An introduction to coding theory

Adrish Banerjee

Department of Electrical Engineering
Indian Institute of Technology Kanpur
Kanpur, Uttar Pradesh
India

Jan. 23, 2017

Lecture #1C: Introduction to error control coding-III

Outline of the lecture

- Types of Codes

Outline of the lecture

- Types of Codes
- Decoding strategies

Outline of the lecture

- Types of Codes
- Decoding strategies
- Error control strategies

Types of Channel Codes

Block codes:

- The information sequence is partitions into message blocks of k-information bits each, represented as:

$$\mathbf{u} = (u_0, u_1, \cdots, u_{k-1})$$

Types of Channel Codes

Block codes:

- The information sequence is partitions into message blocks of k-information bits each, represented as:

$$\mathbf{u} = (u_0, u_1, \cdots, u_{k-1})$$

- The encoder maps each block of k-information bits $\mathbf{u}$ to an n-bit codeword $\mathbf{v}$.

$$\mathbf{v} = (v_0, v_1, \cdots, v_{n-1})$$

Types of Channel Codes

Block codes:

- The information sequence is partitions into message blocks of k-information bits each, represented as:

$$\mathbf{u} = (u_0, u_1, \cdots, u_{k-1})$$

- The encoder maps each block of k-information bits $\mathbf{u}$ to an n-bit codeword $\mathbf{v}$.

$$\mathbf{v} = (v_0, v_1, \cdots, v_{n-1})$$

- The encoder for a block code is memoryless.

Types of Channel Codes

Block codes:

- The ratio k/n is known as *code rate* denoted by R.

Types of Channel Codes

Block codes:

- The ratio k/n is known as *code rate* denoted by R.
- $n - k$ is the number of redundant bits (also known as *parity bits*) added to each message to protect against errors.

Types of Channel Codes

Block codes:

- The ratio k/n is known as *code rate* denoted by R.
- $n - k$ is the number of redundant bits (also known as *parity bits*) added to each message to protect against errors.
- The set of 2^k code words of length n is called a binary (n, k) block code.

Types of Channel Codes

Block codes:

- The ratio k/n is known as *code rate* denoted by R.
- $n - k$ is the number of redundant bits (also known as *parity bits*) added to each message to protect against errors.
- The set of 2^k code words of length n is called a binary (n, k) block code.
- The codeword sequence, in general, can be non-binary, but we only consider binary codes since they are the most commonly used in practice.

Block Codes

- *Example*: Let $k = 3$ and $n = 6$. The following table gives a block code of length 6. The code rate is $R = \frac{1}{2}$.

Message	Codewords
(0 0 0)	(0 0 0 0 0 0)
(1 0 0)	(0 1 1 1 0 0)
(0 1 0)	(1 0 1 0 1 0)
(1 1 0)	(1 1 0 1 1 0)
(0 0 1)	(1 1 0 0 0 1)
(1 0 1)	(1 0 1 1 0 1)
(0 1 1)	(0 1 1 0 1 1)
(1 1 1)	(0 0 0 1 1 1)

Types of Channel Codes

Convolutional codes:

- A convolutional encoder processes the information sequence continuously.

Types of Channel Codes

Convolutional codes:

- A convolutional encoder processes the information sequence continuously.
- The n-bit encoder output at a particular time depends not only on the k-bit information sequence, but also on m previous input blocks, i.e., a convolutional encoder has a memory order of m.

Types of Channel Codes

Convolutional codes:

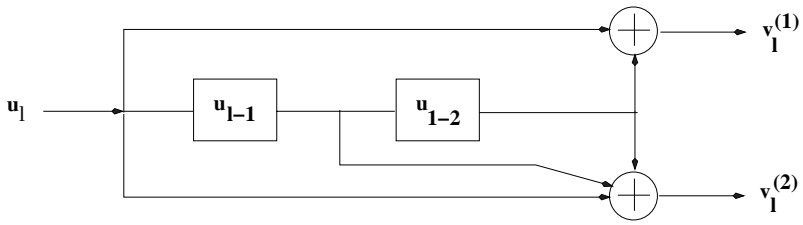
- A convolutional encoder processes the information sequence continuously.
- The n-bit encoder output at a particular time depends not only on the k-bit information sequence, but also on m previous input blocks, i.e., a convolutional encoder has a memory order of m.
- The set of sequences produced by a k-input, n-output encoder of memory order m is called an (n, k, m) convolutional code.

Types of Channel Codes

Convolutional codes:

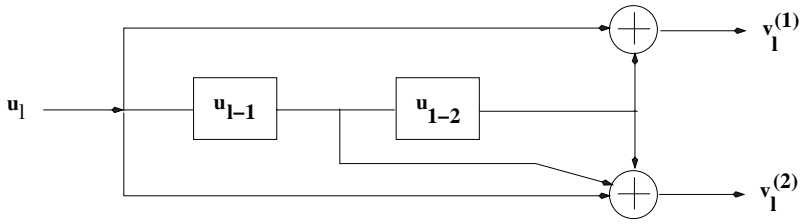
- A convolutional encoder processes the information sequence continuously.
- The n-bit encoder output at a particular time depends not only on the k-bit information sequence, but also on m previous input blocks, i.e., a convolutional encoder has a memory order of m.
- The set of sequences produced by a k-input, n-output encoder of memory order m is called an (n, k, m) convolutional code.
- The values of n and k are much smaller for convolutional codes compared to the block codes.

Convolutional Codes



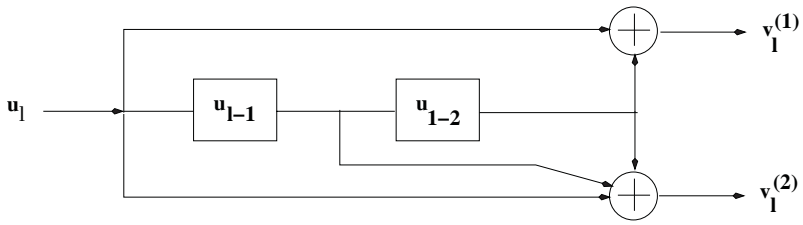
- Let $k = 1$, $n = 2$ and $m = 2$. The following circuit generates a $(2, 1, 2)$ convolutional code.

Convolutional Codes



- Let $k = 1$, $n = 2$ and $m = 2$. The following circuit generates a $(2, 1, 2)$ convolutional code.
- Input: u_l

Convolutional Codes



- Let $k = 1$, $n = 2$ and $m = 2$. The following circuit generates a $(2, 1, 2)$ convolutional code.
- Input: u_l
- Outputs:

$$\begin{aligned} v_l^{(1)} &= u_l + u_{l-2} \\ v_l^{(2)} &= u_l + u_{l-1} + u_{l-2} \end{aligned}$$

Decoding Strategies

- The decoder produces an estimate $\hat{\mathbf{u}}$ of the information sequence based on the received sequence $\mathbf{r}$.

Decoding Strategies

- The decoder produces an estimate $\hat{\mathbf{u}}$ of the information sequence based on the received sequence $\mathbf{r}$.
- Equivalently, the decoder can estimate $\hat{\mathbf{v}}$ of the code sequence and then use inverse encoder mapping to find the information sequence $\hat{\mathbf{u}}$ corresponding to $\hat{\mathbf{v}}$.

Decoding Strategies

- The decoder produces an estimate $\hat{\mathbf{u}}$ of the information sequence based on the received sequence $\mathbf{r}$.
- Equivalently, the decoder can estimate $\hat{\mathbf{v}}$ of the code sequence and then use inverse encoder mapping to find the information sequence $\hat{\mathbf{u}}$ corresponding to $\hat{\mathbf{v}}$.
- A *decoding rule* is an assignment of an estimate $\hat{\mathbf{v}}$ to each of the received sequence $\mathbf{r}$.

Decoding Strategies

- The average probability of error is given by

$$\begin{aligned} P(E) = P(\hat{\mathbf{v}} \neq \mathbf{v}) &= \sum_{\mathbf{r}} P(E/\mathbf{r})P(\mathbf{r}) \\ &= \sum_{\mathbf{r}} P(\hat{\mathbf{v}} \neq \mathbf{v}/\mathbf{r})P(\mathbf{r}) \end{aligned}$$

Decoding Strategies

- The average probability of error is given by

$$\begin{aligned} P(E) = P(\hat{\mathbf{v}} \neq \mathbf{v}) &= \sum_{\mathbf{r}} P(E/\mathbf{r})P(\mathbf{r}) \\ &= \sum_{\mathbf{r}} P(\hat{\mathbf{v}} \neq \mathbf{v}/\mathbf{r})P(\mathbf{r}) \end{aligned}$$

- Choose $\hat{\mathbf{v}}$ such that $P(\hat{\mathbf{v}} \neq \mathbf{v}/\mathbf{r})$ is minimized for each $\mathbf{r}$.

Decoding Strategies

- The average probability of error is given by

$$\begin{aligned} P(E) = P(\hat{\mathbf{v}} \neq \mathbf{v}) &= \sum_{\mathbf{r}} P(E/\mathbf{r})P(\mathbf{r}) \\ &= \sum_{\mathbf{r}} P(\hat{\mathbf{v}} \neq \mathbf{v}/\mathbf{r})P(\mathbf{r}) \end{aligned}$$

- Choose $\hat{\mathbf{v}}$ such that $P(\hat{\mathbf{v}} \neq \mathbf{v}/\mathbf{r})$ is minimized for each $\mathbf{r}$.
- Minimizing $P(\hat{\mathbf{v}} \neq \mathbf{v}/\mathbf{r})$ is equivalent to maximizing $P(\hat{\mathbf{v}} = \mathbf{v}/\mathbf{r})$.

Decoding Strategies

- For each $\mathbf{r}$, compute

$$P(\mathbf{v}/\mathbf{r}) = \frac{P(\mathbf{r}/\mathbf{v})P(\mathbf{v})}{P(\mathbf{r})} \quad \text{Bayes' rule}$$

  for every $\mathbf{v}$, and choose $\mathbf{v}$ that maximizes $P(\mathbf{v}/\mathbf{r})$.

Decoding Strategies

- For each $\mathbf{r}$, compute

$$P(\mathbf{v}/\mathbf{r}) = \frac{P(\mathbf{r}/\mathbf{v})P(\mathbf{v})}{P(\mathbf{r})} \quad \text{Bayes' rule}$$

  for every $\mathbf{v}$, and choose $\mathbf{v}$ that maximizes $P(\mathbf{v}/\mathbf{r})$.
- Equivalently, maximizing $P(\mathbf{v}/\mathbf{r})$ is same as maximizing $P(\mathbf{r}/\mathbf{v})P(\mathbf{v})$, since P($\mathbf{r}$) doesn't depend on $\mathbf{v}$.

Decoding Strategies

- For each $\mathbf{r}$, compute

$$P(\mathbf{v}/\mathbf{r}) = \frac{P(\mathbf{r}/\mathbf{v})P(\mathbf{v})}{P(\mathbf{r})} \quad \text{Bayes' rule}$$

  for every $\mathbf{v}$, and choose $\mathbf{v}$ that maximizes $P(\mathbf{v}/\mathbf{r})$.
- Equivalently, maximizing $P(\mathbf{v}/\mathbf{r})$ is same as maximizing $P(\mathbf{r}/\mathbf{v})P(\mathbf{v})$, since P($\mathbf{r}$) doesn't depend on $\mathbf{v}$.
- A Maximum a-posteriori probability (MAP) decoder chooses $\hat{v}$ such that $P(\mathbf{v}/\mathbf{r})$ is maximized.

Decoding Strategies

- If all code words are equally likely, maximizing $P(\mathbf{v}/\mathbf{r})$ is same as maximizing $P(\mathbf{r}/\mathbf{v})$.

Decoding Strategies

- If all code words are equally likely, maximizing $P(\mathbf{v}/\mathbf{r})$ is same as maximizing $P(\mathbf{r}/\mathbf{v})$.
- A maximum likelihood (ML) decoder chooses $\hat{v}$ such that $P(\mathbf{r}/\mathbf{v})$ is maximized.

Decoding Strategies

- If all code words are equally likely, maximizing $P(\mathbf{v}/\mathbf{r})$ is same as maximizing $P(\mathbf{r}/\mathbf{v})$.
- A maximum likelihood (ML) decoder chooses $\hat{v}$ such that $P(\mathbf{r}/\mathbf{v})$ is maximized.
- For a discrete memoryless channel (DMC) where each received symbol depends only on the corresponding transmitted symbol

$$P(\mathbf{r}/\mathbf{v}) = \prod_i P(r_i/v_i)$$

Decoding Strategies

- If all code words are equally likely, maximizing $P(\mathbf{v}/\mathbf{r})$ is same as maximizing $P(\mathbf{r}/\mathbf{v})$.
- A maximum likelihood (ML) decoder chooses $\hat{v}$ such that $P(\mathbf{r}/\mathbf{v})$ is maximized.
- For a discrete memoryless channel (DMC) where each received symbol depends only on the corresponding transmitted symbol

$$P(\mathbf{r}/\mathbf{v}) = \prod_i P(r_i/v_i)$$

- Since $\log x$ is a monotone increasing function of x, maximizing $P(\mathbf{r}/\mathbf{v})$ is equivalent to maximizing $\log P(\mathbf{r}/\mathbf{v})$.

Decoding Strategies

Example: ML rule for BSC

- For a codeword of length n transmitted on a BSC channel with crossover probability p, what should be the ML decoding rule?

Decoding Strategies

Example: ML rule for BSC

- For a codeword of length n transmitted on a BSC channel with crossover probability p, what should be the ML decoding rule?
- Hamming distance $d(\mathbf{r}, \mathbf{v})$ between $\mathbf{r}$ and $\mathbf{v}$ is the number of positions for which $r_i \neq v_i$.

$$\log P(\mathbf{r}/\mathbf{v}) = d(\mathbf{r}, \mathbf{v}) \log \left(\frac{p}{1-p} \right) + n \log(1-p)$$

Decoding Strategies

Example: ML rule for BSC

- For a codeword of length n transmitted on a BSC channel with crossover probability p, what should be the ML decoding rule?
- Hamming distance $d(\mathbf{r}, \mathbf{v})$ between $\mathbf{r}$ and $\mathbf{v}$ is the number of positions for which $r_i \neq v_i$.

$$\log P(\mathbf{r}/\mathbf{v}) = d(\mathbf{r}, \mathbf{v}) \log \left(\frac{p}{1-p} \right) + n \log(1-p)$$

- Note $\log \left(\frac{p}{1-p} \right) < 0$ for $p < 1/2$ and $n \log(1-p)$ is a constant.

Decoding Strategies

Example: ML rule for BSC

- For a codeword of length n transmitted on a BSC channel with crossover probability p, what should be the ML decoding rule?
- Hamming distance $d(\mathbf{r}, \mathbf{v})$ between $\mathbf{r}$ and $\mathbf{v}$ is the number of positions for which $r_i \neq v_i$.

$$\log P(\mathbf{r}/\mathbf{v}) = d(\mathbf{r}, \mathbf{v}) \log \left(\frac{p}{1-p} \right) + n \log(1-p)$$

- Note $\log \left(\frac{p}{1-p} \right) < 0$ for $p < 1/2$ and $n \log(1-p)$ is a constant.
- For each $\mathbf{r}$, choose $\hat{\mathbf{v}}$ as the codeword $\mathbf{v}$ which minimizes the Hamming distance $d(\mathbf{r}, \mathbf{v})$.

Error control strategies

Forward Error Correction (FEC):

- In *one-way* system, transmission takes place only in one direction, i.e. from transmitter to receiver.

Error control strategies

Forward Error Correction (FEC):

- In *one-way* system, transmission takes place only in one direction, i.e. from transmitter to receiver.
- The error correcting codes used in such a system are referred as *forward error correction (FEC)* codes.

Error control strategies

Automatic Repeat reQuest (ARQ):

- In *two-way* system, there exist a feedback path from the receiver to the transmitter.

Error control strategies

Automatic Repeat reQuest (ARQ):

- In *two-way* system, there exist a feedback path from the receiver to the transmitter.
- Error correction can be achieved for two-way system using error detection and retransmission, also known as *automatic repeat request (ARQ)*.

Error control strategies

Hybrid Automatic Repeat reQuest (HARQ):

- For channels with feedback, one can use ARQ protocols in combination with FEC to improve system performance. These types of schemes are known as hybrid-ARQ (HARQ) schemes.

Error control strategies

Hybrid Automatic Repeat reQuest (HARQ):

- For channels with feedback, one can use ARQ protocols in combination with FEC to improve system performance. These types of schemes are known as hybrid-ARQ (HARQ) schemes.
- In HARQ protocols, the transmitted data is encoded for both error correction and error detection. If the receiver detects an error after decoding, it sends a negative acknowledgment (NACK) to the transmitter, which then retransmits the data.