Outline

- Basics
- Block Codes
- Convolutional Codes
- Modern Channel Code
Why Coding

1. Use redundancy to enhance robustness: if the information bit is impaired (by noise, fading, interference etc), it can still be recovered from redundant bits – error correcting codes.

2. We can also use channel codes to detect transmission errors without correcting them. Then the data can be retransmitted.

3. Used in wireless communications, storage systems (hard disk) et al, coding is THE ART of communications.

4. A simple example: repeated codes.
Some Terminology

- **Codeword**: some number of channel uses to represent some information bits.
- **Codebook**: the ensemble of codewords.
- **Codeword length**: number of channel uses.
- **Encoding**: mapping from information bits to codewords.
- **Decoding**: mapping from received signal to information bits.
Requirements of A Good Coding Scheme

- Low bit/block error rate.
- Low-complexity encoder and decoder. (random coding has optimal performance, but the decoding procedure has to look up a huge codebook)
- Reasonable codeword length (otherwise the delay is too long)
- No performance floor.
Block Codes and Convolutional Codes

In block codes, a block of $k$ information digits is encoded by a codeword of $n$ digits $n \geq k$. The codings of different blocks are independent.

In convolutional codes, the coded sequence of $n$ digits depends not only on the $k$ data digits but also the previous $N - 1$ digits.
Hamming Distance

- Codeword weight: number of nonzero bits.
- We can consider each codeword as a point in the space.
- Hamming distance $d_{ij}$: the number of different coded bits in two codewords $c_i$ and $c_j$ (the weight of codeword $c_1 - c_2$).
- What does a larger Hamming distance mean?
Minimum Distance

Minimum distance (also minimum weight of nonzero codewords):

\[ d_{\text{min}} = \min_{i,j} d_{ij}, \]

which characterizes how close the codewords are close to each other (the closer, the worse performance).

Singleton bound:

\[ d_{\text{min}} \leq n - k + 1. \]

If equality holds, it is code maximum distance separation (MDS) code. A codeword in a MDS code is uniquely determined by any \( k \) elements (why?).
Capability of Error Detection and Correction

- Error detection: at most $d_{\text{min}} - 1$ error bits can be detected (but may not be corrected).
- Error correction: at most $t = \left\lfloor \frac{d_{\text{min}} - 1}{2} \right\rfloor$ error bits can be corrected.
- Hamming bound:

\[
2^{n-k} \geq \sum_{j=0}^{t} \binom{n}{j}
\]
### Table 16.1
Some examples of error correcting codes

<table>
<thead>
<tr>
<th></th>
<th>$n$</th>
<th>$k$</th>
<th>Code</th>
<th>Code Efficiency (or Code Rate)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single-error correcting, $t = 1$</td>
<td>3</td>
<td>1</td>
<td>(3, 1)</td>
<td>0.33</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>1</td>
<td>(4, 1)</td>
<td>0.25</td>
</tr>
<tr>
<td>Minimum code separation 3</td>
<td>5</td>
<td>2</td>
<td>(5, 2)</td>
<td>0.4</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>3</td>
<td>(6, 3)</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>4</td>
<td>(7, 4)</td>
<td>0.57</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>11</td>
<td>(15, 11)</td>
<td>0.73</td>
</tr>
<tr>
<td></td>
<td>31</td>
<td>26</td>
<td>(31, 26)</td>
<td>0.838</td>
</tr>
<tr>
<td>Double-error correcting, $t = 2$</td>
<td>10</td>
<td>4</td>
<td>(10, 4)</td>
<td>0.4</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>8</td>
<td>(15, 8)</td>
<td>0.533</td>
</tr>
<tr>
<td>Minimum code separation 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Triple-error correcting, $t = 3$</td>
<td>10</td>
<td>2</td>
<td>(10, 2)</td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>5</td>
<td>(15, 5)</td>
<td>0.33</td>
</tr>
<tr>
<td>Minimum code separation 7</td>
<td>23</td>
<td>12</td>
<td>(23, 12)</td>
<td>0.52</td>
</tr>
</tbody>
</table>
Outline

- Basics
- Block Codes
- Convolutional Codes
- Modern Channel Code
Generating Matrix

1. Information bits: $k$-dimensional row vector space; codewords: $n$-dimensional row vector space.

2. Generator matrix: mapping from information bit space to codeword space, namely a $k \times n$ matrix.

3. Systematic code: if the generator matrix contains a $k \times k$ identity submatrix (the information bits appear in $k$ locations of codeword).

4. Encoding of information bit vector $u$:

$$c = uG$$
Parity Check Matrix

1. Parity check matrix $H (n - k \times n)$ of generator matrix $G$ satisfies:

   \[ GH^T = 0. \]

2. If receiving $s = c + e$ (called senseword), where $e$ is error vector, multiplying $H$ yields

   \[ sH^T = uGH^T + eH^T = eH^T, \]

   which is called syndrome of $s$.

3. If syndrome is nonzero, we can claim decoding error. If establishing a mapping between syndrome and error vector (from an $n - k$-vector to an $n$-vector), we can correct some error bits.
Encoder

Generating matrix:

\[
G = \begin{pmatrix}
1 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 \\
\end{pmatrix}
\] (1)
Cyclic Code

1. Cyclic code: any cyclic shift of any codeword is another codeword.

2. Polynomial representation \((z\) is a shift operator) \(c(z) = g(z)u(z)\):

   - message: \(u(z) = \sum_{i=0}^{k-1} u_i z^i\)
   - generator: \(g(z) = \sum_{i=0}^{n-k} g_i z^i\)
   - codeword: \(c(z) = \sum_{i=0}^{n-1} c_i z^i\)
Hard Decision Decoding (HDD)

The maximum likelihood detection of codewords is based on a distance decoding metric.
The error probability is upper bounded by

\[ P_e \leq \sum_{j=t+1}^{n} \binom{n}{j} p^j (1 - p)^{n-j}, \]

where \( p \) is the demodulation error probability of a single bit.

The low bound is given by

\[ P_e \geq \sum_{j=t+1}^{d_{\text{min}}} \binom{d_{\text{min}}}{j} p^j (1 - p)^{d_{\text{min}}-j}, \]
Soft Decoding

1. Choose the codeword with maximum \textit{a posteriori} probability:

\[
C^* = \arg \min_C p(R|C),
\]

where \( R \) is observation.

2. For AWGN,

\[
p(R|C) \sim \prod_{t=1}^n \exp\left(-\frac{R_t - \sqrt{E_c(2C_t - 1)^2}}{2\sigma_n^2}\right).
\]

3. It is difficult to do exhaustive search for all codewords.
Outline

- Basics
- Block Codes
- Convolutional Codes
- Modern Channel Code
History of Convolutional Code

1. Convolutional code was first proposed by Elias (1954), developed by Wozencraft (1957) and rediscovered by Hagelbarger (1959).
2. Viterbi proposed his celebrated algorithm for hard decoding of convolutional code (essentially the same as Bellman’s dynamic programming) in 1967.
3. Soft decision algorithm (BCJR algorithm) was proposed by Bahl, Cocke, Jelinek and Raviv in 1974 (essentially the same as forward-backward algorithm in HMM).
4. Convolutional code is widely used in ... for its high efficient encoding and decoding algorithms.
Diagram Representation

1. Constituent: memory, output operators.
2. Constraint length $\nu$: length of memory.
3. Rate $p/q$: $q$ output bits when $p$ information bits are input.
There are $2^\nu$ states. Possible state transitions are labeled.

For each combination of input information bit and state, the outputs are labeled in the trellis.
Polynomial Representation

1. If one input information bit one time, use $p$ (number of output operators) polynomials $\{g_i\}_{i=1,...,p}$ to represent the code, where $x$ is delay operator (like $z$ transform). E.g. $g_0(x) = x^2 + x + 1$ and $g_1(x) = x^2 + 1$.

2. If multiple inputs, we can use polynomial matrix. For example

$$G(x) = \begin{pmatrix} x & 1 & 0 \\ 1 & x^2 & 1 \end{pmatrix}$$
Hard Decoding

1. Demodulation output is binary, 0 and 1 (hard).
2. Viterbi algorithm searches a path in the trellis with minimum discrepancy compared with the demodulation output.
3. For the $t$-th demodulation output, each node in the trellis chooses the node in the $t - 1$-th output having least output discrepancy, inherits its path and carries the accumulated discrepancy.
Soft Decoding: BCJR Algorithm

1. Forward pass:

\[ \alpha_t(m) = \sum_{m'} \alpha_{t-1}(m') \gamma_t(m', m), \]

with initial condition \( \alpha_0(0) = 1 \) and \( \alpha_0(m) = 0 \) for \( m \neq 0 \)

2. Backward pass:

\[ \beta_t(m) = \sum_{m'} \beta_{t+1}(m') \gamma_{t+1}(m, m'), \]

with initial condition \( \beta_n(0) = 1 \) and \( \beta_n(m) = 0 \) for \( m \neq 0 \)

3. Compute joint probability, from which we can compare \textit{a posteriori} probability:

\[ \lambda_t(m) = \alpha_t(m) \beta_t(m). \]
Comparison Between Hard and Soft Decoding

1. Soft decoding utilizes more information (demodulation with much ambiguity is substantially ignored). Therefore, soft decoder achieves better performance.

2. For convolutional code with constraint length 3 and transmission rate 1/3, soft decoding has 2-3dB power gain (for achieving the same decoding error probability, soft decoding needs 2-3dB less power than hard decoding).

3. Soft decoding has more computational cost than hard decoding.
State Diagram
Transfer Function

1. State equations:

\[ X_c = D^3 X_a + DX_b, \quad X_b = DX_c + DX_d \]
\[ X_d = D^2 X_c + D^2 X_d, \quad X_e = D^2 X_b \]

2. Transfer function:

\[ T(D) = \frac{D^6}{1 - 2D^2} = D^6 + 2D^8 + 4D^{10} + \ldots \]

minimum distance is 6; there are two paths with Hamming distance 8.
Concatenated Codes
Problem 1: Consider a (3,1) linear block code where each codeword consists of 3 data bits and 1 parity bit. (a) find all codewords in this code. (b) find the minimum distance of the code $d_{\text{min}}$.

Problem 2: Consider a (7,4) code with generator matrix

$$G = \begin{pmatrix}
0 & 1 & 0 & 1 & 1 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 1 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}$$

(a) Find all codewords of the code; (b) What is $d_{\text{min}}$? (c) Find the parity check matrix of the code; (d) Find the syndrome of the received vector $R = [1101011]$. 
Homework 5

Problem 3: Sketch the trellis diagram and the state diagram for the above convolutional encoder. What is the memory length? What is the coding rate?

Problem 4: Suppose that the input message bits are 1100010. What is the output coded bits? Suppose that transmission errors occur in the 4-th coded bit and the last coded bit. Use the Viterbi’s algorithm for the decoding procedure.
Outline

- Basics
- Block Codes
- Convolutional Codes
- Modern Channel Code
Storm in Coding Theory: Turbo Code

1. Before 1993, practical code only achieved performance several dBs beyond Shannon capacity.
3. "Their simulation curves claimed unbelievable performance, way beyond what was thought possible". "The thing that blew everyone away about turbo codes is not just that they get so close to Shannon capacity but that they’re so easy.” — McEliece
4. "They said we must have made an error in our simulations” — Berrou
Rediscovery of LDPC

1. Gallager devised LDPC in his PhD thesis in 1960. It was still impractical at that time.
2. People (Mackay, Richardson, Urbanke) rediscovered LDPC in late 1990s.
3. LDPC beat turbo code and is now very close to Shannon limit (0.0045dB away, Chung 2001).
4. “A piece of 21st century coding that happened to fall in the 20th century” — Forney
5. “We’re close enough to the Shannon limit that from now on, the improvements will only be incremental” — Tom Richardson
A turbo encoder has several component encoders (e.g. convolutional code) and an interleaver (the magic part!).
An interleaver is used to permute the information bit sequence. It brings randomness to the encoder (similar to Shannon’s random coding)
Decoding is done in an iterative (turbo) way.

2. The soft output (e.g. *a posteriori* probability) of one decoder is used as the *a priori* probability of another decoder.

3. Such a turbo principle can be applied in many other fields: turbo equalization, turbo multiuser detection and so on.
A low-density parity-check code is a linear code with a sparse check matrix ($H$). It could be represented by Tanner graphs.

Parity check matrix:

$$
H = \begin{pmatrix}
1 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 1 \\
1 & 0 & 0 & 1 & 1 & 0
\end{pmatrix}
$$
LDPC Decoder

1. Message Passing is also called Belief Propagation.

2. Step 1: at each check node, the messages from variables nodes are passed to neighboring variable nodes.

3. Step 2: at each variable node, the messages from check nodes are passed to neighboring check nodes (incorporating the channel observations).

4. Repeat above two steps.