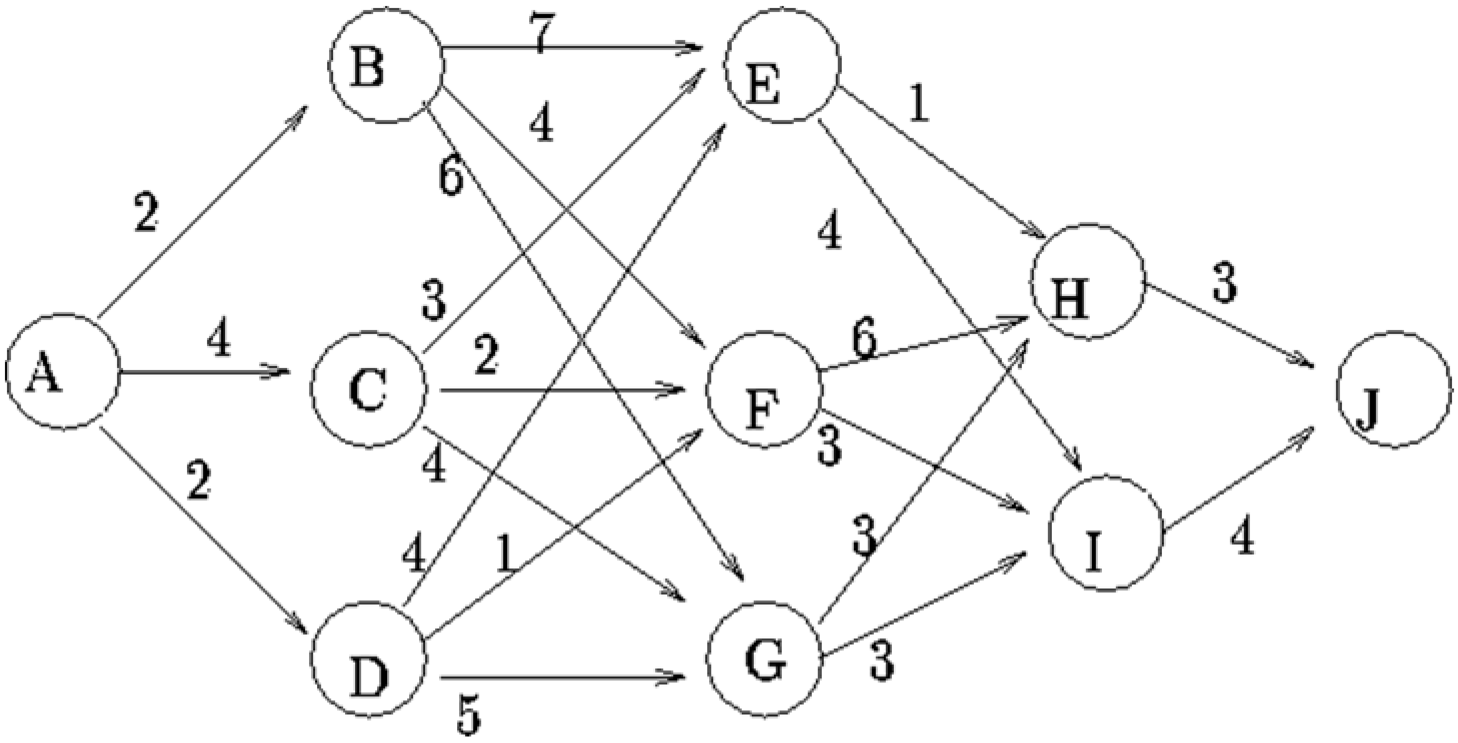
STA 5106
Computational Methods in Statistics I

Department of Statistics
Florida State University

Class 23
November 21, 2019

Review: Example Two

- **Shortest path problem:** Find the shortest path from A to J in the following road network.



Review: Dynamic Programming

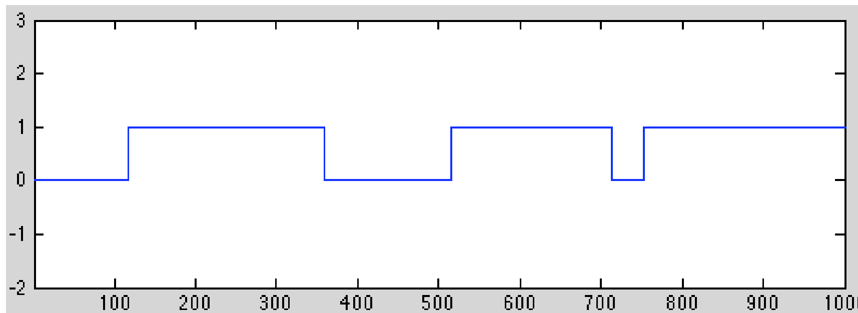
- Let A be the 1^{st} layer, J be the 5^{th} layer, and let a, b, c denote the choice in the 2^{nd}, 3^{rd}, and 4^{th} layers.

Backward dynamic programming:

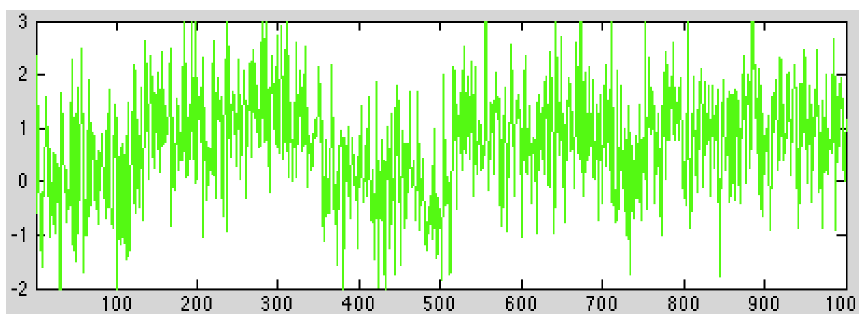
- If b = E, then c = H ($4 < 8$)
- If b = F, then c = I ($7 < 9$)
- If b = G, then c = H ($6 < 7$)
- If a = B, then b = E or F ($11 = 11 < 12$)
- If a = C, then b = E ($7 < 9 < 10$)
- If a = D, then b = E or F ($8 = 8 < 11$)
- Therefore, a = D, b = E or F, c = H or I ($10 < 11 < 13$); that is, the shortest path is ADEHJ or ADFIJ.

Function Reconstruction

- x(t): piecewise constant function



- y(t): noisy observation



- **Goal: Reconstruct x(t) from y(t).**

Discretized Case

- $x(t)$ is discretized as $x = (x_1, \ldots, x_N)$
- $\{x_i\}$ is binary and we assume

$$x_1 = \begin{cases} 0 & \text{with probability } 0.5 \\ 1 & \text{with probability } 0.5 \end{cases}$$

- Other x_i follow a Markovian transition:

$$\Pr(x_i \mid x_{i-1}) = p^{1_{x_i = x_{i-1}}} (1-p)^{1_{x_i \neq x_{i-1}}} = \begin{cases} p & \text{if } x_i = x_{i-1} \\ 1-p & \text{if } x_i \neq x_{i-1} \end{cases}$$

- That is,

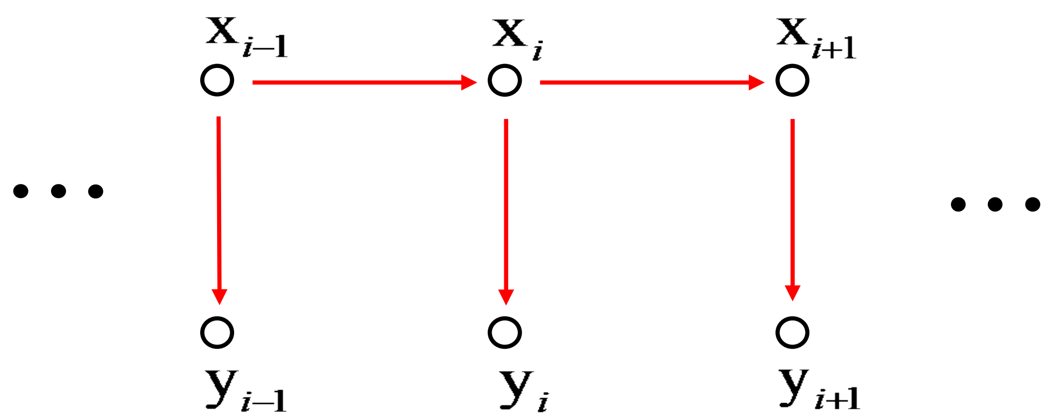
$$x_i = \begin{cases} x_{i-1} & \text{with probability } p \\ 1 - x_{i-1} & \text{with probability } 1-p \end{cases}$$

Discretized Case

- y = (y_1, …, y_N) is a noisy version of x:

$$y_i = x_i + \varepsilon_i, \qquad \varepsilon_i \sim N(0, \sigma^2)$$

- Graphical Model:



Likelihood, Prior, and Posterior

- Likelihood:

$$\Pr(\{y_i\}\mid\{x_i\})=\prod_{i=1}^{N}\Pr(y_i\mid x_i)=\prod_{i=1}^{N}\frac{1}{\sqrt{2\pi\sigma}}\exp(-\frac{(y_i-x_i)^2}{2\sigma^2})$$

- Prior:

$$\Pr(\{x_i\})=\Pr(x_1)\prod_{i=2}^{N}\Pr(x_i\mid x_{i-1})=\frac{1}{2}\prod_{i=2}^{N}p^{1_{x_i=x_{i-1}}}(1-p)^{1_{x_i\neq x_{i-1}}}$$

- Posterior:

$$\Pr(\{x_i\}\mid\{y_i\})=\Pr(\{y_i\}\mid\{x_i\})\Pr(\{x_i\})/\Pr(\{y_i\})$$

MAP Estimate

- Maximum A Posteriori (MAP) Estimate:

$$\{\hat{x}_i\} = \underset{\{x_i\}}{\operatorname{argmax}} \Pr(\{x_i\} \mid \{y_i\}) = \underset{\{x_i\}}{\operatorname{argmax}} \log(\Pr(\{x_i\} \mid \{y_i\}))$$

$$\log(\Pr(\{x_i\} \mid \{y_i\}))$$

$$= \log\left(\prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y_i - x_i)^2}{2\sigma^2}\right)\right) + \log\left(\frac{1}{2}\prod_{i=2}^{N} p^{1_{x_i = x_{i-1}}} (1-p)^{1_{x_i \neq x_{i-1}}}\right) + const$$

$$= \sum_{i=1}^{N} \left(-\frac{(y_i - x_i)^2}{2\sigma^2}\right) + \sum_{i=2}^{N} \log(1_{x_i = x_{i-1}} p + 1_{x_i \neq x_{i-1}} (1-p)) + const$$

- The maximum can be computed using **Dynamic Programming**.

Pseudo Code

1. $S(1,1) = -y(1)^2/(2\sigma^2)$, $S(1,2) = -(y(1)-1)^2/(2\sigma^2)$
2. for k from 2 to N

    for i from 0 to 1 (note: $x_k = 0$, or 1)

        h(1) = sum from 1 to k for $x_{k-1} = 0$

        h(2) = sum from 1 to k for $x_{k-1} = 1$

        if h(1) > h(2)

            $x_{k-1} = 0$ (given $x_k = i$), sum from 1 to k = h(1)

        else

            $x_{k-1} = 1$ (given $x_k = i$), sum from 1 to k = h(2)

        end

    end

end

Pseudo Code

3. if S(N,1) < S(N,2)

    $z_N = 1$ (z denotes estimated x)

   else

    $z_N = 0$

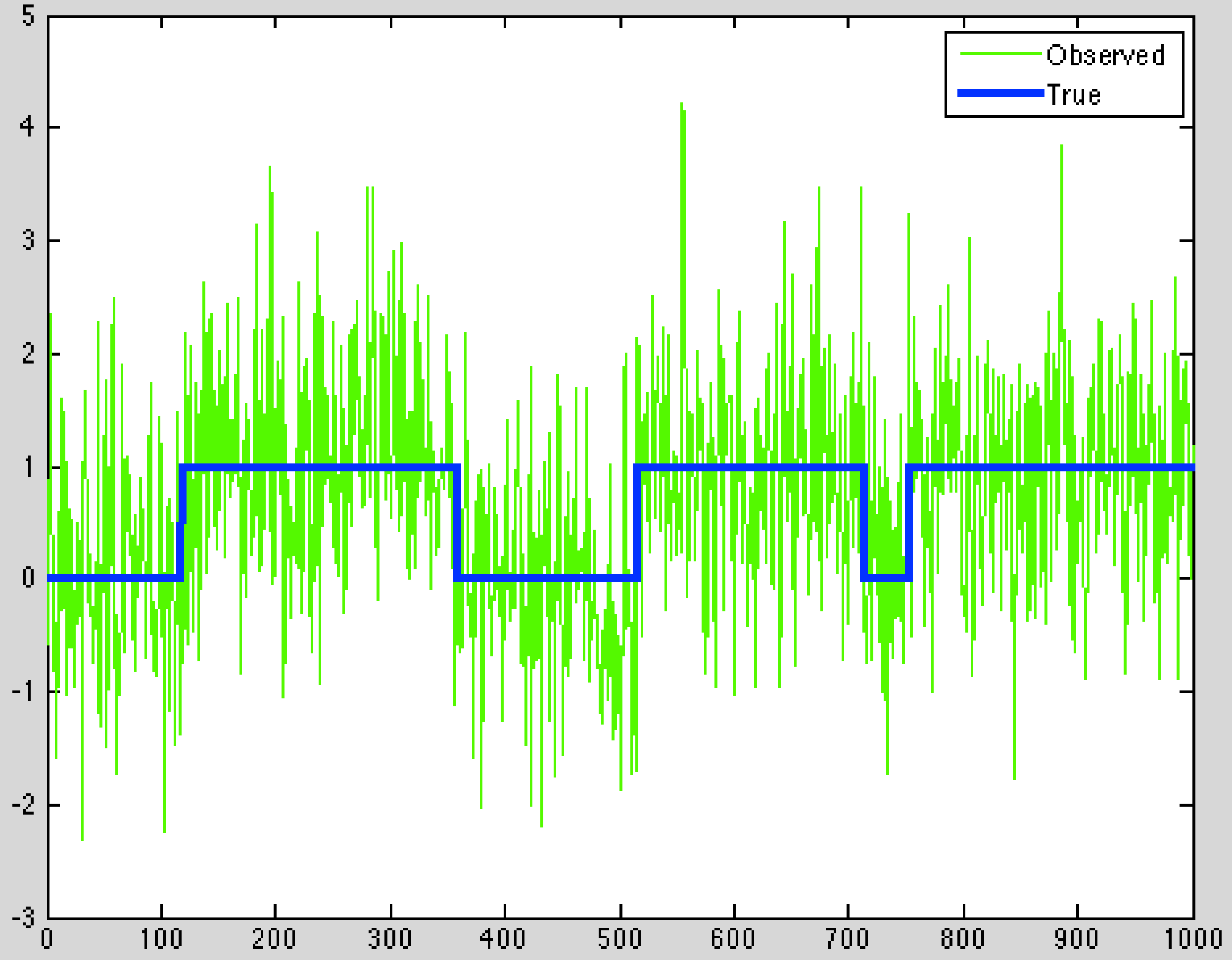
   end

4. for k from N – 1 to 1

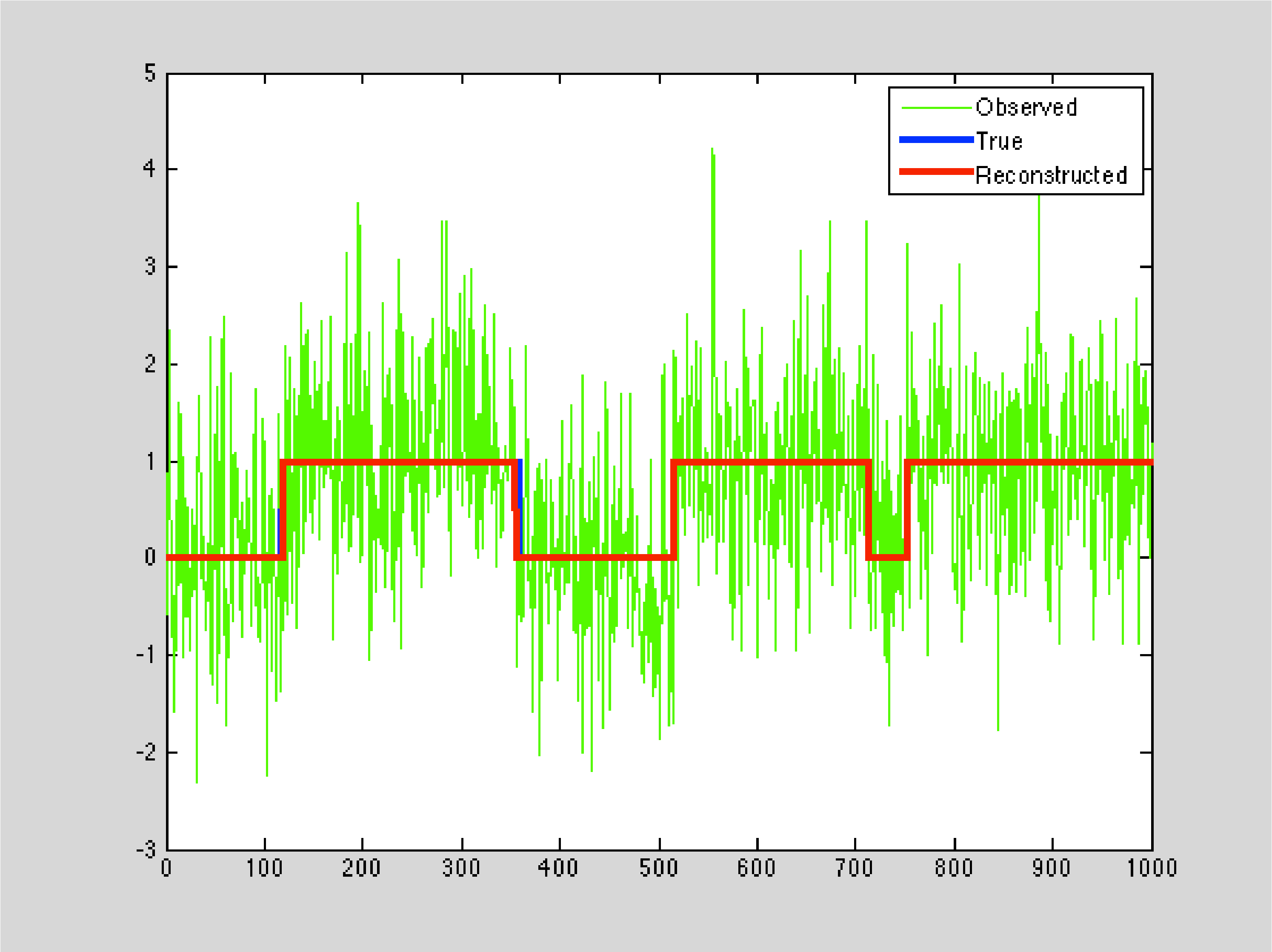
    $z_k =$ optimal x_{k-1} given x_k

   end

Example


Observed
True
5
4
3
2
1
0
-1
-2
-3
0
100
200
300
400
500
600
700
800
900
1000

Example



Observed
True
Reconstructed
5
4
3
2
1
0
-1
-2
-3
0
100
200
300
400
500
600
700
800
900
1000

Final Project

- **Out: Today**
- **Due: 12/12 (Thursday) by 8am (email submission).**
- **Students who did not finish Python homework:**
  - **Turn in your final report in the full format: introduction, methodology, programming code, results, and conclusions.**
- **One bonus point for students who finished Python homework:**
  - **Only for parameters p = 0.99 and σ = 1.**
  - **Submit a two-page report**
    1. **Reconstruction plot with observed, true, and reconstructed**
    2. **Your programming code**