DSSS, Viterbi Algorithm, MAC

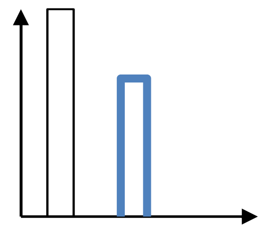
Jie Gao

9/17/09

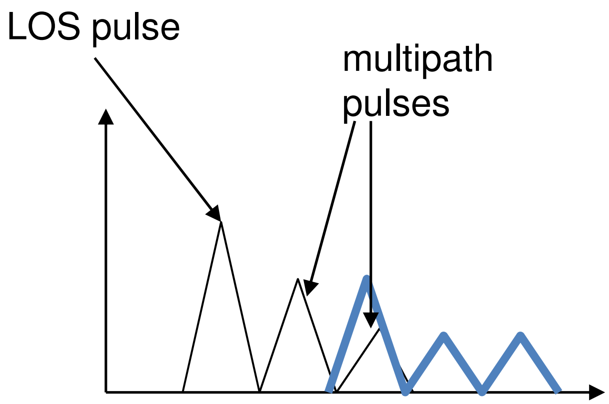
Acknowledgement: Slides borrowed from Richard Yang @ Yale.

Multipath Can Cause ISI

dispersed signal can cause interference between "neighbor" symbols, Inter Symbol Interference (ISI)



signal at sender



signal at receiver

LOS: Line Of Sight

Viterbi algorithm

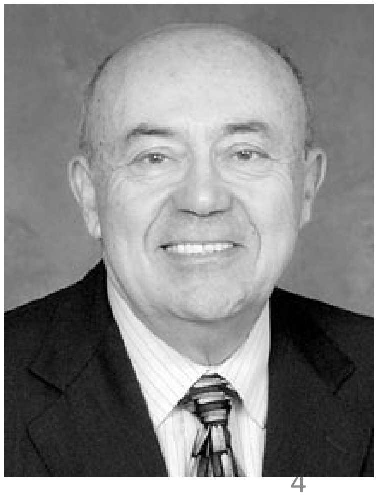
- Convolutional coding
- Applying Viterbi decoding algorithm to solve ISI problem: equalization.

ISI Problem Formulation

- The problem: given received y[m], m = 1, …, L+2, where L is frame size and assume 3 delay taps (it is easy to generalize to D taps):

  y[1] = x[1]h0 + w[1]
  y[2] = x[2]h0 + x[1]h1 + w[2]
  y[3] = x[3]h0 + x[2]h1 + x[3] h2 + w[3]
  y[4] = x[4]h0 + x[3]h1 + x[2] h2 + w[4]
  y[5] = x[5]h0 + x[4]h1 + x[3] h2 + w[5]
  …
  y[L] = x[L]h0 + x[L-1]h1 + x[L-2]h2 + w[L]
  y[L+1] = x[L]h1 + x[L-1]h2 + w[L+1]
  y[L+2] = x[L]h2 + w[L+2]

determine x[1], x[2], … x[L]



http://en.wikipedia.org/wiki/Andrew_Viterbi

ISI Equalization:
Given y, what is x?

x

y[1] = x[1] h0 + w[1]
y[2] = x[2]h0 + x[1] h1 + w[2]
y[3] = x[3]h0 + x[2]h1 + x[3] h2 + w[3]
y[4] = x[4]h0 + x[3]h1 + x[2] h2 + w[4]
y[5] = x[5]h0 + x[4]h1 + x[3] h2 + w[5]
…
y[L] = x[L]h0 + x[L-1]h1 + x[L-2]h2 + w[L]
y[L+1] = x[L]h1 + x[L-1]h2 + w[L+1]
y[L+2] = x[L]h2 + w[L+2]

y

Solution Technique

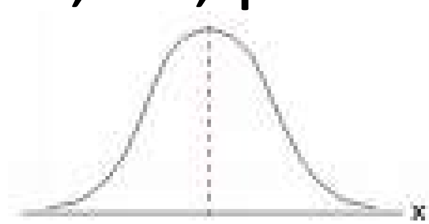
- Maximum likelihood detection:
  - if the transmitted sequence is x[1], …, x[L], then there is a likelihood we observe y[1], y[2], …, y[L+2]
  - we choose the x sequence such that the likelihood of observing y is the largest

y[1] = x[1] h0 + w[1]
y[2] = x[2]h0 + x[1] h1 + w[2]
y[3] = x[3]h0 + x[2]h1 + x[3] h2 + w[3]
y[4] = x[4]h0 + x[3]h1 + x[2] h2 + w[4]
y[5] = x[5]h0 + x[4]h1 + x[3] h2 + w[5]
…
y[L] = x[L]h0 + x[L-1]h1 + x[L-2]h2 + w[L]
y[L+1] = x[L]h1 + x[L-1]h2 + w[L+1]
y[L+2] = x[L]h2 + w[L+2]

Likelihood

- For given sequence x[1], x[2], …, x[L]
- Assume white noise, i.e, prob. w = z is

$$f(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{z^2}{2\sigma^2}}$$



- What is the likelihood (prob.) of observing y[1]?
  – it is the prob. of noise being w[1] = y[1] – x[1] h0

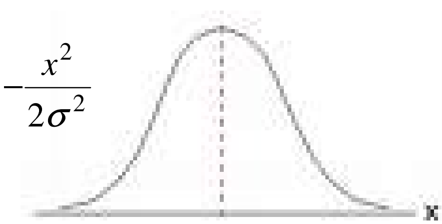
$$\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{w[1]^2}{2\sigma^2}}$$

Likelihood

- The likelihood of observing y[2]
  - it is the prob. of noise being w[2] = y[2] – x[2]h0 – x[1]h1, which is

$$\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{w[2]^2}{2\sigma^2}}$$

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}$$



- The overall likelihood of observing the whole y sequence (y[1], …, y[L+2]) is the product of the preceding probabilities

One Technique: Enumeration

foreach sequence (x[1], …, x[L])

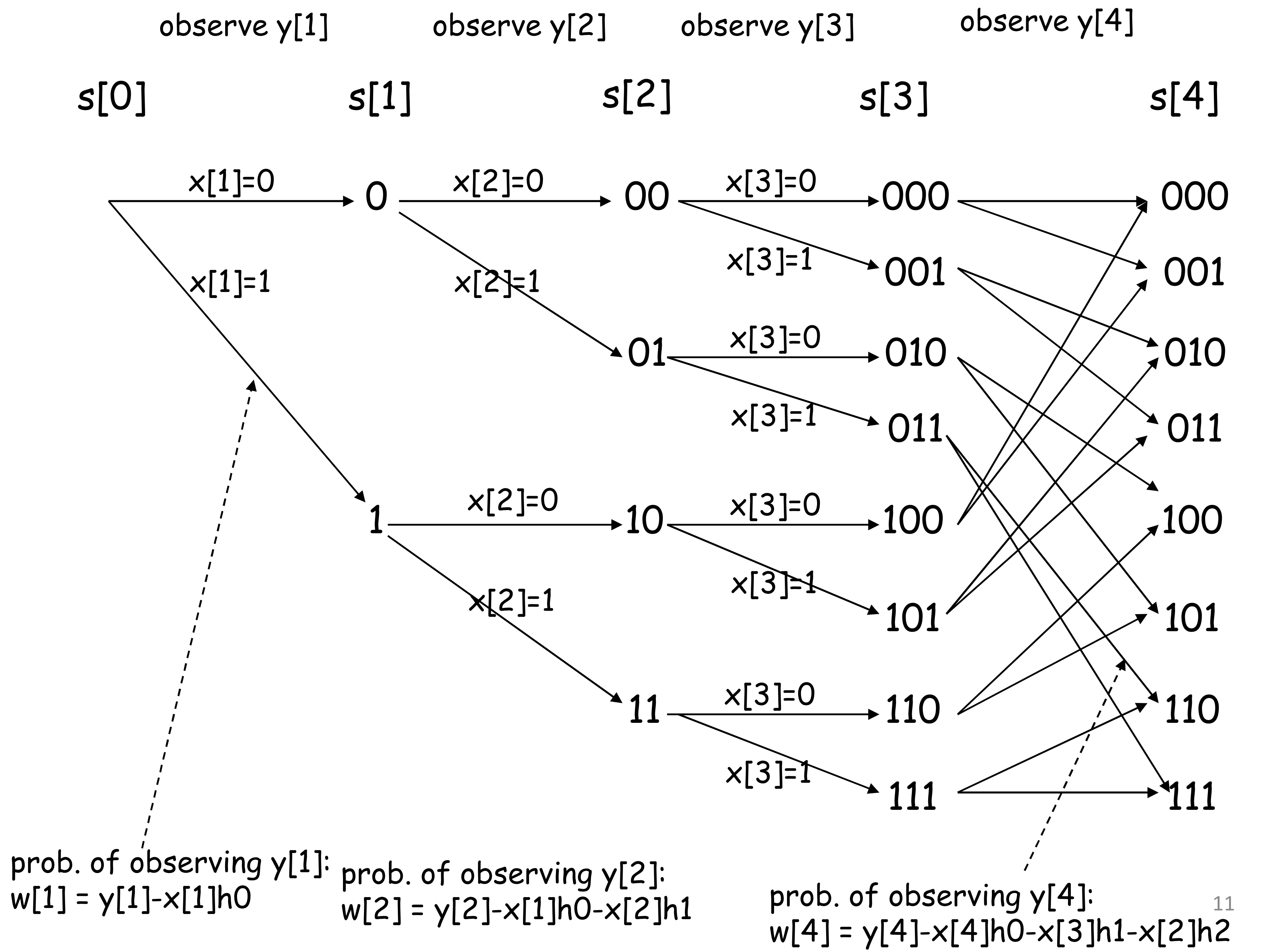
compute the likelihood of observing the y sequence

pick the x sequence with the highest likelihood

Computational complexity: exponential in L.

Viterbi Algorithm

- Objective: avoid the enumeration of the x sequences
- Key observation: the memory (state) of the wireless channel is only 3 (or generally D for D taps)
- Let s[0], s[1], … be the states of the channel as symbols are transmitted
  - s[0]: initial state---empty
  - s[1]: x[1] is transmitted, two possibilities: 0, or 1
  - s[2]: x[2] is transmitted, four possibilities: 00, 01, 10, 11
  - s[3]: x[3] is transmitted, eight possibilities: 000, 001, …, 111
  - s[4]: x[4] is transmitted, eight possibilities: 000, 001, …, 111
- We can construct a state transition diagram
- If we know the x sequence we can construct s, and vice versa


observe y[1]
observe y[2]
observe y[3]
observe y[4]
s[0]
s[1]
s[2]
s[3]
s[4]
x[1]=0
x[1]=1
0
1
x[2]=0
x[2]=1
x[2]=0
x[2]=1
00
01
10
11
x[3]=0
x[3]=1
x[3]=0
x[3]=1
x[3]=0
x[3]=1
x[3]=0
x[3]=1
000
001
010
011
100
101
110
111
000
001
010
011
100
101
110
111
prob. of observing y[1]:
w[1] = y[1]-x[1]h0
prob. of observing y[2]:
w[2] = y[2]-x[1]h0-x[2]h1
prob. of observing y[4]:
w[4] = y[4]-x[4]h0-x[3]h1-x[2]h2

Viterbi Algorithm

- Each path on the state-transition diagram corresponds to a x sequence
  - each edge has a probability
  - the product of the probabilities on the edges of a path corresponds to the likelihood that we observe y if x is the sequence sent

- Then the problem becomes identifying the path with the largest product of probabilities

Viterbi Algorithm: Largest Product to Shortest Path

- If we take $-\log$ of the probability of each edge, the problem becomes identifying the shortest path problem!
- Solve with dynamic programming.

Viterbi Algorithm: Summary

- Invented in 1967
- Utilized in CDMA, GSM, 802.11, Dial-up modem, and deep space communications

- Also commonly used in
  - speech recognition,
  - computational linguistics, and
  - bioinformatics

Original paper: Andrew J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm, April 1967
http://ieeexplore.ieee.org/search/wrapper.jsp?arnumber=1054010