Convolutional Coding & Viterbi Algorithm

Er Liu

liuer@cc.hut.fi
Communications Laboratory
16.11.2004
Outline

Convolutional Coding
- Convolutional code
- Generator sequence
- Trellis and state diagram

Viterbi Algorithm
- Maximum-Likelihood decoding
- Viterbi algorithm
Convoluotional Encoding

Convoluotional codes are applied in applications that require good performance with low implementation cost. They operate on data stream, not static block.

Convoluotional codes have memory that uses previous bits to encode or decode following bits.

It is denoted by \((n,k,L)\), where \(L\) is code memory depth.

Code rate \(r\) is determined by input rate and output rate:

\[
r = \frac{r_{input}}{r_{output}} < 1
\]
Convolutional Encoder

- Convolutional encoder is a finite state machine (FSM), processing information bits in a serial manner.
- Thus the generated code is a function of input and the states of the FSM.
- In this \((n,k,L) = (2,1,2)\) encoder each message bits influences a span of \(n(L+1) = 6\) successive output bits.

\[
\begin{align*}
    x'_j &= m_{j-2} \oplus m_{j-1} \oplus m_j \\
    x''_j &= m_{j-2} \oplus m_j
\end{align*}
\]

\((n,k,L) = (2,1,2)\) encoder
Another Encoder example

(3,2,1) Convolutional encoder, $k = 2$

\[ x'_j = m_{j-3} \oplus m_{j-2} \oplus m_j \]
\[ x''_j = m_{j-3} \oplus m_{j-2} \oplus m_j \]
\[ x'''_j = m_{j-2} \oplus m_j \]

Here each message bit influences a span of $C = n(L+1) = 3(1+1) = 6$ successive output bits.
Convolutional code can be described by generator sequences $g^{(1)}, g^{(2)}, \ldots, g^{(n)}$ that are the impulse responses of each coder output branch.

Generator sequences specify convolutional code completely by the associated generator matrix.

Encoded convolutional code is produced by matrix multiplication of input and the generator matrix.

\[
\begin{align*}
g^{(1)} &= [1 \ 0 \ 1 \ 1] \\
g^{(2)} &= [1 \ 1 \ 1 \ 1]
\end{align*}
\]
Example of Using Generator Matrix

If \( u = (1 \ 0 \ 1 \ 1 \ 1) \), then

\[
v = uG = \begin{bmatrix}
1 & 1 & 0 & 1 & 1 & 1 & 1 \\
1 & 1 & 0 & 1 & 1 & 1 & 1 \\
1 & 1 & 0 & 1 & 1 & 1 & 1 \\
1 & 1 & 0 & 1 & 1 & 1 & 1 \\
\end{bmatrix}
\]

\[
= (1 \ 0 \ 1 \ 1 \ 1)
\]

It can also use polynomial multiplication
Representation – Code Tree

This tells how one input bit is transformed into two output bits (initially register is all zero)
Trellis and State Diagram

(a) Current state
- 00 = α
- 01 = β
- 10 = γ
- 11 = δ

Output
- 00
- 11
- 01
- 10

Next state
- 00
- 01
- 11
- 10

(b) Input state ‘1’ indicated by dashed line
- b
- c
- d

(c) Input
- 1 1 0 1 1 1 0 0 1 0 0 0 0

State
- a b d c b d d c a b c a a

Output
- 11 01 01 00 01 10 01 11 11 10 11 00

Shift register states

Convolutional Coding & Viterbi Algorithm

Er Liu (liuer@cc.hut.fi)
Minimum Hamming Distance

The path representing the state sequence $S_0S_1S_3S_5S_6S_5S_2S_4S_0$ has path gain $X^2X^1X^1X^1X^2X^2X^2=X^{12}$ and the corresponding code word has the weight 12.

$$T(X) = \sum_i A_i X^i$$

$$= X^0 + 3X^7 + 5X^8 + 11X^9 + 25X^{10} + \ldots$$

Where does these terms come from?
Maximum-Likelihood Decoding

Maximum likelihood decoding means finding the code branch in the code trellis that was most likely to transmitted.

Therefore maximum likelihood decoding is based on calculating the hamming distances for each branch forming encode word.

Probability to decode sequence is then

\[ p(y, x) = \prod_{j=0}^{\infty} p(y_j | x_j) \]

The most likely path through the trellis will maximize this metric.
Example of Maximal Likelihood Detection

Assume a three bit message is to transmitted. To clear the encoder two zero-bits are appended after message. Thus 5 bits are inserted into encoder and 10 bits produced. Assume channel error probability is $p=0.1$. After the channel 10, 01, 10, 11, 00 is produced. What comes after decoder, e.g. what was most likely the transmitted sequence?
Example of Maximal Likelihood Detection

The Hamming distance between this path and the received sequence is 5. All paths (specified by the encoder input bits) and their path metrics and Hamming distances are listed below.

Received sequence: 10, 01, 10, 11, 00

<table>
<thead>
<tr>
<th>Path</th>
<th>Code Sequence</th>
<th>Path Metric</th>
<th>Hamming Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>0, 0, 0, 0, 0</td>
<td>00, 00, 00, 00, 00</td>
<td>-12.05</td>
<td>5</td>
</tr>
<tr>
<td>0, 0, 1, 0, 0</td>
<td>00, 00, 11, 10, 11</td>
<td>-11.34</td>
<td>6</td>
</tr>
<tr>
<td>0, 1, 0, 0, 0</td>
<td>00, 11, 10, 11, 11</td>
<td>-5.48</td>
<td>2</td>
</tr>
<tr>
<td>0, 1, 1, 0, 0</td>
<td>00, 11, 01, 01, 11</td>
<td>-16.43</td>
<td>7</td>
</tr>
<tr>
<td>1, 0, 0, 0, 0</td>
<td>11, 10, 11, 00, 00</td>
<td>-14.24</td>
<td>6</td>
</tr>
<tr>
<td>1, 0, 1, 0, 0</td>
<td>11, 10, 00, 10, 11</td>
<td>-16.43</td>
<td>7</td>
</tr>
<tr>
<td>1, 1, 0, 0, 0</td>
<td>11, 01, 01, 11, 00</td>
<td>-7.67</td>
<td>3</td>
</tr>
<tr>
<td>1, 1, 1, 0, 0</td>
<td>11, 01, 10, 01, 11</td>
<td>-9.86</td>
<td>4</td>
</tr>
</tbody>
</table>

Note also the Hamming distances!
Viterbi Algorithm

- ML algorithm is too complex to search all available paths
  - End to end calculation
- Viterbi algorithm performs ML decoding by reducing its complexity
  - Eliminate least likely trellis path at each transmission stage
  - Reduce decoding complexity with early rejection of unlike pathes
- Viterbi algorithm gets its efficiency via concentrating on survival paths of the trellis
Viterbi decoding

- Viterbi decoding compares the Hamming distance between the branch code and the received code.
- Path producing larger Hamming distance is eliminated (after constraint length).
Example of Viterbi decoding

Input data: \( m = 1 \ 1 \ 0 \ 1 \ 1 \)
Codeword: \( X = 11 \ 01 \ 01 \ 00 \ 01 \)
Received code: \( Z = 11 \ 01 \ 01 \ 10 \ 01 \)
Homework

Please use Viterbi algorithm to decode the received sequence:

\[ Z = [11 \ 10 \ 10 \ 10 \ 01] \]

Please draw the trellis and state diagram.
Any questions?

Thanks!