rle AT MIT

www.rle.mit.edu

Recitation 9

10-6-2011

Last Time:

~ Encoding CC's using various methods.

– state table, state diagram, generator sequence & D-transform, etc.

Announcements:

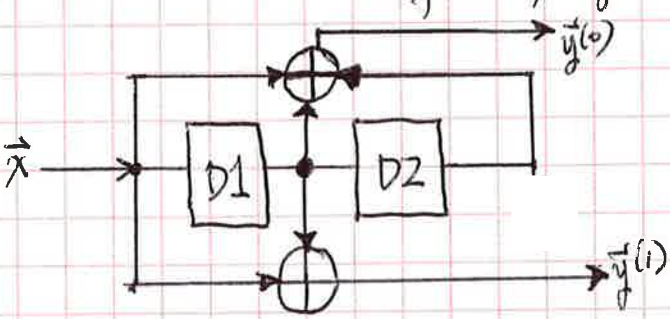
1. – Steve Jobs' passing
2. – Quiz 1 review
3. – Office Hrs by Appt. (please send topics ahead of time)

Today:

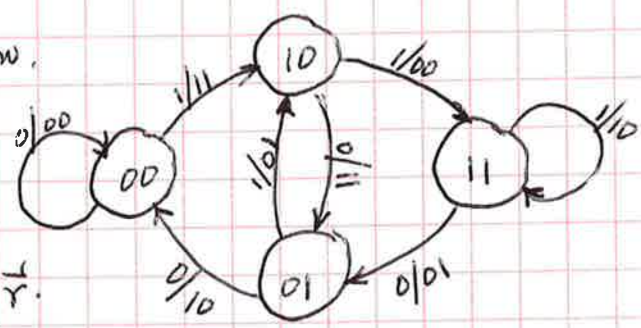
Viterbi Decoding of CC's.

Example 1:

(*) Consider the convolutional encoder defined by $\vec{g}^{(0)} = (1\,1\,1)$ and $\vec{g}^{(1)} = (1\,1\,0)$ shown below. (Reproduced from lecture notes)



The corresponding state diagram is as below.



We can construct a trellis diagram, very useful in decoding a received vector $\vec{r}$.

Trellis: ~ extension of state diagram with explicit time passage!

Observations:

1. As with state diagram, trellis has $2^m = 2^{K-1}$ (constraint length) states (nodes), for m-memory or K constraint length encoder.
2. For (n, k) binary CC, each node has 2^k (input) branches leaving and 2^k branches entering (so for our rate-1/n codes, $2^1 = 2$-branches leave and enter each node.
3. We'll assume – that encoder starts at zero-state (state S_0)
   – that encoder is time-invariant, so $\vec{y}$ is constant
   – that noise is uncorrelated from stage-to-stage.

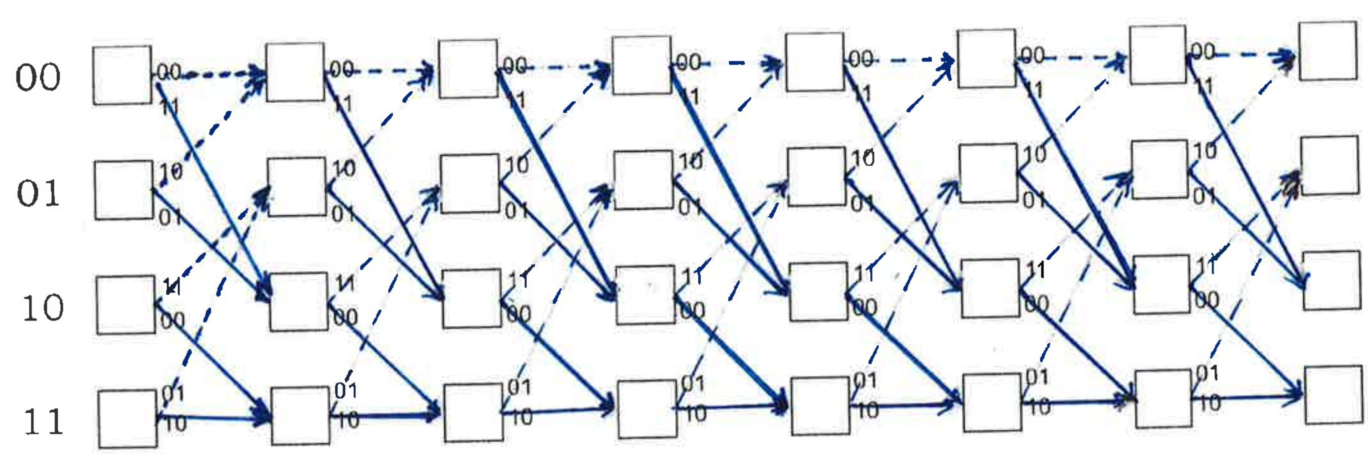
rle AT MIT

www.rle.mit.edu

(4) After all bits of input vector $\vec{x}$ entered, need m-state transitions (K-1) to return encoder to all-zero state S_0. Thus, given input sequence length L, trellis must have $L+m$ stages, starting and stopping at S_0.

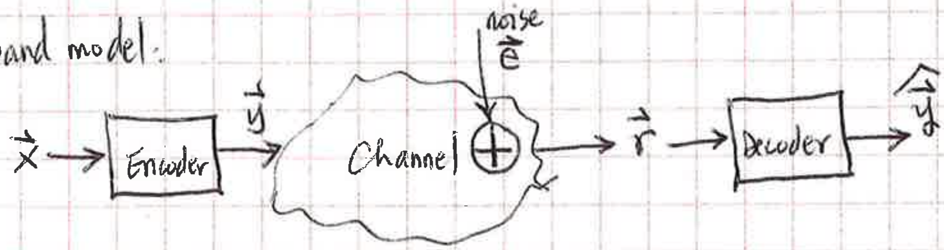
(5) There are 2^L distinct paths through trellis, each corresponding to a length $n(L+m)$ possible codeword
(We'll see how Viterbi addresses this issue)

Below is the trellis diagram for the encoder in our example: (copied from lecture notes)



The Decoding Problem:

Recall our baseband model:



(*) ML decoder:
~ selects estimate $\hat{\vec{y}}$ that maximizes $P(\vec{r}\,|\,\vec{y})$

Naive ML decoder: compares $\vec{r}$ with 2^L possible L-bit sequences (e.g. for 16-bit $\vec{r}$, need 64K compares)

Viterbi decoder: Need not consider all 2^L possible paths in trellis; at any stage, only 2^m paths need be retained

Assumes: channel is memoryless (i.e. errors uncorrelated from stage-to-stage)

Thm: The path selected by the Viterbi decoder is the ML path! O(linear) vs O(exponential)

MIT

The Viterbi Algorithm:

Define: $S_{j,t}$ ~ node S_j at time t.

$PM(S_{j,t})$ ~ path metric of node S_j at time t.

VA:

1. Initialize $t=0$
   Set $PM(S_{0,0})=0$; $PM(S_{j,0})=\infty \;\forall j\neq 0$.

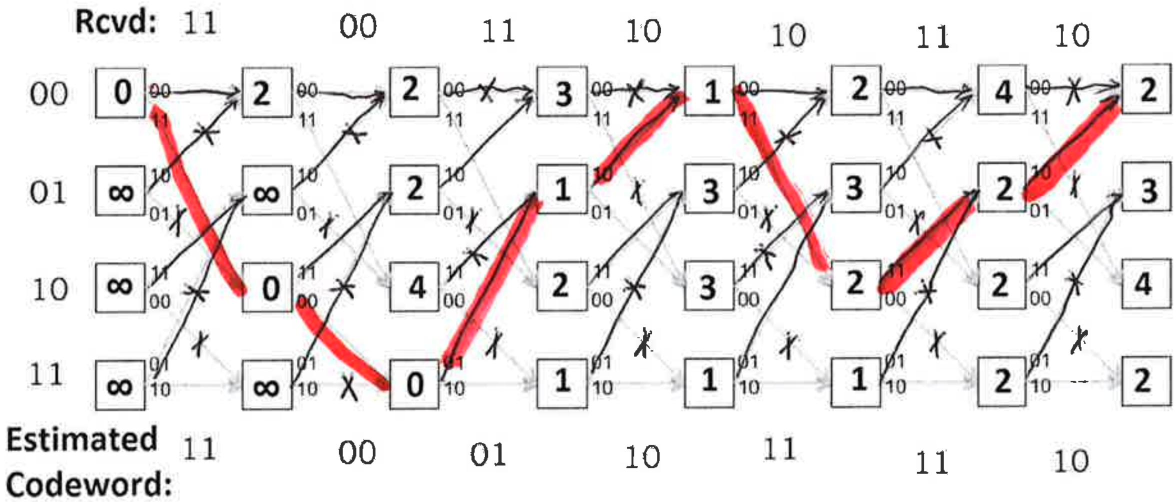
2. While $(t < L+m)$
   increment t.
   Compute PPMs (partial path metrics) $\forall$ paths entering each node.
   Set $PM(S_{j,t})$ = Best PPM entering node S_j at time t.
   Delete non-surviving branches.

3. Trace back from S_0 at time $t=L+m$, following surviving branches.
   The path thus defined is the <u>unique</u> ML codeword!

Example: Our example encoder encodes the message sequence $\vec{x}=(11001)$, generating the codeword $\vec{y}=(11, 00, 01, 10, 11, 11, 10)$. If $\vec{y}$ is transmitted over a noisy BSC, so that the received word is $\vec{r}=(11, 00, \underline{1}1, 10, 1\underline{0}, 11, 10)$. Use the Viterbi decoder to obtain the ML codeword $\hat{\vec{y}}$ and corresponding transmitted sequence $\hat{\vec{x}}$.

Soln: We'll use the constructed trellis with calculated PPMs, BMs and PMs using HD (d_H) as our metric (next page).

(*) We obtain $\hat{\vec{y}}=(11, 00, 01, 10, 11, 11, 10)$ which corrected the errors!
The transmitted estimated sequence is thus (11001) as expected.



HD

ML path minimizes $d_H(\vec{r}-\vec{y})$, so choose min. PM.

$PM(s_{i,j}) = \min\{PM(s_{i-j,j}) + BM(s_{i-j}, s_{i,j})\ \forall$ branches into $s_{i,j}\}$

where BM ~ $d_H(\vec{r}-\vec{y})$ at stage.