Outline

- Viterbi decoder
  - How it works
  - What is it used for
  - Digital implementations
  - Analog implementations
Viterbi Algorithm

- Example of dynamic programming [Bellman'57]
- Invented by A. Viterbi in 1967
- Explained by Forney in 1972, 1973

Used for:
- Decoding convolutional codes
- Decoding trellis codes
- Maximum likelihood detection
- Speech recognition, etc.

Types:
- Hard-input, hard-output
- Soft-input, hard-output
- Soft-input, soft-output

Jared’s Trip to Berkeley
Jared’s Trip to Berkeley

Trellis

- States + edges
- No loops
- Weights in minutes

Diagram:

- Milpitas 15
- Union City 10
- Hayward 15
- Oakland 15
- Mountain View 5
- Palo Alto 20
- San Mateo 20
- San Francisco 40
- Berkeley
Shortest Time to Get to Berkeley?

- What is the best path to take to:
  - Union City?
  - Hayward?

- Choose the minimum cost at each point (state)
Shortest Time to Get to Berkeley?

What is the best path to take to:

- Hayward?

![Map showing distances and routes between locations including Berkeley, Oakland, San Francisco, San Mateo, Palo Alto, Union City, and Milpitas.]
**Digital Baseband Transceiver**

![Diagram of a digital baseband transceiver]

**Convolutional Codes**

- **Adding redundancy**
  
  ![Convolutional Code Diagram]

- **Generators:**
  
  \( G_1 = 101 \)
  
  \( G_2 = 111 \)
State Transition Diagrams

Error Sequences

Minimum distance error event is 4 bits
Decisions at Each Step

- In Gaussian channel:
  \[ bm = (y_k - s_k)^2 \]
- In BSC:
  \[ bm = d_H(y_k, s_k) = |y_k - s_k| \]

\( d_H \) is Hamming distance

- If received \( y_k = 10 \)
  \[ d_H(10, 00) = 1 \]
  \[ d_H(10, 01) = 2 \]

Trellis Diagram

- Time-indexed state diagram
The Viterbi Algorithm

Illustrated by 2-state trellis

\[ \begin{align*}
sm_{1n} &= \min \left( sm_{1n-1} + bm_1, sm_{2n-1} + bm_3 \right) \\
sm_{2n} &= \min \left( sm_{1n-1} + bm_2, sm_{2n-1} + bm_4 \right)
\end{align*} \]

Viterbi Decoder

Implements Viterbi algorithm

Three main components

- Branch metric calculation
- Path metric accumulation (add-compare-select recursion)
- Survivor path decode
Calculating Branch Metrics

- Assume $G_1$ and $G_2$ both output a 0
- Soft decoder inputs are 0.1 and 0.25 instead
- Branch metrics:
  - $bm_{00} = |0 - 0.1| + |0 - 0.25| = 0.35$
  - $bm_{01} = |0 - 0.1| + |1 - 0.25| = 0.85$
  - $bm_{10} = |1 - 0.1| + |0 - 0.25| = 1.15$
  - $bm_{11} = |1 - 0.1| + |1 - 0.25| = 1.65$
Calculating Branch Metrics

- Euclidean distances (AWGN channel)
  \[ b_{m_i} = (y_k - s_k)^2 \]
- \( s_k \) are usually integers
  \[ b_{m_i} = y_k^2 - 2C_i y_k + C_i^2 \]
- Since \( y_k \) terms are common to all branch metrics they drop out in ACS comparison, and can be eliminated.
- \( C_i^2 \) are precomputed and \( 2C_i y_k \) are shifts and adds.

Add-Compare-Select Recursion

Eight state trellis

Conventional add-compare-select unit
Add-Compare-Select Recursion
Add-Compare-Select Recursion

One step lookahead applied to an eight-state trellis

Radix-4 add-compare-select unit
Add-Compare-Select Recursion

4-way ACS:
2 additions +
6 comparisons

Add-Compare-Select Recursion

Concurrent add and compare

Sridharan, Carley, JSSC’00
Add-Compare-Select Recursion

Retimed ACS

Fettweiss, Globecom'95
Add-Compare-Select Recursion

Area comparison of various ACS structures

Power comparison of various ACS structures
Extreme ACSs

» LSB-first computation

Fettweiss, Meyr, TCom’90
Yeung, Rabaey, ISSCC’95
Drawings by Haratsch

» MSB-first computation

RNS-Based ACS

\[ S = \sum_{i=0}^{w-1} s_i 2^i \]

\( s_i \in \{0, 1, 2\} \), and are two-bit encoded as \( s_i \in \{00, 01, 11\} \)
**Bit-Level Pipelined ACS**

Yeung, Rabaey, ISSCC’95

**Analog ACS**

Fukahori, ISSCC’98
Interleaved Analog ACS

Metric Normalization

State metrics $sm_i$ and $sm_j$ represented on a number circle
Survivor Path Decode

4-state trellis example

Register Exchange