# Hidden Markov Models

Outline

I. Most likely explanation

II. Simplified matrix algorithms

III. HMM application to robot localization

* Figures/images are from the textbook site.

# Most Likely Weather Sequence

Umbrella sequence on the guard's first five days: [*true*, *true*, *false*, *true*, *true*]

**Question** What weather sequence is most likely?

# Most Likely Weather Sequence

Umbrella sequence on the guard's first five days: $[true, true, false, true, true]$

**Question** What weather sequence is most likely?

- $2^5$ possible weather sequences to pick from.

# Most Likely Weather Sequence

Umbrella sequence on the guard's first five days: $[true, true, false, true, true]$

**Question** What weather sequence is most likely?

♣ $2^5$ possible weather sequences to pick from.

♠ Use smoothing to find posterior distribution for weather at each time step, and select at the step the weather that scores the highest.

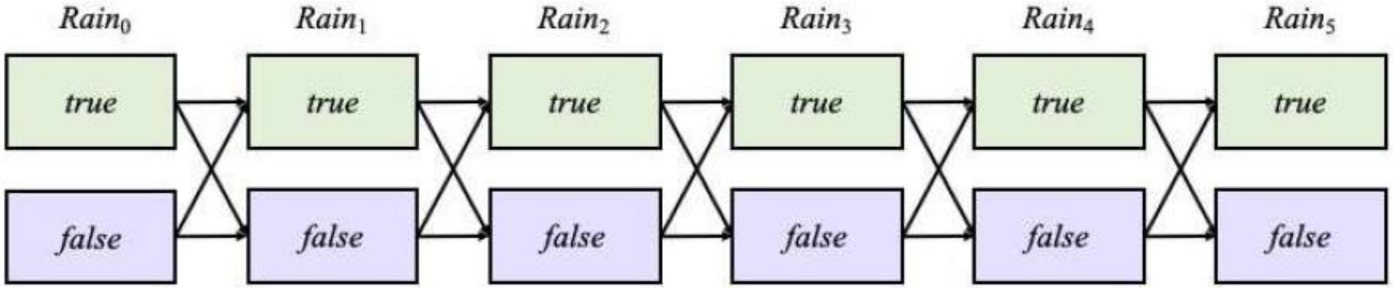
- Posterior distribution by smoothing are distributed over single time steps.
- Joint probabilities over all the time steps must be considered to find the most likely sequence.

# Most Likely Weather Sequence

Umbrella sequence on the guard's first five days: [*true*, *true*, *false*, *true*, *true*]

**Question** What weather sequence is most likely?

- ♣ $2^5$ possible weather sequences to pick from.
- ♠ Use smoothing to find posterior distribution for weather at each time step, and select at the step the weather that scores the highest.
  - Posterior distribution by smoothing are distributed over single time steps.
  - Joint probabilities over all the time steps must be considered to find the most likely sequence.
- ♦ View each sequence as a path through a graph whose nodes are the possible states at each time step.

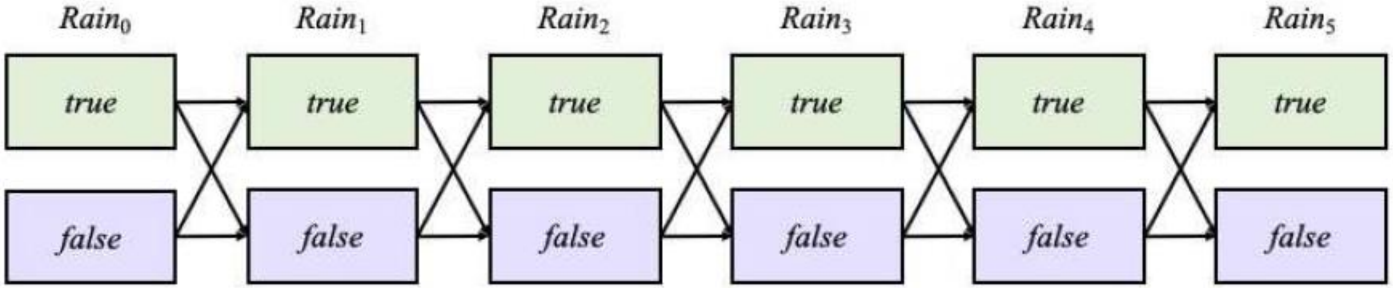


# Most Likely Weather Sequence

Umbrella sequence on the guard's first five days: [*true*, *true*, *false*, *true*, *true*]

**Question** What weather sequence is most likely?

- ♣ $2^5$ possible weather sequences to pick from.
- ♠ Use smoothing to find posterior distribution for weather at each time step, and select at the step the weather that scores the highest.
  - Posterior distribution by smoothing are distributed over single time steps.
  - Joint probabilities over all the time steps must be considered to find the most likely sequence.
- ♦ View each sequence as a path through a graph whose nodes are the possible states at each time step.




- ♦ Find the most likely path.

# Computing the Most Likely Path

Likelihood of a path is the product of

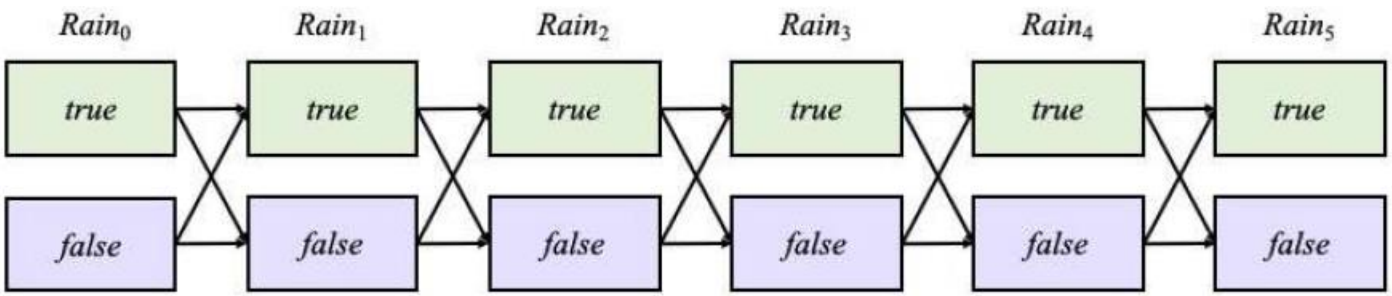
- the transition probabilities along the path,
- and the probabilities of the given observations at each state.

# Computing the Most Likely Path

Likelihood of a path is the product of

- the transition probabilities along the path,
- and the probabilities of the given observations at each state.

Consider paths that reach the state $Rain_5 = true$

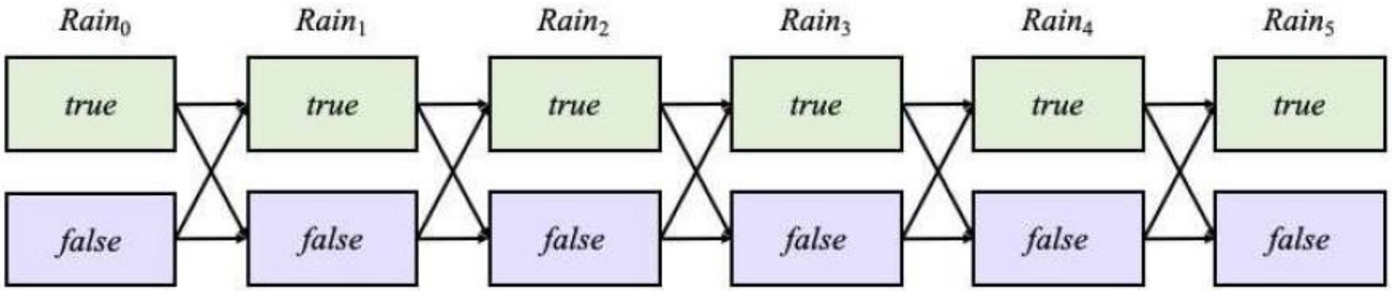


# Computing the Most Likely Path

Likelihood of a path is the product of

- the transition probabilities along the path,
- and the probabilities of the given observations at each state.

Consider paths that reach the state $Rain_5 = true$



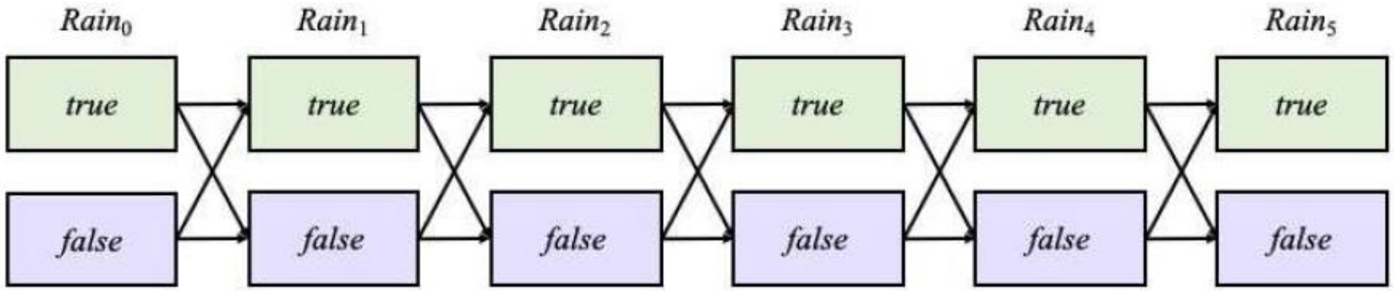

- The most likely path to $Rain_5 = true$ consists of the most likely path to some state at time 4 (optimal substructure) followed by a transition.

# Computing the Most Likely Path

Likelihood of a path is the product of

- the transition probabilities along the path,
- and the probabilities of the given observations at each state.

Consider paths that reach the state $Rain_5 = true$




- The most likely path to $Rain_5 = true$ consists of the most likely path to some state at time 4 (optimal substructure) followed by a transition.
- The state at time 4 that gets chosen maximizes the likelihood of the path to $Rain_5 = true$ .

# Recurrence

Define

$$\boldsymbol{m}_{1:k} = \max_{\boldsymbol{x}_{1:k-1}} \boldsymbol{P}(\boldsymbol{x}_{1:k-1}, \boldsymbol{X}_k, \boldsymbol{e}_{1:k})$$

# Recurrence

Define

$$\boldsymbol{m}_{1:k} = \max_{\boldsymbol{x}_{1:k-1}} \boldsymbol{P}(\boldsymbol{x}_{1:k-1}, \boldsymbol{X}_k, \boldsymbol{e}_{1:k})$$

We derive a recurrence for the term as follows:

$$\boldsymbol{m}_{1:k+1} = \max_{\boldsymbol{x}_{1:k}} \boldsymbol{P}(\boldsymbol{x}_{1:k}, \boldsymbol{X}_{k+1}, \boldsymbol{e}_{1:k+1})$$

# Recurrence

Define

$$\boldsymbol{m}_{1:k} = \max_{\boldsymbol{x}_{1:k-1}} \boldsymbol{P}(\boldsymbol{x}_{1:k-1}, \boldsymbol{X}_k, \boldsymbol{e}_{1:k})$$

We derive a recurrence for the term as follows:

$$\boldsymbol{m}_{1:k+1} = \max_{\boldsymbol{x}_{1:k}} \boldsymbol{P}(\boldsymbol{x}_{1:k}, \boldsymbol{X}_{k+1}, \boldsymbol{e}_{1:k+1})$$

$$= \max_{\boldsymbol{x}_{1:k}} \boldsymbol{P}(\boldsymbol{x}_{1:k}, \boldsymbol{X}_{k+1}, \boldsymbol{e}_{1:k}, e_{k+1})$$

# Recurrence

Define

$$\boldsymbol{m}_{1:k} = \max_{\boldsymbol{x}_{1:k-1}} \boldsymbol{P}(\boldsymbol{x}_{1:k-1}, \boldsymbol{X}_k, \boldsymbol{e}_{1:k})$$

We derive a recurrence for the term as follows:

$$\begin{aligned}
\boldsymbol{m}_{1:k+1} &= \max_{\boldsymbol{x}_{1:k}} \boldsymbol{P}(\boldsymbol{x}_{1:k}, \boldsymbol{X}_{k+1}, \boldsymbol{e}_{1:k+1}) \\
&= \max_{\boldsymbol{x}_{1:k}} \boldsymbol{P}(\boldsymbol{x}_{1:k}, \boldsymbol{X}_{k+1}, \boldsymbol{e}_{1:k}, e_{k+1}) \\
&= \max_{\boldsymbol{x}_{1:k}} \boldsymbol{P}(e_{k+1} \mid \boldsymbol{x}_{1:k}, \boldsymbol{X}_{k+1}, \boldsymbol{e}_{1:k}) \boldsymbol{P}(\boldsymbol{x}_{1:k}, \boldsymbol{X}_{k+1}, \boldsymbol{e}_{1:k})
\end{aligned}$$

# Recurrence

Define

$$\boldsymbol{m}_{1:k} = \max_{\boldsymbol{x}_{1:k-1}} \boldsymbol{P}(\boldsymbol{x}_{1:k-1}, \boldsymbol{X}_k, \boldsymbol{e}_{1:k})$$

We derive a recurrence for the term as follows:

$$\begin{aligned}
\boldsymbol{m}_{1:k+1} &= \max_{\boldsymbol{x}_{1:k}} \boldsymbol{P}(\boldsymbol{x}_{1:k}, \boldsymbol{X}_{k+1}, \boldsymbol{e}_{1:k+1}) \\
&= \max_{\boldsymbol{x}_{1:k}} \boldsymbol{P}(\boldsymbol{x}_{1:k}, \boldsymbol{X}_{k+1}, \boldsymbol{e}_{1:k}, e_{k+1}) \\
&= \max_{\boldsymbol{x}_{1:k}} \boldsymbol{P}(e_{k+1} \mid \boldsymbol{x}_{1:k}, \boldsymbol{X}_{k+1}, \boldsymbol{e}_{1:k}) \boldsymbol{P}(\boldsymbol{x}_{1:k}, \boldsymbol{X}_{k+1}, \boldsymbol{e}_{1:k}) \\
&= \boldsymbol{P}(e_{k+1} \mid \boldsymbol{X}_{k+1}) \max_{\boldsymbol{x}_{1:k}} \boldsymbol{P}(\boldsymbol{X}_{k+1} \mid \boldsymbol{x}_k) P(\boldsymbol{x}_{1:k}, \boldsymbol{e}_{1:k})
\end{aligned}$$

# Recurrence

Define

$$\boldsymbol{m}_{1:k} = \max_{\boldsymbol{x}_{1:k-1}} \boldsymbol{P}(\boldsymbol{x}_{1:k-1}, \boldsymbol{X}_k, \boldsymbol{e}_{1:k})$$

We derive a recurrence for the term as follows:

$$\begin{aligned}
\boldsymbol{m}_{1:k+1} &= \max_{\boldsymbol{x}_{1:k}} \boldsymbol{P}(\boldsymbol{x}_{1:k}, \boldsymbol{X}_{k+1}, \boldsymbol{e}_{1:k+1}) \\
&= \max_{\boldsymbol{x}_{1:k}} \boldsymbol{P}(\boldsymbol{x}_{1:k}, \boldsymbol{X}_{k+1}, \boldsymbol{e}_{1:k}, e_{k+1}) \\
&= \max_{\boldsymbol{x}_{1:k}} \boldsymbol{P}(e_{k+1} \mid \boldsymbol{x}_{1:k}, \boldsymbol{X}_{k+1}, \boldsymbol{e}_{1:k}) \boldsymbol{P}(\boldsymbol{x}_{1:k}, \boldsymbol{X}_{k+1}, \boldsymbol{e}_{1:k}) \\
&= \boldsymbol{P}(e_{k+1} \mid \boldsymbol{X}_{k+1}) \max_{\boldsymbol{x}_{1:k}} \boldsymbol{P}(\boldsymbol{X}_{k+1} \mid \boldsymbol{x}_k) P(\boldsymbol{x}_{1:k}, \boldsymbol{e}_{1:k}) \\
&= \boldsymbol{P}(e_{k+1} \mid \boldsymbol{X}_{k+1}) \max_{\boldsymbol{x}_k} \boldsymbol{P}(\boldsymbol{X}_{k+1} \mid \boldsymbol{x}_k) \max_{\boldsymbol{x}_{1:k-1}} P(\boldsymbol{x}_{1:k-1}, \boldsymbol{x}_k, \boldsymbol{e}_{1:k})
\end{aligned}$$

# Recurrence

Define

$$\boldsymbol{m}_{1:k} = \max_{\boldsymbol{x}_{1:k-1}} \boldsymbol{P}(\boldsymbol{x}_{1:k-1}, \boldsymbol{X}_k, \boldsymbol{e}_{1:k})$$

Entry for $\boldsymbol{X}_k = \boldsymbol{x}_k$

We derive a recurrence for the term as follows:

$$\boldsymbol{m}_{1:k+1} = \max_{\boldsymbol{x}_{1:k}} \boldsymbol{P}(\boldsymbol{x}_{1:k}, \boldsymbol{X}_{k+1}, \boldsymbol{e}_{1:k+1})$$

$$= \max_{\boldsymbol{x}_{1:k}} \boldsymbol{P}(\boldsymbol{x}_{1:k}, \boldsymbol{X}_{k+1}, \boldsymbol{e}_{1:k}, e_{k+1})$$

$$= \max_{\boldsymbol{x}_{1:k}} \boldsymbol{P}(e_{k+1} \mid \boldsymbol{x}_{1:k}, \boldsymbol{X}_{k+1}, \boldsymbol{e}_{1:k}) \boldsymbol{P}(\boldsymbol{x}_{1:k}, \boldsymbol{X}_{k+1}, \boldsymbol{e}_{1:k})$$

$$= \boldsymbol{P}(e_{k+1} \mid \boldsymbol{X}_{k+1}) \max_{\boldsymbol{x}_{1:k}} \boldsymbol{P}(\boldsymbol{X}_{k+1} \mid \boldsymbol{x}_k) P(\boldsymbol{x}_{1:k}, \boldsymbol{e}_{1:k})$$

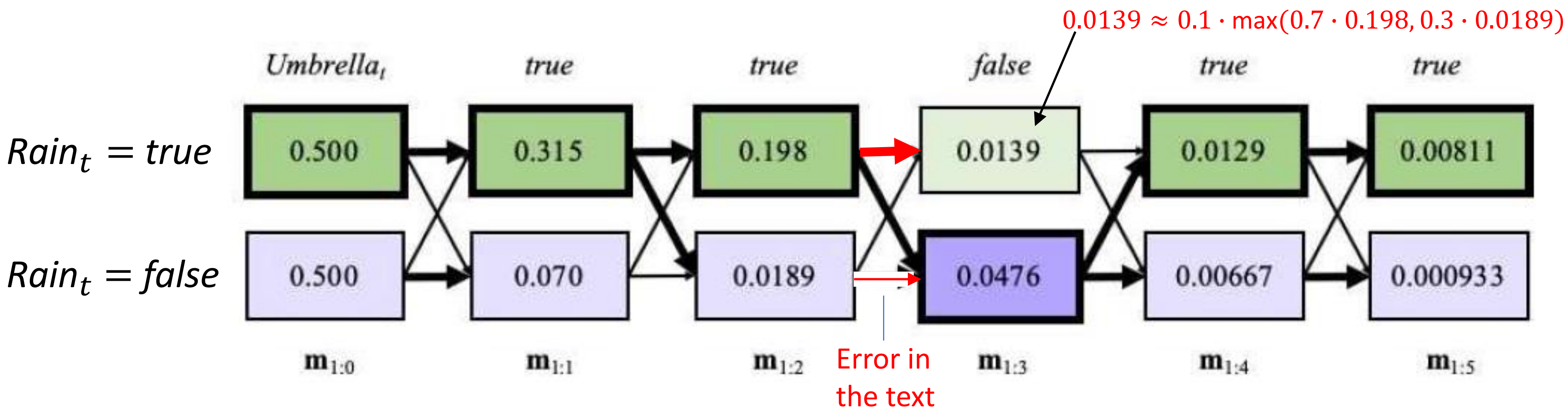
$$= \boldsymbol{P}(e_{k+1} \mid \boldsymbol{X}_{k+1}) \max_{\boldsymbol{x}_k} \boldsymbol{P}(\boldsymbol{X}_{k+1} \mid \boldsymbol{x}_k) \max_{\boldsymbol{x}_{1:k-1}} P(\boldsymbol{x}_{1:k-1}, \boldsymbol{x}_k, \boldsymbol{e}_{1:k})$$

# Viterbi Algorithm

$$\boldsymbol{m}_{1:k} = \max_{\boldsymbol{x}_{1:k-1}} \boldsymbol{P}(\boldsymbol{x}_{1:k-1}, \boldsymbol{X}_k, \boldsymbol{e}_{1:k})$$

- $\boldsymbol{m}_{1:0} = \boldsymbol{P}(\boldsymbol{X}_0) \rightarrow \boldsymbol{m}_{1:1} \rightarrow \cdots \rightarrow \boldsymbol{m}_{1:t}$

- To identify the actual sequence, record, for each state, the best state (i.e., predecessor) that leads to it.

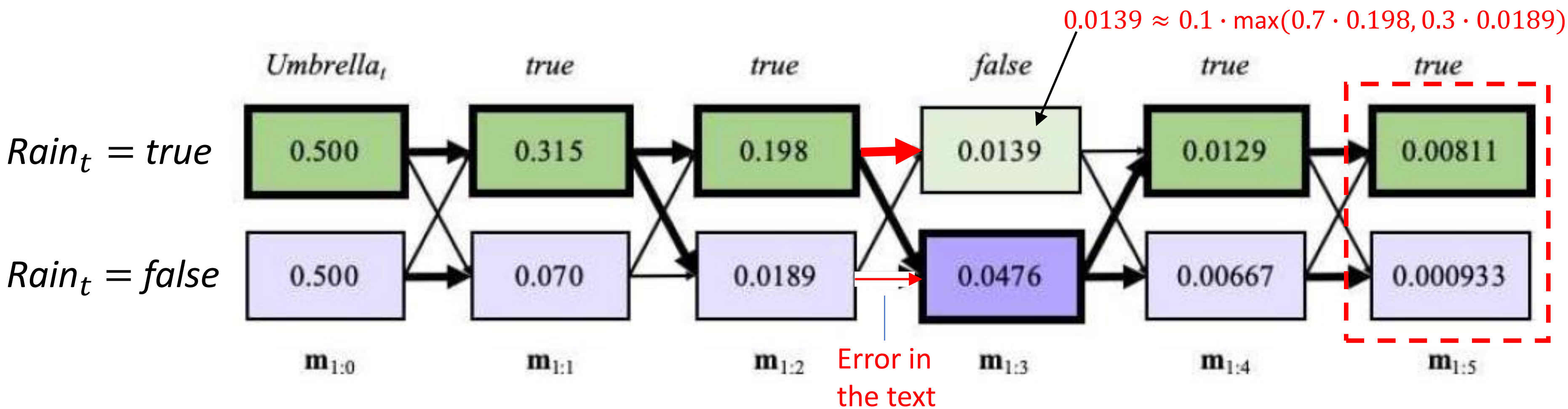


# Viterbi Algorithm

$$\boldsymbol{m}_{1:k} = \max_{\boldsymbol{x}_{1:k-1}} \boldsymbol{P}(\boldsymbol{x}_{1:k-1}, \boldsymbol{X}_k, \boldsymbol{e}_{1:k})$$

- $\boldsymbol{m}_{1:0} = \boldsymbol{P}(\boldsymbol{X}_0) \rightarrow \boldsymbol{m}_{1:1} \rightarrow \cdots \rightarrow \boldsymbol{m}_{1:t}$

- To identify the actual sequence, record, for each state, the best state (i.e., predecessor) that leads to it.



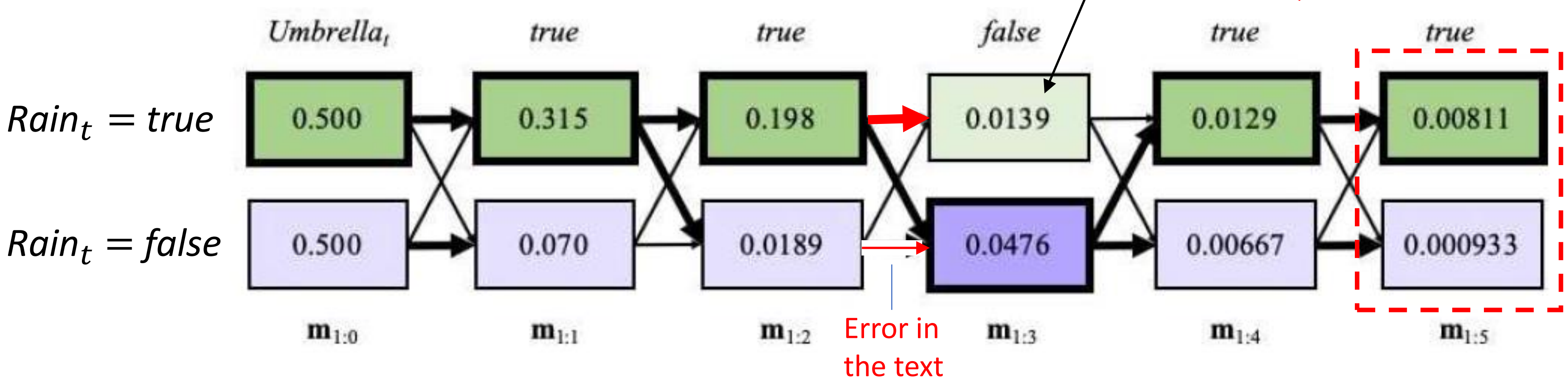
# Viterbi Algorithm

$$\boldsymbol{m}_{1:k} = \max_{\boldsymbol{x}_{1:k-1}} \boldsymbol{P}(\boldsymbol{x}_{1:k-1}, \boldsymbol{X}_k, \boldsymbol{e}_{1:k})$$

- $\boldsymbol{m}_{1:0} = \boldsymbol{P}(\boldsymbol{X}_0) \rightarrow \boldsymbol{m}_{1:1} \rightarrow \cdots \rightarrow \boldsymbol{m}_{1:t}$

- To identify the actual sequence, record, for each state, the best state (i.e., predecessor) that leads to it.

$0.0139 \approx 0.1 \cdot \max(0.7 \cdot 0.198, 0.3 \cdot 0.0189)$



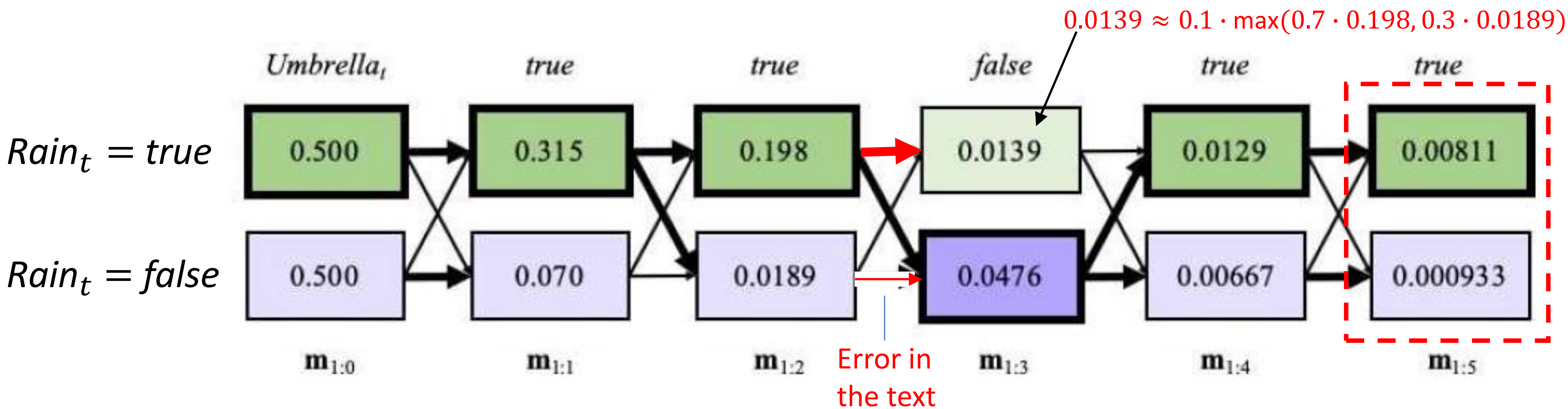

0.00811 is the maximum joint probability of any rain scenario on the first four days, rain on day 5, and the umbrella sequence [*true*, *true*, *false*, *true*, *true*] on the first five days. It is achieved by the weather sequence [*true*, *true*, *false*, *true*, *true*].

# Viterbi Algorithm

$$\boldsymbol{m}_{1:k} = \max_{\boldsymbol{x}_{1:k-1}} \boldsymbol{P}(\boldsymbol{x}_{1:k-1}, \boldsymbol{X}_k, \boldsymbol{e}_{1:k})$$

- $\boldsymbol{m}_{1:0} = \boldsymbol{P}(\boldsymbol{X}_0) \to \boldsymbol{m}_{1:1} \to \cdots \to \boldsymbol{m}_{1:t}$

- To identify the actual sequence, record, for each state, the best state (i.e., predecessor) that leads to it.




0.00811 is the maximum joint probability of any rain scenario on the first four days, rain on day 5, and the umbrella sequence [*true*, *true*, *false*, *true*, *true*] on the first five days. It is achieved by the weather sequence [*true*, *true*, *false*, *true*, *true*].

0.000933 is the maximum joint probability of any rain scenario on the first four days, no rain on day 5, and the umbrella sequence [*true*, *true*, *false*, *true*, *true*] on the first five days. It is achieved by the weather sequence [*true*, *true*, *false*, *false*, *false*].

# Hidden Markov Models

- The state is described by a single, discrete random variable $X_t$.
- Its values are the possible states of the world.
- There can be many evidence variables, both discrete and continuous.

# Hidden Markov Models

- The state is described by a single, discrete random variable $X_t$.
- Its values are the possible states of the world.
- There can be many evidence variables, both discrete and continuous.

Denote the values of $X_t$ by $1, \ldots, S$.

# Hidden Markov Models

- The state is described by a single, discrete random variable $X_t$.
- Its values are the possible states of the world.
- There can be many evidence variables, both discrete and continuous.

Denote the values of $X_t$ by $1, \dots, S$.

Transition model:

$$\boldsymbol{T} = \begin{pmatrix} T_{11} & \cdots & T_{1S} \\ \vdots & \ddots & \vdots \\ T_{S1} & \cdots & T_{SS} \end{pmatrix}$$

# Hidden Markov Models

- The state is described by a single, discrete random variable $X_t$.
- Its values are the possible states of the world.
- There can be many evidence variables, both discrete and continuous.

Denote the values of $X_t$ by $1, \ldots, S$.

Transition model:

$$\boldsymbol{T} = \begin{pmatrix} T_{11} & \cdots & T_{1S} \\ \vdots & \ddots & \vdots \\ T_{S1} & \cdots & T_{SS} \end{pmatrix}$$

where $T_{ij} = \underbrace{P(X_t = j \mid X_{t-1} = i)}_{\text{probability of transition from state } i \text{ to state } j}$

# Hidden Markov Models

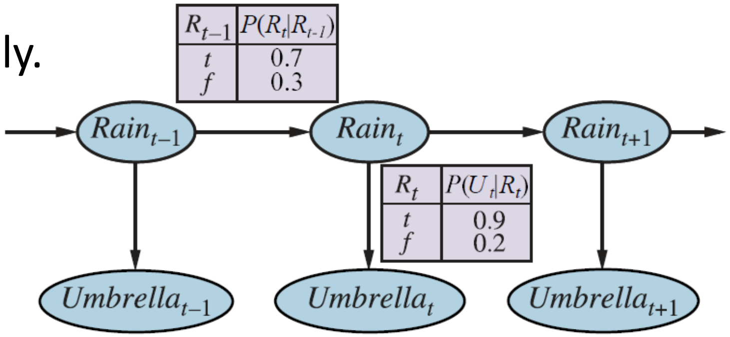
- The state is described by a single, discrete random variable $X_t$.
- Its values are the possible states of the world.
- There can be many evidence variables, both discrete and continuous.

Denote the values of $X_t$ by $1, \dots, S$.

Transition model:

$$\boldsymbol{T} = \begin{pmatrix} T_{11} & \cdots & T_{1S} \\ \vdots & \ddots & \vdots \\ T_{S1} & \cdots & T_{SS} \end{pmatrix} \quad \text{where } \underbrace{T_{ij} = P(X_t = j \mid X_{t-1} = i)}_{\text{probability of transition from state } i \text{ to state } j}$$

For the umbrella world, we denote the states *Rain* = *true* and *Rain* = *false* by 1 and 2, respectively.



# Hidden Markov Models

- The state is described by a single, discrete random variable $X_t$.
- Its values are the possible states of the world.
- There can be many evidence variables, both discrete and continuous.

Denote the values of $X_t$ by $1, \dots, S$.

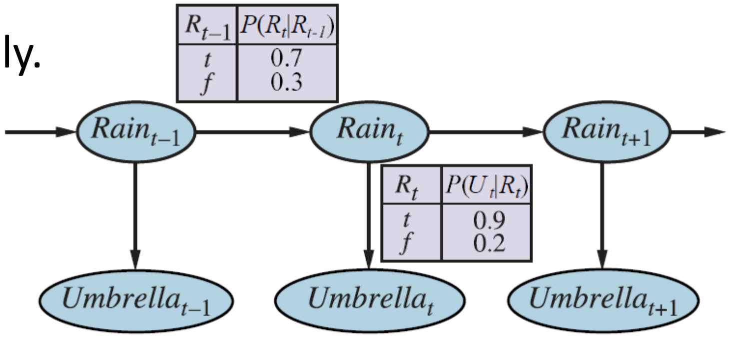
Transition model:

$$\boldsymbol{T} = \begin{pmatrix} T_{11} & \cdots & T_{1S} \\ \vdots & \ddots & \vdots \\ T_{S1} & \cdots & T_{SS} \end{pmatrix}$$

where $\underbrace{T_{ij} = P(X_t = j \mid X_{t-1} = i)}_{\text{probability of transition from state } i \text{ to state } j}$

For the umbrella world, we denote the states *Rain* = *true* and *Rain* = *false* by 1 and 2, respectively.

$$\boldsymbol{T} = P(X_t \mid X_{t-1}) = \begin{pmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{pmatrix}$$



# Observation Matrix

The value $e_t$ of the evidence variable $E_t$ is known at time $t$.

Specify, for each state $i = 1, \ldots, S$, the likelihood that it causes $e_t$ to appear.

$$P(e_t \mid X_t = i)$$

# Observation Matrix

The value $e_t$ of the evidence variable $E_t$ is known at time $t$.

Specify, for each state $i = 1, \dots, S$, the likelihood that it causes $e_t$ to appear.

$$P(e_t \mid X_t = i)$$

$$\text{Observation matrix } O_t = \underbrace{\begin{pmatrix} P(e_t \mid X_t = 1) & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & P(e_t \mid X_t = S) \end{pmatrix}}_{S \times S}$$

# Observation Matrix

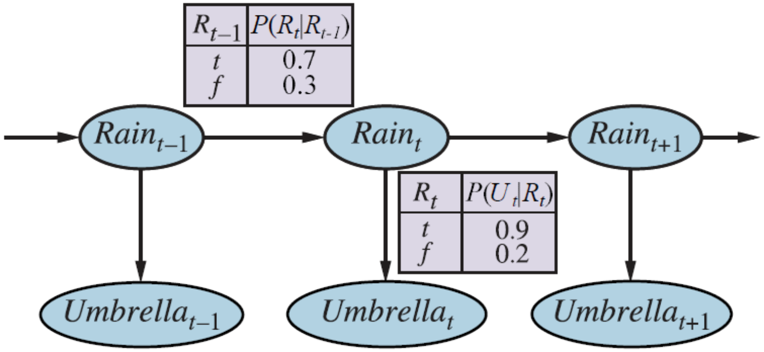
The value $e_t$ of the evidence variable $E_t$ is known at time $t$.

Specify, for each state $i = 1, \ldots, S$, the likelihood that it causes $e_t$ to appear.

$$P(e_t \mid X_t = i)$$

Observation matrix $O_t = \underbrace{\begin{pmatrix} P(e_t \mid X_t = 1) & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & P(e_t \mid X_t = S) \end{pmatrix}}_{S \times S}$

Suppose that $U_1 = \mathit{true}$ and $U_2 = \mathit{false}$.



# Observation Matrix

The value $e_t$ of the evidence variable $E_t$ is known at time $t$.

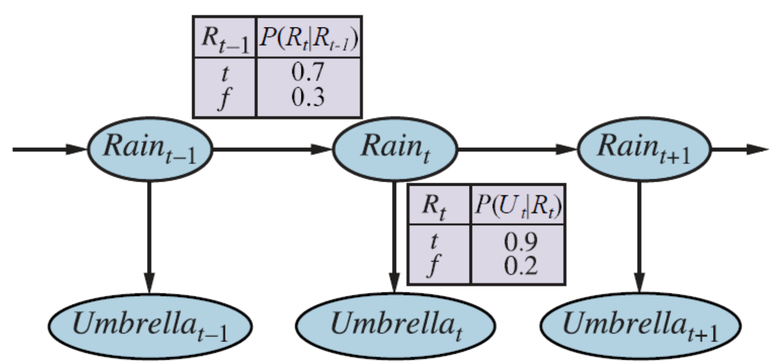
Specify, for each state $i = 1, \dots, S$, the likelihood that it causes $e_t$ to appear.

$$P(e_t \mid X_t = i)$$

$$\text{Observation matrix } O_t = \underbrace{\begin{pmatrix} P(e_t \mid X_t = 1) & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & P(e_t \mid X_t = S) \end{pmatrix}}_{S \times S}$$

Suppose that $U_1 = \textit{true}$ and $U_2 = \textit{false}$.

$$O_1 = \begin{pmatrix} 0.9 & 0 \\ 0 & 0.2 \end{pmatrix} \qquad O_3 = \begin{pmatrix} 0.1 & 0 \\ 0 & 0.8 \end{pmatrix}$$



# Matrix Formulations of Filtering and Smoothing

Forward message used in filtering: $\boldsymbol{f}_{1:t} \equiv \boldsymbol{P}(\boldsymbol{X}_t \mid \boldsymbol{e}_{1:t})$

$S \times 1$ matrix

# Matrix Formulations of Filtering and Smoothing

Forward message used in filtering: $\boldsymbol{f}_{1:t} \equiv \boldsymbol{P}(\boldsymbol{X}_t \mid \boldsymbol{e}_{1:t})$

$S \times 1$ matrix

$$\boldsymbol{f}_{1:t+1} \equiv \boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t+1}) = \alpha \boldsymbol{P}(\boldsymbol{e}_{t+1} \mid \boldsymbol{X}_{t+1}) \sum_{\boldsymbol{x}_t} \boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{x}_t) P(\boldsymbol{x}_t \mid \boldsymbol{e}_{1:t})$$

# Matrix Formulations of Filtering and Smoothing

Forward message used in filtering: $\boldsymbol{f}_{1:t} \equiv \boldsymbol{P}(\boldsymbol{X}_t \mid \boldsymbol{e}_{1:t})$

$S \times 1$ matrix

$$\boldsymbol{f}_{1:t+1} \equiv \boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t+1}) = \alpha \boldsymbol{P}(\boldsymbol{e}_{t+1} \mid \boldsymbol{X}_{t+1}) \sum_{\boldsymbol{x}_t} \boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{x}_t) P(\boldsymbol{x}_t \mid \boldsymbol{e}_{1:t})$$

Rewrite pointwise products as matrix products.

$$\boxed{\boldsymbol{f}_{1:t+1} = \alpha \boldsymbol{O}_{t+1} \boldsymbol{T}^T \boldsymbol{f}_{1:t}} \qquad \text{(filtering)}$$

# Matrix Formulations of Filtering and Smoothing

Forward message used in filtering: $\boldsymbol{f}_{1:t} \equiv \boldsymbol{P}(\boldsymbol{X}_t \mid \boldsymbol{e}_{1:t})$

$S \times 1$ matrix

$$\boldsymbol{f}_{1:t+1} \equiv \boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t+1}) = \alpha \boldsymbol{P}(\boldsymbol{e}_{t+1} \mid \boldsymbol{X}_{t+1}) \sum_{\boldsymbol{x}_t} \boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{x}_t) P(\boldsymbol{x}_t \mid \boldsymbol{e}_{1:t})$$

Rewrite pointwise products as matrix products.

$$\Downarrow$$

$$\boxed{\boldsymbol{f}_{1:t+1} = \alpha \boldsymbol{O}_{t+1} \boldsymbol{T}^T \boldsymbol{f}_{1:t}} \qquad \text{(filtering)}$$

Backward message used in smoothing: $\boldsymbol{b}_{k+1:t} \equiv \boldsymbol{P}(\boldsymbol{e}_{k+1:t} \mid \boldsymbol{X}_k)$

# Matrix Formulations of Filtering and Smoothing

Forward message used in filtering: $\boldsymbol{f}_{1:t} \equiv \boldsymbol{P}(\boldsymbol{X}_t \mid \boldsymbol{e}_{1:t})$

$S \times 1$ matrix

$$\boldsymbol{f}_{1:t+1} \equiv \boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t+1}) = \alpha \boldsymbol{P}(\boldsymbol{e}_{t+1} \mid \boldsymbol{X}_{t+1}) \sum_{\boldsymbol{x}_t} \boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{x}_t) P(\boldsymbol{x}_t \mid \boldsymbol{e}_{1:t})$$

Rewrite pointwise products
as matrix products.

$$\boldsymbol{f}_{1:t+1} = \alpha \boldsymbol{O}_{t+1} \boldsymbol{T}^T \boldsymbol{f}_{1:t}$$

(filtering)

Backward message used in smoothing: $\boldsymbol{b}_{k+1:t} \equiv \boldsymbol{P}(\boldsymbol{e}_{k+1:t} \mid \boldsymbol{X}_k)$

$$\boldsymbol{P}(\boldsymbol{e}_{k+1:t} \mid \boldsymbol{X}_k) = \sum_{\boldsymbol{x}_{k+1}} P(\boldsymbol{e}_{k+1} \mid \boldsymbol{x}_{k+1}) P(\boldsymbol{e}_{k+2:t} \mid \boldsymbol{x}_{k+1}) \boldsymbol{P}(\boldsymbol{x}_{k+1} \mid \boldsymbol{X}_k)$$

# Matrix Formulations of Filtering and Smoothing

Forward message used in filtering: $\boldsymbol{f}_{1:t} \equiv \boldsymbol{P}(\boldsymbol{X}_t \mid \boldsymbol{e}_{1:t})$

$S \times 1$ matrix

$$\boldsymbol{f}_{1:t+1} \equiv \boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t+1}) = \alpha \boldsymbol{P}(\boldsymbol{e}_{t+1} \mid \boldsymbol{X}_{t+1}) \sum_{\boldsymbol{x}_t} \boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{x}_t) P(\boldsymbol{x}_t \mid \boldsymbol{e}_{1:t})$$

Rewrite pointwise products
as matrix products.

$$\boldsymbol{f}_{1:t+1} = \alpha \boldsymbol{O}_{t+1} \boldsymbol{T}^T \boldsymbol{f}_{1:t}$$

(filtering)

Backward message used in smoothing: $\boldsymbol{b}_{k+1:t} \equiv \boldsymbol{P}(\boldsymbol{e}_{k+1:t} \mid \boldsymbol{X}_k)$

$$\boldsymbol{P}(\boldsymbol{e}_{k+1:t} \mid \boldsymbol{X}_k) = \sum_{\boldsymbol{x}_{k+1}} P(\boldsymbol{e}_{k+1} \mid \boldsymbol{x}_{k+1}) \, P(\boldsymbol{e}_{k+2:t} \mid \boldsymbol{x}_{k+1}) \boldsymbol{P}(\boldsymbol{x}_{k+1} \mid \boldsymbol{X}_k)$$

$$\boldsymbol{b}_{k+1:t} = \boldsymbol{T} \boldsymbol{O}_{k+1} \, \boldsymbol{b}_{k+2:t}$$

(smoothing)

# Complexities and Improvements

$$\boldsymbol{f}_{1:t+1} = \alpha \boldsymbol{O}_{t+1} \boldsymbol{T}^{T} \boldsymbol{f}_{1:t}$$

$$\boldsymbol{b}_{k+1:t} = \boldsymbol{T}\boldsymbol{O}_{k+1}\, \boldsymbol{b}_{k+\mathbf{2}:t}$$

# Complexities and Improvements

$$\boldsymbol{f}_{1:t+1} = \alpha \boldsymbol{O}_{t+1} \boldsymbol{T}^T \boldsymbol{f}_{1:t}$$

$$\boldsymbol{b}_{k+1:t} = \boldsymbol{T} \boldsymbol{O}_{k+1} \, \boldsymbol{b}_{k+\mathbf{2}:t}$$

Time complexity: $O(S^2 t)$ // $t$ steps, each requiring two rounds of $O(S^2)$ time
// multiplication of an $S \times S$ matrix by an $S$-vector.

# Complexities and Improvements

$$\boldsymbol{f}_{1:t+1} = \alpha \boldsymbol{O}_{t+1} \boldsymbol{T}^T \boldsymbol{f}_{1:t}$$

$$\boldsymbol{b}_{k+1:t} = \boldsymbol{T} \boldsymbol{O}_{k+1} \, \boldsymbol{b}_{k+\mathbf{2}:t}$$

Time complexity: $O(S^2 t)$ // $t$ steps, each requiring two rounds of $O(S^2)$ time
// multiplication of an $S \times S$ matrix by an $S$-vector.

Space complexity: $O(St)$ // $t > S$. The forward pass stores $t$ vectors of size $S$.

# Complexities and Improvements

$$\boldsymbol{f}_{1:t+1} = \alpha \boldsymbol{O}_{t+1} \boldsymbol{T}^T \boldsymbol{f}_{1:t}$$

$$\boldsymbol{b}_{k+1:t} = \boldsymbol{T}\boldsymbol{O}_{k+1}\, \boldsymbol{b}_{k+\mathbf{2}:t}$$

Time complexity: $O(S^2 t)$ // $t$ steps, each requiring two rounds of $O(S^2)$ time
// multiplication of an $S \times S$ matrix by an $S$-vector.

Space complexity: $O(St)$ // $t > S$. The forward pass stores $t$ vectors of size $S$.

Improvements:

- Allows smoothing to be carried out in constant space, independent of the lengths of the sequence.

# Complexities and Improvements

$$\boldsymbol{f}_{1:t+1} = \alpha \boldsymbol{O}_{t+1} \boldsymbol{T}^T \boldsymbol{f}_{1:t}$$

$$\boldsymbol{b}_{k+1:t} = \boldsymbol{T}\boldsymbol{O}_{k+1}\, \boldsymbol{b}_{k+\mathbf{2}:t}$$

Time complexity: $O(S^2 t)$ // $t$ steps, each requiring two rounds of $O(S^2)$ time
// multiplication of an $S \times S$ matrix by an $S$-vector.

Space complexity: $O(St)$ // $t > S$. The forward pass stores $t$ vectors of size $S$.

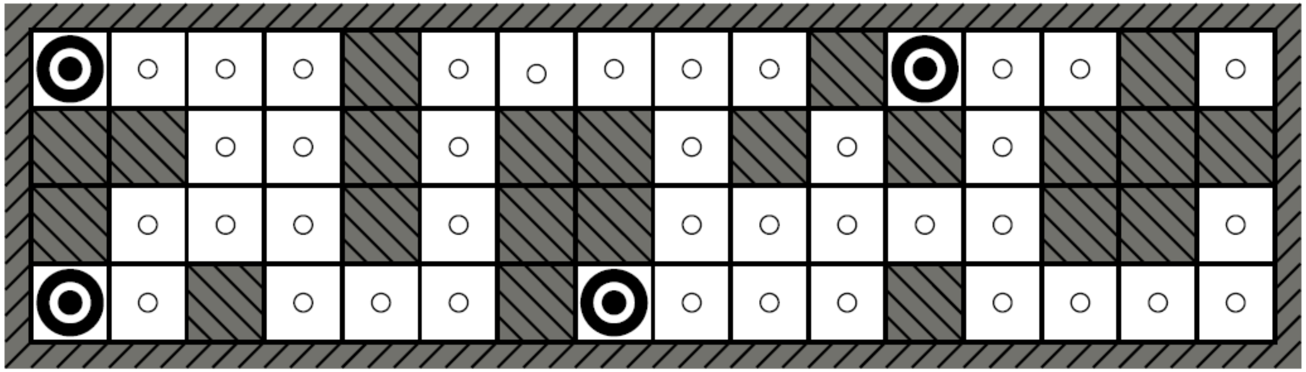
Improvements:

- Allows smoothing to be carried out in constant space, independent of the lengths of the sequence.
- Leads to an algorithm whose time complexity is independent of the length $d$ of the *lag*.

Smoothing at time $t - d$, where the current time is $t$.

# Revisiting the Localization Task

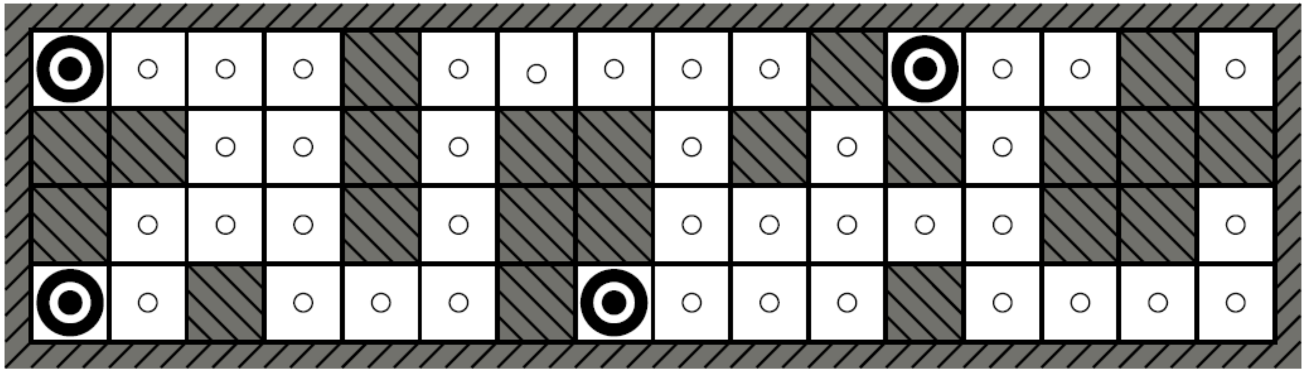
The robot had a single action Move and a perfect sensor to report whether obstacles are immediately to the north, east, south, and west.



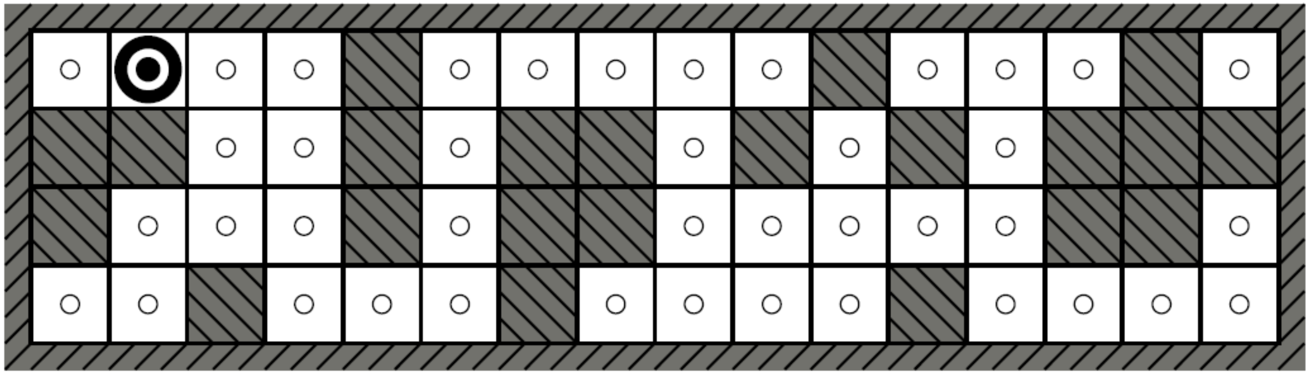
Possible locations of robot after $E_1 = \underset{\text{NESW}}{1011}$ (obstacles in the north, south, and west, but not east).

# Revisiting the Localization Task

The robot had a single action Move and a perfect sensor to report whether obstacles are immediately to the north, east, south, and west.



Possible locations of robot after $E_1 = \underset{\text{NESW}}{1011}$ (obstacles in the north, south, and west, but not east).



Possible locations after $E_1 = 1011$, $E_2 = 1010$.

# HMM Formulation

We now make the problem more realistic:

- Allow noise in sensing whether or not obstacles are immediately to the north, east, south, and west.

- The robot is equally likely to move to any adjacent square.

# HMM Formulation

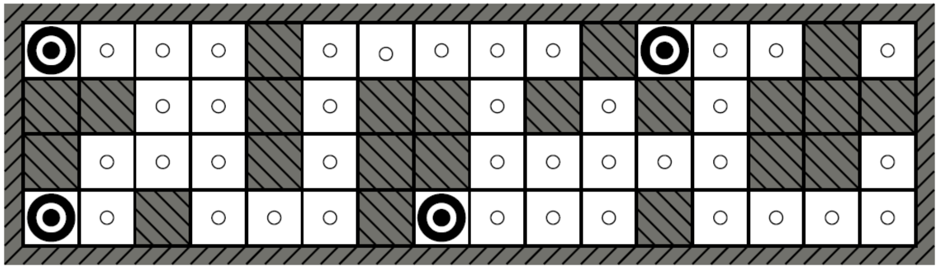
We now make the problem more realistic:

- ♣ Allow noise in sensing whether or not obstacles are immediately to the north, east, south, and west.
- ♣ The robot is equally likely to move to any adjacent square.

- State $X_t$: robot location
- $\{1, \ldots, S\}$: set of empty squares (labelled by integers)
- $\textsc{Neighbors}(i)$: set of empty squares that are adjacent to $i$
- $N(i)$: size of $\textsc{Neighbors}(i)$

# Transition Model

$$P(X_{t+1} = j \mid X_t = i) = \boldsymbol{T}_{ij} = \begin{cases} 1/N(i) & \text{if } j \in \textsc{Neighbors}(i) \\ 0 & \text{otherwise} \end{cases}$$

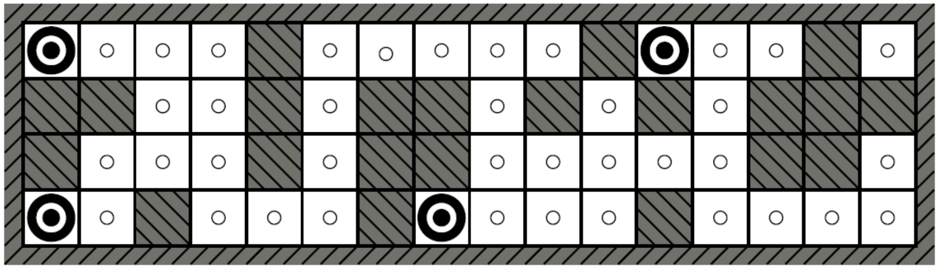


$S = 42$

Matrix $\boldsymbol{T} = (\boldsymbol{T}_{ij})$ has $42 \times 42 = 1764$ entries

# Transition Model

$$P(X_{t+1} = j \mid X_t = i) = \boldsymbol{T}_{ij} = \begin{cases} 1/N(i) & \text{if } j \in \text{Neighbors}(i) \\ 0 & \text{otherwise} \end{cases}$$



$S = 42$

Matrix $\boldsymbol{T} = (\boldsymbol{T}_{ij})$ has $42 \times 42 = 1764$ entries

Assume a uniform distribution of the robot's starting location:

$$P(X_0 = i) = 1/S \quad \text{for } 1 \le i \le S$$

# Sensor Model

Sensor variable $E_t = NESW$ has four bits and $2^4 = 16$ possible values.

# Sensor Model

Sensor variable $E_t = NESW$ has four bits and $2^4 = 16$ possible values.

Assumptions:

- Each sensor (N, E, S, or W) has an error rate $\varepsilon$.

# Sensor Model

Sensor variable $E_t = NESW$ has four bits and $2^4 = 16$ possible values.

Assumptions:

- Each sensor (N, E, S, or W) has an error rate $\varepsilon$.
- Errors occur independently for the four sensors.

# Sensor Model

Sensor variable $E_t = NESW$ has four bits and $2^4 = 16$ possible values.

Assumptions:

- Each sensor (N, E, S, or W) has an error rate $\varepsilon$.
- Errors occur independently for the four sensors.

  Probability of getting all four bits correct: $(1-\varepsilon)^4$

  Probability of getting all of them wrong: $\varepsilon^4$

# Sensor Model

Sensor variable $E_t = NESW$ has four bits and $2^4 = 16$ possible values.

Assumptions:

- Each sensor (N, E, S, or W) has an error rate $\varepsilon$.
- Errors occur independently for the four sensors.

  Probability of getting all four bits correct: $(1-\varepsilon)^4$

  Probability of getting all of them wrong: $\varepsilon^4$

$d_{it}$ = #bits that are different between the true values for square $i$ and actual reading $e_t$

# Sensor Model

Sensor variable $E_t = NESW$ has four bits and $2^4 = 16$ possible values.

Assumptions:

- Each sensor (N, E, S, or W) has an error rate $\varepsilon$.
- Errors occur independently for the four sensors.

Probability of getting all four bits correct: $(1-\varepsilon)^4$

Probability of getting all of them wrong: $\varepsilon^4$

$d_{it}$ = #bits that are different between the true values for square $i$ and actual reading $e_t$

Probability that the robot in square $i$ would receive a sensor reading $e_t$:

$$P(E_t = e_t \mid X_t = i) = (\boldsymbol{O}_t)_{ii} = (1-\varepsilon)^{4-d_{it}}\varepsilon^{d_{it}}$$

diagonal observation matrix

# Sensor Model

Sensor variable $E_t = NESW$ has four bits and $2^4 = 16$ possible values.

Assumptions:

- Each sensor (N, E, S, or W) has an error rate $\varepsilon$.
- Errors occur independently for the four sensors.

Probability of getting all four bits correct: $(1-\varepsilon)^4$

Probability of getting all of them wrong: $\varepsilon^4$

$d_{it}$ = #bits that are different between the true values for square $i$ and actual reading $e_t$

Probability that the robot in square $i$ would receive a sensor reading $e_t$:

$$P(E_t = e_t \mid X_t = i) = (\boldsymbol{O}_t)_{ii} = (1-\varepsilon)^{4-d_{it}} \varepsilon^{d_{it}}$$

diagonal observation matrix

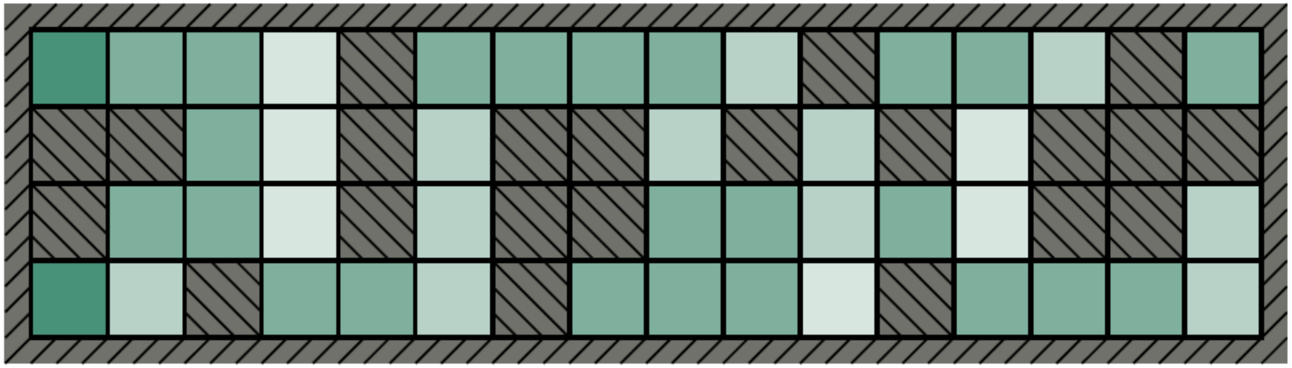
A square with obstacles to the north and south would produce a sensor reading of 1110 with probability $(1-\varepsilon)^3 \varepsilon^1$.

# Localization

Posterior distribution over locations using the filtering equation:

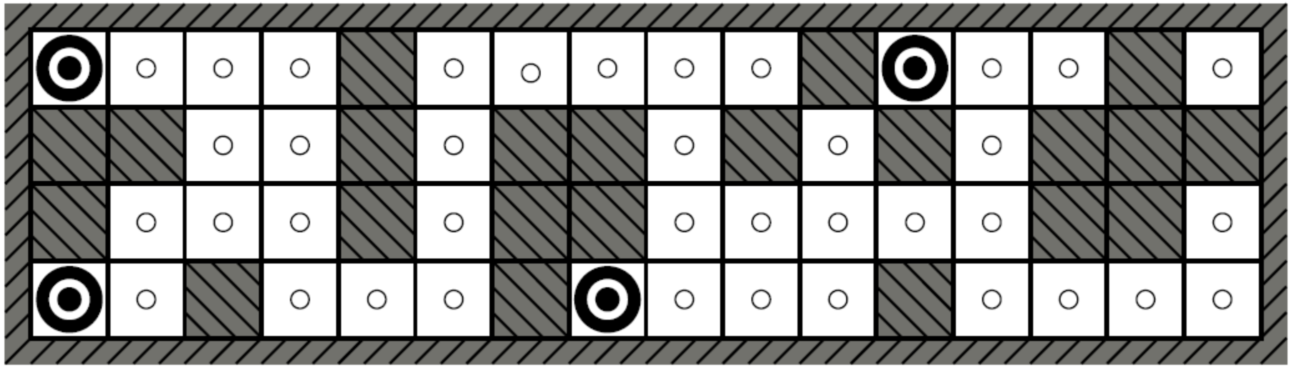
$$\underbrace{\boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t+1})}_{S \times 1 \text{ column vector } \boldsymbol{f}_{1:t+1}} = \alpha \boldsymbol{O}_{t+1} \boldsymbol{T}^T \boldsymbol{P}(\boldsymbol{X}_t \mid \boldsymbol{e}_{1:t})$$

$\varepsilon = 0.2$



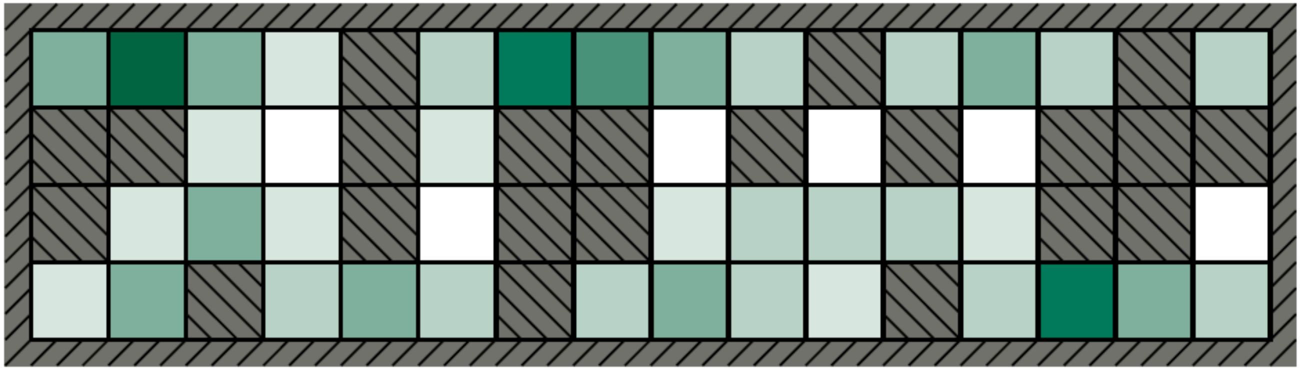
(a) Posterior distribution over robot location after $E_1 = 1011$

Perfect sensing



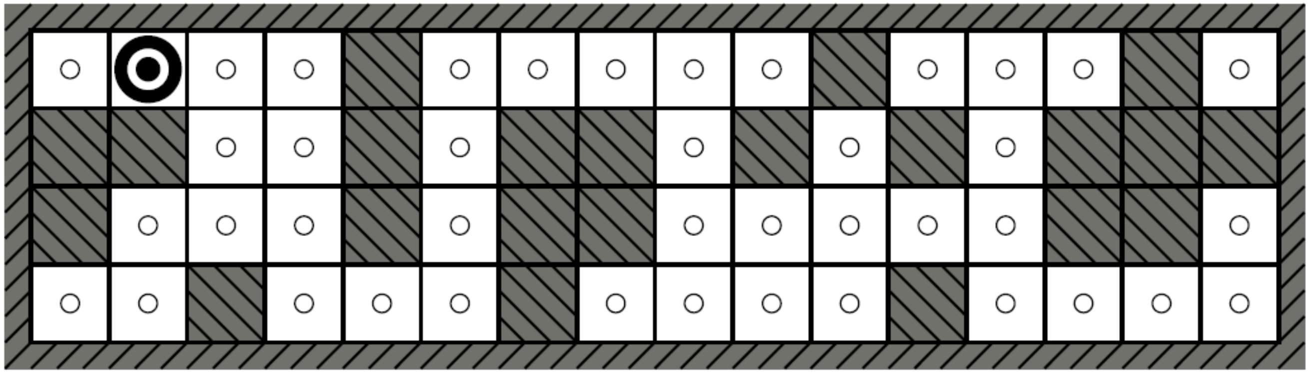
# Localization (cont'd)

$\varepsilon = 0.2$



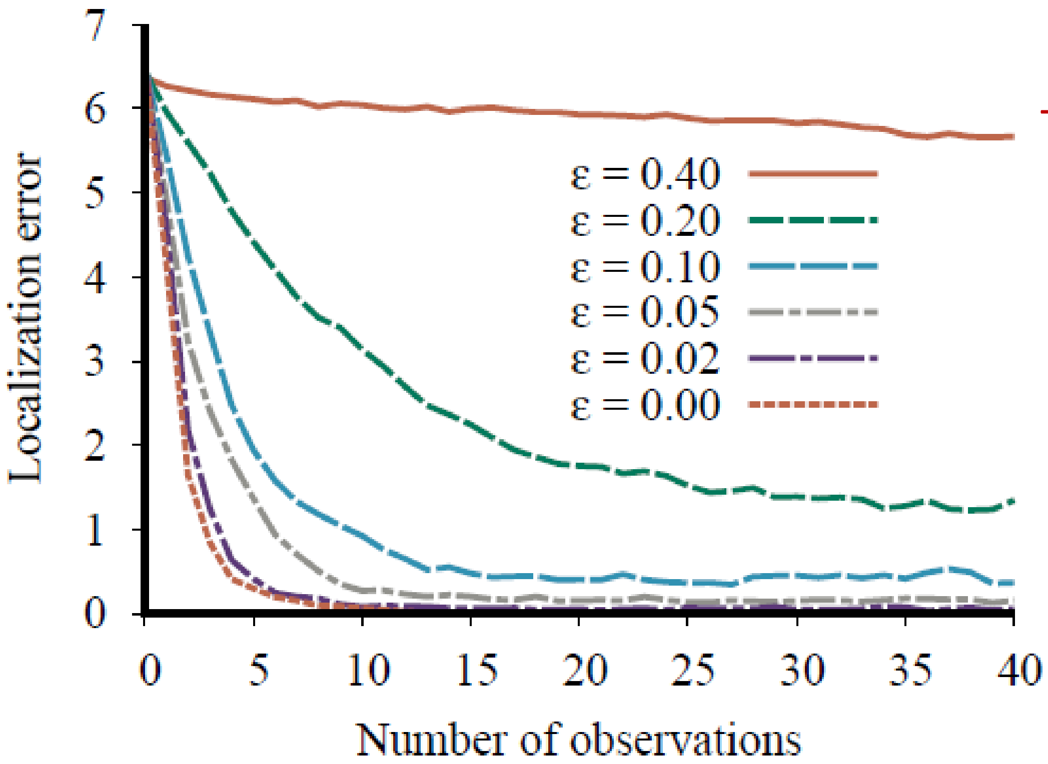
(b) Posterior distribution over robot location after $E_1 = 1011$, $E_2 = 1010$

Perfect sensing



# Localization Error

Measured as the Manhattan distance from the true location.




The robot is lost!

The robot can also use smoothing to work out where it was at a given past time.

# Viterbi Path Error

Error measured as the average Manhattan distance of states on the Viterbi path from corresponding states on the true path.