What is this “Viterbi Decoding”

Andrew A. Lamb
MIT
Laboratory for Computer Science
Computer Architecture Group
(8/2/2002)
State Machine/Trellis Encoding?

- You can also represent the convolution encoder as a finite state machine.

- The possible state evolution of the convolution encoder traces out a “trellis” like structure in state-time space.
Convolutional Encoder

Source: ASTC Standard A53, revision B
Interesting Part
State Transition Diagram

- **00**: 
  - **In**: 0
  - **Out**: 0
- **01**: 
  - **In**: 1
  - **Out**: 0
  - **In**: 1
  - **Out**: 1
- **11**: 
  - **In**: 0
  - **Out**: 1
  - **In**: 1
  - **Out**: 1
- **10**: 
  - **In**: 1
  - **Out**: 0
  - **In**: 0
  - **Out**: 1

**In** = 0
**Out** = 0
**In** = 0
**Out** = 1
Note: it is assumed that you start in state 00
Trellis Paths

- Each set of inputs traces a unique path through the trellis.

Input:
01101
01110
Symbol Mapping

<table>
<thead>
<tr>
<th>Z2</th>
<th>Z1</th>
<th>Z0</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-7</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-5</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-3</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>+1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>+3</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>+5</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>+7</td>
</tr>
</tbody>
</table>
Trellis Decoder

- Use Viterbi Algorithm to undo Convolution Coding (not signal symbol unmapping)

- Two flavors of determining unmapping
  - Hard: make a decision as to which symbol the received signal most closely matches.
  - Soft: Assign weights to all symbols based on their respective likelihood given received signal

- Viterbi implementation we present assumes a hard decision model.
Viterbi Decoding (Overview)

- The Viterbi algorithm:
  - Given a sequence of received symbols, (that were produced by a convolution encoder, sent over a channel)
  - Determine what the input to the convolution encoder was
  - It does this by determining the most likely path through the trellis
Viterbi Decoding (Main Idea)

- Dynamic Programming
  - Keep a table $c[s,t]$ that records the number of errors* that would have been accumulated if the encoder was in state $s$ at time $t$.
  - Also keep a table $p[s,t]$ which records the state that the encoder would have been in at time $t-1$ if it were in state $s$ at time $t$.

* Typically calculated using either Hamming or Euclidian distance
Filling out the tables

- At time $t$, we receive the symbol $R_t$.
- For each state $s$,
  - Let $q_0$ and $q_1$ be the two possible previous states of $s$ at time $t-1$.
  - Let $e_0$ be the error between $R_t$ and $q_0 \to s$.
  - Let $e_1$ be the error between $R_t$ and $q_1 \to s$.
  - $c[s,t] \leftarrow \min(c[q_0,t-1]+e_0, c[q_1,t-1]+e_1)$
  - Update $p[s,t]$ appropriately with $q_0$ or $q_1$. 
Traceback

- When the algorithm has examined $T$ input symbols, it looks for the minimum entry among all states in $c[s,T]$.

- Then the algorithm traces back through the trellis using the entries of $p[s,t]$. 
Example

- Input to Encoder:
  - 0110101110

- Output of Encoder:
  - 00 10 11 01 11 00 11 11 10 00
Decoder, time = 1
Decoder, time = 2
Decoder, time = 3
Decoder, time = 4

State

Time

received symbol N/A 00 10 11 01

err=0 err=1 err=2 err=3

input=0 input=1

Decoder diagram and state transition graph.
Decoder, time = 5
Decoder, time = 6

State

received symbol  N/A  00  10  11  01  11  00

Time

20
Decoder, time = 7
Decoder, time = 8
Decoder, time = 9
Decoder, time = 10
Decoder, Traceback
Original Path
Questions

- Is there a way to express the Viterbi algorithm in a fine grained stream graph?
- Variable rates for this decoder would mean pop(2) for 5K times, and then push(5K). This is deterministic, if not constant rate.
- Perhaps 2 stage filters?