{C}:\mathcal{D}(\mathbf{y})\neq\mathbf{c}} P(\mathbf{c} \mid \mathbf{y}) &= \sum_{\mathbf{c}\in\mathcal{C}:\mathbf{c}\neq\hat{\mathbf{c}}} P(\mathbf{c} \mid \mathbf{y}) \\
&= \underbrace{\sum_{\mathbf{c}\in\mathcal{C}} P(\mathbf{c} \mid \mathbf{y})}_{\text{does not depend on } \mathcal{D}} - \underbrace{P(\hat{\mathbf{c}} \mid \mathbf{y})}_{\text{max when } \mathcal{D} \text{ is MAP rule}}
\end{aligned}$$

Maximum A-Posteriori Probability (MAP) Decoding Rule:

> *Given a received word* $\mathbf{y}$, *decode to a codeword* $\hat{\mathbf{c}} \in \mathcal{C}$ *that maximizes* $P(\mathbf{c} \textit{ transmitted} \mid \mathbf{y} \textit{ received})$.
> *(Ties are broken arbitrarily.)*

MAP Decoding

Maximum A-Posteriori Probability (MAP) Decoding Rule:

Given a received word $\mathbf{y}$*, decode to a codeword* $\hat{\mathbf{c}} \in \mathcal{C}$ *that maximizes* $P(\mathbf{c} \text{ transmitted} \mid \mathbf{y} \text{ received})$.
(Ties are broken arbitrarily.)

- For each $\mathbf{y}$, the MAP decoder minimizes, among all decoders $\mathcal{D}$, the prob. of decoding error given $\mathbf{y}$ received.

$$\Downarrow$$

$$\boxed{\text{The MAP decoder minimizes } \overline{P}_{\text{err}} \text{ among all decoders } \mathcal{D}}$$

ML Decoding

- Given a received word $\mathbf{y}$, MAP rule requires finding

$$\begin{aligned}\arg\max_{\mathbf{c}\in\mathcal{C}} P(\mathbf{c} \mid \mathbf{y}) &= \arg\max_{\mathbf{c}\in\mathcal{C}} \frac{\Pr[\mathbf{y} \mid \mathbf{c}]\, P(\mathbf{c})}{P(\mathbf{y})} \\ &= \arg\max_{\mathbf{c}\in\mathcal{C}} \Pr[\mathbf{y} \mid \mathbf{c}]\, P(\mathbf{c})\end{aligned}$$

- Since the prior probabilities $P(\mathbf{c})$ are typically unknown, we usually make the "equally likely codewords" assumption: $P(\mathbf{c}) = \frac{1}{M}\ \forall\, \mathbf{c} \in \mathcal{C}$.

ML Decoding

- Given a received word $\mathbf{y}$, MAP rule requires finding

$$\begin{aligned}\arg\max_{\mathbf{c}\in\mathcal{C}} P(\mathbf{c} \mid \mathbf{y}) &= \arg\max_{\mathbf{c}\in\mathcal{C}} \frac{\Pr[\mathbf{y} \mid \mathbf{c}]\, P(\mathbf{c})}{P(\mathbf{y})} \\ &= \arg\max_{\mathbf{c}\in\mathcal{C}} \Pr[\mathbf{y} \mid \mathbf{c}]\, P(\mathbf{c})\end{aligned}$$

- Since the prior probabilities $P(\mathbf{c})$ are typically unknown, we usually make the "equally likely codewords" assumption: $P(\mathbf{c}) = \frac{1}{M}\ \forall\, \mathbf{c} \in \mathcal{C}$.

- With this, MAP reduces to

  Maximum Likelihood (ML) Decoding Rule:

  > *Given a received word* $\mathbf{y}$, *decode to a codeword* $\mathbf{c} \in \mathcal{C}$ *that maximizes* $\Pr[\mathbf{y} \mid \mathbf{c}]$. *(Ties are broken arbitrarily.)*

ML Decoding

- Given a received word $\mathbf{y}$, MAP rule requires finding

$$\begin{aligned}\arg\max_{\mathbf{c}\in\mathcal{C}} P(\mathbf{c} \mid \mathbf{y}) &= \arg\max_{\mathbf{c}\in\mathcal{C}} \frac{\Pr[\mathbf{y} \mid \mathbf{c}]\, P(\mathbf{c})}{P(\mathbf{y})} \\ &= \arg\max_{\mathbf{c}\in\mathcal{C}} \Pr[\mathbf{y} \mid \mathbf{c}]\, P(\mathbf{c})\end{aligned}$$

- Since the prior probabilities $P(\mathbf{c})$ are typically unknown, we usually make the "equally likely codewords" assumption: $P(\mathbf{c}) = \frac{1}{M}\ \forall\, \mathbf{c} \in \mathcal{C}$.

- With this, MAP reduces to

  Maximum Likelihood (ML) Decoding Rule:

  > *Given a received word* $\mathbf{y}$, *decode to a codeword* $\mathbf{c} \in \mathcal{C}$ *that maximizes* $\Pr[\mathbf{y} \mid \mathbf{c}]$. *(Ties are broken arbitrarily.)*

- Thus, under the "equally likely codewords" assumption, the ML decoder minimizes $\overline{P}_{\text{err}}$ among all decoders $\mathcal{D}$.

Special Case: BSC(p)

Consider the special case of the memoryless binary symmetric channel (BSC) with cross-over probability p. We have

$$\Pr[\mathbf{y} \mid \mathbf{c}] \;=\; p^d (1-p)^{n-d} \;=\; (1-p)^n \left(\frac{p}{1-p}\right)^d$$

where $d \equiv d_H(\mathbf{y}, \mathbf{c}) \triangleq$ Hamming distance between $\mathbf{y}$ and $\mathbf{c}$, defined as the number of positions in which $\mathbf{y}$ and $\mathbf{c}$ differ.

- p and n are fixed, so $(1-p)^n$ is a constant.
- Also, $0 \leq p < {}^1\!/_2 \iff 0 \leq \frac{p}{1-p} < 1$ $\iff \left(\frac{p}{1-p}\right)^d$ decreases as d increases.
- Consequently, when $0 \leq p < {}^1\!/_2$, maximizing $\Pr[\mathbf{y} \mid \mathbf{c}]$ is equivalent to minimizing $d_H(\mathbf{y}, \mathbf{c})$.

Minimum Distance Decoding

Thus, for p in the range $0 \leq p < 1/2$, ML decoding is equivalent to

Minimum Distance Decoding (MDD):

> *Given a received word* $\mathbf{y}$*, decode to a codeword* $\mathbf{c} \in \mathcal{C}$
> *that minimizes* $d_H(\mathbf{y}, \mathbf{c})$*.* *(Ties are broken arbitrarily.)*

Minimum Distance Decoding

Thus, for p in the range $0 \le p < 1/2$, ML decoding is equivalent to Minimum Distance Decoding (MDD):

> *Given a received word* $\mathbf{y}$*, decode to a codeword* $\mathbf{c} \in \mathcal{C}$ *that minimizes* $d_H(\mathbf{y}, \mathbf{c})$*.* *(Ties are broken arbitrarily.)*

Remark: The equivalence between MLD and MDD also applies to q-ary block codes, provided we operate over a memoryless q-ary symmetric channel $(\mathbb{F}_q, \mathbb{F}_q, \Pr)$, defined by

- $\Pr[y \mid x] = \begin{cases} 1-p & \text{if } y = x \\ \frac{p}{q-1} & \text{if } y \neq x \end{cases} \quad \text{for } x, y \in \mathbb{F}_q$
- $\Pr[\mathbf{y} \mid \mathbf{x}] = \prod_{i=1}^{n} \Pr[y_i \mid x_i]$
  for $\mathbf{x} = (x_1, \ldots, x_n)$ and $\mathbf{y} = (y_1, \ldots, y_n)$.
- $0 \le p < 1 - \frac{1}{q}$

Hamming Distance

Definition: The Hamming distance between two words $\mathbf{x} = (x_1, \ldots, x_n)$ and $\mathbf{y} = (y_1, \ldots, y_n)$ (over any alphabet) is defined as

$$d_H(\mathbf{x}, \mathbf{y}) = \#\{i : x_i \neq y_i\}$$

Properties: For $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{F}^n$,

- $d_H(\mathbf{x}, \mathbf{y}) \geq 0$, with equality iff $\mathbf{x} = \mathbf{y}$
- $d_H(\mathbf{x}, \mathbf{y}) = d_H(\mathbf{y}, \mathbf{x})$
- $d_H(\mathbf{x}, \mathbf{z}) \leq d_H(\mathbf{x}, \mathbf{y}) + d_H(\mathbf{y}, \mathbf{z})$ (the triangle inequality)
  To prove the triangle inequality, interpret $d_H(\mathbf{x}, \mathbf{z})$ as the minimum number of symbol changes needed to convert $\mathbf{x}$ to $\mathbf{z}$.

The Minimum Distance of a Block Code

Definition: The minimum distance of a block code $\mathcal{C}$ is defined as

$$d(\mathcal{C}) \equiv d_{\min}(\mathcal{C}) \; = \; \min_{\substack{\mathbf{x},\mathbf{y} \in \mathcal{C} \\ \mathbf{x} \neq \mathbf{y}}} d_H(\mathbf{x}, \mathbf{y}).$$

Notation: An (n, M) block code with minimum distance d is referred to as an (n, M, d) block code.

The minimum distance of a block code is closely tied to its ability to handle errors.

Handling Channel Errors

- Let $\mathcal{C}$ be an (n, M, d) block code over the alphabet $\mathbb{F}$.
- We consider its ability to handle channel errors.
- An channel error is when one symbol in the transmitted codeword gets replaced by another.

  t errors $\Longrightarrow$ t symbols of the transmitted codeword are changed.

Error Detection

$\mathcal{C}$ an (n, M, d) block code.

Proposition:
There is a decoder for $\mathcal{C}$ that detects any occurrence of up to $d - 1$ channel errors.

Proof: Consider the decoder

$$\mathcal{D}(\mathbf{y}) = \begin{cases} \mathbf{y} & \text{if } \mathbf{y} \in \mathcal{C} \\ \text{ERROR} & \text{otherwise.} \end{cases}$$

Error detection fails iff one codeword gets changed to another by the channel. This happens only if at least d channel errors occur.

$\square$

Error Correction

Error correction entails finding the error locations and determining the correct symbol at each error location.
(For binary codes, it is enough to locate errors.)

Proposition:
There is a decoder for $\mathcal{C}$ that corrects any occurrence of up to $\lfloor \frac{d-1}{2} \rfloor$ channel errors.

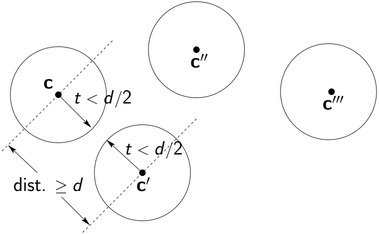
Error Correction

Error correction entails finding the error locations and determining the correct symbol at each error location.
(For binary codes, it is enough to locate errors.)

Proposition:
There is a decoder for $\mathcal{C}$ that corrects any occurrence of up to $\lfloor \frac{d-1}{2} \rfloor$ channel errors.

Proof: Take $\mathcal{D}$ to be the minimum distance decoder for $\mathcal{C}$.

- Consider the Hamming balls of radius $t = \lfloor \frac{d-1}{2} \rfloor$ centred at each $\mathbf{c} \in \mathcal{C}$:
  $B(\mathbf{c}, t) = \{\mathbf{y} \in \mathbb{F}^n : d_H(\mathbf{y}, \mathbf{c}) \leq t\}$.
- These Hamming balls are all disjoint.



Error Correction

Proof (cont'd):

- Suppose that $\mathbf{c} \in \mathcal{C}$ was transmitted and at most t errors occur.
- The received word $\mathbf{y}$ then lies in $B(\mathbf{c}, t)$.
- Since the Hamming balls are all disjoint, $\mathbf{y}$ lies outside $B(\mathbf{c}', t)$ for all $\mathbf{c}' \in \mathcal{C}$, $\mathbf{c}' \neq \mathbf{c}$.
  In other words, $d(\mathbf{y}, \mathbf{c}) \leq t$ but $d(\mathbf{y}, \mathbf{c}') > t$ for all $\mathbf{c}' \neq \mathbf{c}$.
- Hence, the minimum distance decoder $\mathcal{D}$ applied to $\mathbf{y}$ correctly recovers $\mathbf{c}$, showing that any pattern of up to $t = \lfloor \frac{d-1}{2} \rfloor$ channel errors can be corrected. $\square$

Error Detection and Correction

Again, $\mathcal{C}$ an (n, M, d) block code.

Proposition:
Let σ, τ be non-negative integers such that $2\tau + \sigma \leq d - 1$. There is a decoder for $\mathcal{C}$ that does the following:

- if the number of channel errors is $\leq \tau$, then all errors will be corrected
- if the number of channel errors is $\leq \tau + \sigma$, then the errors will be detected.

Error Detection and Correction

Again, $\mathcal{C}$ an (n, M, d) block code.

Proposition:
Let σ, τ be non-negative integers such that $2\tau + \sigma \le d - 1$. There is a decoder for $\mathcal{C}$ that does the following:

- if the number of channel errors is $\le \tau$, then all errors will be corrected
- if the number of channel errors is $\le \tau + \sigma$, then the errors will be detected.

Proof: Consider the decoder

$$\mathcal{D}(\mathbf{y}) \;=\; \begin{cases} \mathbf{c} & \text{if there is a } \mathbf{c} \in \mathcal{C} \text{ s.t. } d_H(\mathbf{y}, \mathbf{c}) \le \tau \\ \text{ERROR} & \text{otherwise.} \end{cases}$$

- if the number of channel errors is $\le \tau$, which is $\le \lfloor \frac{d-1}{2} \rfloor$, then they get corrected for the same reason as in prev. proposition.

Error Detection and Correction

Proof (cont'd):

- Now, consider the case of $\mathbf{c}$ transmitted, $\mathbf{y}$ received, with $d_H(\mathbf{y}, \mathbf{c}) \leq \tau + \sigma$.
- If these errors go undetected, it would mean that there is a $\mathbf{c}' \in \mathcal{C}$ s.t. $d_H(\mathbf{y}, \mathbf{c}') \leq \tau$.
- In this case,

$$\begin{aligned} d_H(\mathbf{c}, \mathbf{c}') &\leq d_H(\mathbf{c}, \mathbf{y}) + d_H(\mathbf{y}, \mathbf{c}') \\ &\leq (\tau + \sigma) + \tau \\ &= 2\tau + \sigma, \end{aligned}$$

  which is $\leq d - 1$ by the hypothesis of the proposition.
- This contradicts the fact that $d_{\min}(\mathcal{C}) = d$. $\square$

Erasures

- An erasure is an error whose location is known.
- Usually represented by a '?' symbol:

$$c_1 c_2 c_3 \ldots c_n \longrightarrow \boxed{\text{Channel}} \longrightarrow c_1 ? c_3 ?? c_6 \ldots c_n$$

Erasures

- An erasure is an error whose location is known.
- Usually represented by a '?' symbol:

$$c_1 c_2 c_3 \dots c_n \longrightarrow \boxed{\text{Channel}} \longrightarrow c_1 ? c_3 ?? c_6 \dots c_n$$

Proposition:
Let $\mathcal{C}$ be an (n, M, d) block code over $\mathbb{F}$. There is a decoder for $\mathcal{C}$ that corrects any occurrence of up to $d - 1$ erasures.

Erasures

- An erasure is an error whose location is known.
- Usually represented by a '?' symbol:

$$c_1 c_2 c_3 \ldots c_n \longrightarrow \boxed{\text{Channel}} \longrightarrow c_1 ? c_3 ?? c_6 \ldots c_n$$

Proposition:
Let $\mathcal{C}$ be an (n, M, d) block code over $\mathbb{F}$. There is a decoder for $\mathcal{C}$ that corrects any occurrence of up to $d-1$ erasures.

Proof: Let $\Phi = \mathbb{F} \cup \{?\}$.
Consider the decoder defined for each $\mathbf{y} \in \Phi^n$ as

$$\mathcal{D}(\mathbf{y}) = \begin{cases} \mathbf{c} & \text{if } \mathbf{c} \text{ is the } \underline{\text{unique}} \text{ codeword that agrees with } \mathbf{y} \\ & \text{on all unerased positions} \\ \text{ERROR} & \text{otherwise.} \end{cases}$$

Erasures

Proof (cont'd):

- Suppose that $\mathbf{c}$ was transmitted and at most $d-1$ of its coordinates were erased.
- The received word $\mathbf{y}$ contains at most $d-1$ '?' symbols, and agrees with $\mathbf{c}$ on all the unerased positions.
- If there were another $\mathbf{c}' \in \mathcal{C}$ that also agreed with $\mathbf{y}$ on all the unerased positions, then $\mathbf{c}$ and $\mathbf{c}'$ could differ only in those positions where $\mathbf{y}$ has '?' symbols.

$$\begin{array}{lcccccc} \mathbf{y} & \underline{\hspace{2cm}} & ? & ? & \cdots & ? & \underline{\hspace{2cm}} \\ \mathbf{c} & \underline{\hspace{2cm}} & * & * & \cdots & * & \underline{\hspace{2cm}} \\ \mathbf{c}' & \underline{\hspace{2cm}} & \square & \square & \cdots & \square & \underline{\hspace{2cm}} \end{array}$$

  Then, $d_H(\mathbf{c}, \mathbf{c}') \leq d-1$, which contradicts $d_{\min}(\mathcal{C}) = d$.

- Thus, $\mathbf{c}$ is the unique codeword that agrees with $\mathbf{y}$ in all unerased coordinates, and hence, $\mathcal{D}(\mathbf{y}) = \mathbf{c}$. $\square$

Errors and Erasures

Let $\mathcal{C}$ be an (n, M, d) block code .

Proposition:

Let τ, ρ be non-negative integers such that $2\tau + \rho \leq d - 1$. There is a decoder for $\mathcal{C}$ that can correct all error-cum-erasure patterns containing $\leq \tau$ errors and $\leq \rho$ erasures.

Errors and Erasures

Let $\mathcal{C}$ be an (n, M, d) block code .

Proposition:
Let τ, ρ be non-negative integers such that $2\tau + \rho \leq d - 1$. There is a decoder for $\mathcal{C}$ that can correct all error-cum-erasure patterns containing $\leq \tau$ errors and $\leq \rho$ erasures.

Proof is left as a homework exercise.

Some Remarks

- For a given block code $\mathcal{C}$, the choice of decoder determines the manner in which it handles errors.
- Error correction eats up twice as much of the error-handling budget as
  - error detection ($2\tau + \sigma \leq d - 1$)
  - erasures ($2\tau + \rho \leq d - 1$)
- The trouble with arbitrary (n, M, d) block codes is that encoding and decoding can be computationally expensive.

  Efficient encoding and decoding is possible if we impose additional structure on the block code — linearity!