Lecture notes for *Principles of Artificial Intelligence* (Com S 4720/5720)
Yan-Bin Jia, Iowa State University

Inference in Temporal Models

Outline

I. States and observations

II. Transitions and sensor models

III. Filtering

* Figures/images are from the textbook site or the instructor unless the source is cited specifically.

I. Time and Uncertainty

- Probabilistic reasoning in *static worlds* discussed so far.

Every random variable
has a single fixed value.

- Real situations are *dynamic* with evidence evolving with time and thus actions predicted (and chosen) based on the history of evidence:
  - treating a patient
  - robot localization
  - tracking economic activities
  - speech recognition and natural language understanding
  - etc.

I. Time and Uncertainty

- Probabilistic reasoning in *static worlds* discussed so far.

Every random variable
has a single fixed value.

- Real situations are *dynamic* with evidence evolving with time and thus actions predicted (and chosen) based on the history of evidence:
  - treating a patient
  - robot localization
  - tracking economic activities
  - speech recognition and natural language understanding
  - etc.

How to model dynamic situations?

Discrete-Time Model

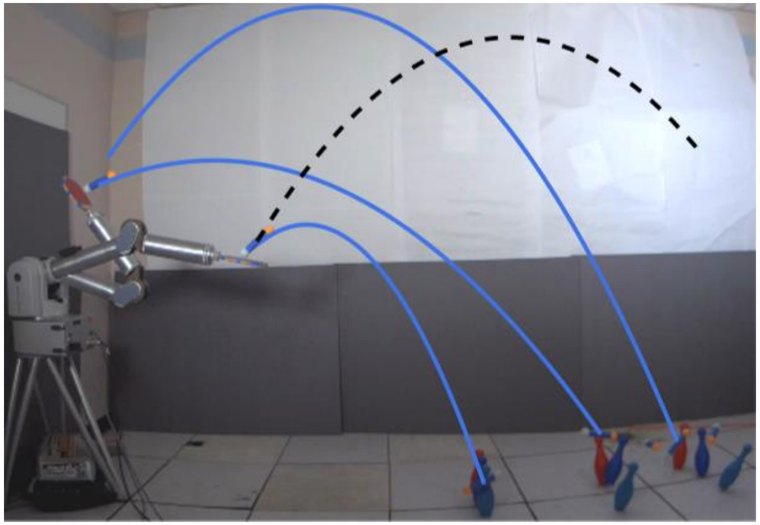
- The world is viewed as a series of time slices numbered 0, 1, 2, ... divided by the same interval Δ.

Discrete-Time Model

- The world is viewed as a series of time slices numbered 0, 1, 2, … divided by the same interval Δ.

Ximea MQ022CG-CM
high speed camera

Frame rate: 170 fps (frames per second)
Resolution: 2048×1088 pixel



https://www.youtube.com/watch?v=dGBevZ54E3s

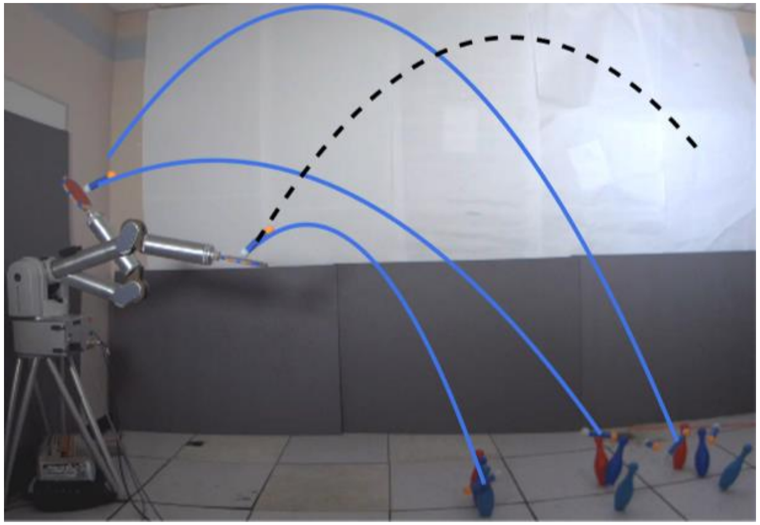
Batting an in-flight dumbbell-shaped object
Duration: 0.6 second with 90 frames
Motion of the object estimated using an extended Kalman filter (EKF).
IEEE Transactions on Robotics, vol. 38, no. 5, pp. 3187-3202, 2022.

Discrete-Time Model

- The world is viewed as a series of time slices numbered 0, 1, 2, … divided by the same interval Δ.

Ximea MQ022CG-CM
high speed camera

Frame rate: 170 fps (frames per second)
Resolution: 2048×1088 pixel



https://www.youtube.com/watch?v=dGBevZ54E3s

Batting an in-flight dumbbell-shaped object
Duration: 0.6 second with 90 frames
Motion of the object estimated using an extended Kalman filter (EKF).
IEEE Transactions on Robotics, vol. 38, no. 5, pp. 3187-3202, 2022.

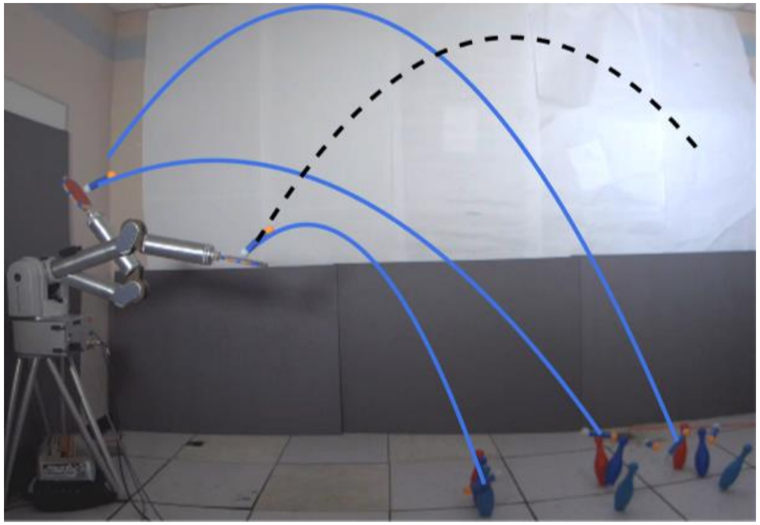
- Each time slice contains a set of random variables, observable or not.

Discrete-Time Model

- ♦ The world is viewed as a series of time slices numbered 0, 1, 2, … divided by the same interval Δ.

Ximea MQ022CG-CM
high speed camera

Frame rate: 170 fps (frames per second)
Resolution: 2048×1088 pixel



https://www.youtube.com/watch?v=dGBevZ54E3s
Batting an in-flight dumbbell-shaped object
Duration: 0.6 second with 90 frames
Motion of the object estimated using an extended Kalman filter (EKF).
IEEE Transactions on Robotics, vol. 38, no. 5, pp. 3187-3202, 2022.

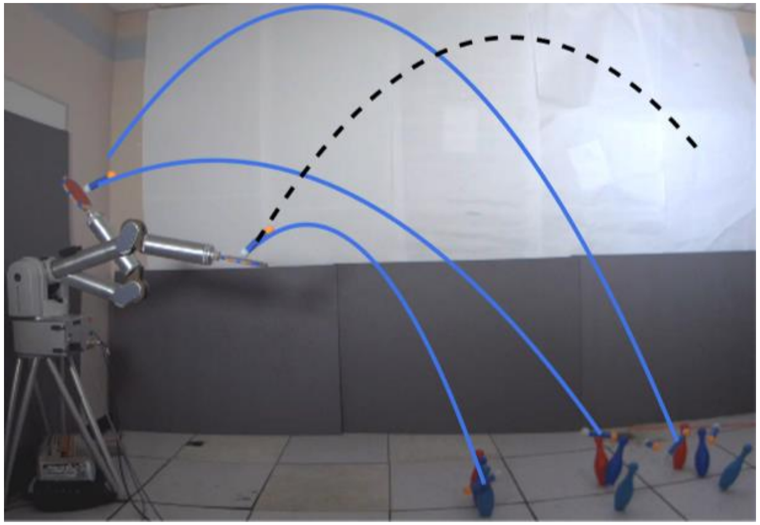
- ♦ Each time slice contains a set of random variables, observable or not.
  - $\boldsymbol{X}_t$: the set of *state variables* (assumed to be unobservable) at time t.
  - $\boldsymbol{E}_t$: the set of observable *evidence variables* at time t.

Discrete-Time Model

- The world is viewed as a series of time slices numbered 0, 1, 2, ... divided by the same interval Δ.

Ximea MQ022CG-CM
high speed camera

Frame rate: 170 fps (frames per second)
Resolution: 2048×1088 pixel



https://www.youtube.com/watch?v=dGBevZ54E3s

Batting an in-flight dumbbell-shaped object
Duration: 0.6 second with 90 frames
Motion of the object estimated using an extended Kalman filter (EKF).
IEEE Transactions on Robotics, vol. 38, no. 5, pp. 3187-3202, 2022.

- Each time slice contains a set of random variables, observable or not.
  - $\boldsymbol{X}_t$: the set of *state variables* (assumed to be unobservable) at time t.
  - $\boldsymbol{E}_t$: the set of observable *evidence variables* at time t.

$$\boldsymbol{E}_t = \boldsymbol{e}_t \text{ for some observed values } \boldsymbol{e}_t$$

The Umbrella World

  - $\boldsymbol{X}_{a:b}$ denotes the state sequence $\boldsymbol{X}_a, \boldsymbol{X}_{a+1}, \ldots, \boldsymbol{X}_b$
  - $\boldsymbol{E}_{a:b}$ denotes the evidence sequence $\boldsymbol{E}_a, \boldsymbol{E}_{a+1}, \ldots, \boldsymbol{E}_b$

- The state sequence starts at $t = 0$.
- Evidence starts at $t = 1$.

The Umbrella World

- $\boldsymbol{X}_{a:b}$ denotes the state sequence $\boldsymbol{X}_a, \boldsymbol{X}_{a+1}, \ldots, \boldsymbol{X}_b$
- $\boldsymbol{E}_{a:b}$ denotes the evidence sequence $\boldsymbol{E}_a, \boldsymbol{E}_{a+1}, \ldots, \boldsymbol{E}_b$

- The state sequence starts at $t = 0$.
- Evidence starts at $t = 1$.

Example A security guard stationed underground tries to infer whether it's raining today from seeing whether others come in with or without an umbrella.

The Umbrella World

  - $\boldsymbol{X}_{a:b}$ denotes the state sequence $\boldsymbol{X}_a, \boldsymbol{X}_{a+1}, \dots, \boldsymbol{X}_b$
  - $\boldsymbol{E}_{a:b}$ denotes the evidence sequence $\boldsymbol{E}_a, \boldsymbol{E}_{a+1}, \dots, \boldsymbol{E}_b$

- The state sequence starts at $t = 0$.
- Evidence starts at $t = 1$.

Example A security guard stationed underground tries to infer whether it’s raining today from seeing whether others come in with or without an umbrella.

For day t:

$$\boldsymbol{E}_t = \{\mathit{Umbrella}_t\} = \{U_t\}$$

$$\boldsymbol{X}_t = \{\mathit{Rain}_t\} = \{R_t\}$$

The Umbrella World

- $\boldsymbol{X}_{a:b}$ denotes the state sequence $\boldsymbol{X}_a, \boldsymbol{X}_{a+1}, \dots, \boldsymbol{X}_b$
- $\boldsymbol{E}_{a:b}$ denotes the evidence sequence $\boldsymbol{E}_a, \boldsymbol{E}_{a+1}, \dots, \boldsymbol{E}_b$

◆ The state sequence starts at $t = 0$.

◆ Evidence starts at $t = 1$.

Example A security guard stationed underground tries to infer whether it's raining today from seeing whether others come in with or without an umbrella.

For day t:

$$\boldsymbol{E}_t = \{\mathit{Umbrella}_t\} = \{U_t\}$$

$$\boldsymbol{X}_t = \{\mathit{Rain}_t\} = \{R_t\}$$

The umbrella world is represented by

state variable sequence: $R_0, R_1, R_2, \dots$

evidence variable sequence: $U_1, U_2, U_3, \dots$

$U_{3:5}$ corresponds to U_3, U_4, U_5.

II. Transition Model

Specifies the probability distribution over the latest state variables given the previous values:

$$\boldsymbol{P}(\boldsymbol{X}_t \mid \boldsymbol{X}_{0:t-1})$$

II. Transition Model

Specifies the probability distribution over the latest state variables given the previous values:

$$\boldsymbol{P}(\boldsymbol{X}_t \mid \boldsymbol{X}_{0:t-1})$$

Andrey Andreyevich Markov

Markov assumption The current state depends on only a finite fixed number of previous states.

* Photo from Wikipedia (https://en.wikipedia.org/wiki/Andrey_Markov).

II. Transition Model

Specifies the probability distribution over the latest state variables given the previous values:

Andrey Andreyevich Markov

$$\boldsymbol{P}(\boldsymbol{X}_t \mid \boldsymbol{X}_{0:t-1})$$

Markov assumption The current state depends on only a finite fixed number of previous states.

Markov chains are processes satisfying this assumption.

* Photo from Wikipedia (https://en.wikipedia.org/wiki/Andrey_Markov).

II. Transition Model

Specifies the probability distribution over the latest state variables given the previous values:

$$\boldsymbol{P}(\boldsymbol{X}_t \mid \boldsymbol{X}_{0:t-1})$$

Andrey Andreyevich Markov

Markov assumption The current state depends on only a finite fixed number of previous states.

Markov chains are processes satisfying this assumption.

kth order Markov chain (or *Markov process*):

$$\boldsymbol{P}(\boldsymbol{X}_t \mid \boldsymbol{X}_{0:t-1}) = \boldsymbol{P}(\boldsymbol{X}_t \mid \boldsymbol{X}_{t-k:t-1})$$

* Photo from Wikipedia (https://en.wikipedia.org/wiki/Andrey_Markov).

II. Transition Model

Specifies the probability distribution over the latest state variables given the previous values:

$$\boldsymbol{P}(\boldsymbol{X}_t \mid \boldsymbol{X}_{0:t-1})$$

Andrey Andreyevich Markov

Markov assumption The current state depends on only a finite fixed number of previous states.

Markov chains are processes satisfying this assumption.

kth order Markov chain (or *Markov process*):

$$\boldsymbol{P}(\boldsymbol{X}_t \mid \boldsymbol{X}_{0:t-1}) = \boldsymbol{P}(\boldsymbol{X}_t \mid \boldsymbol{X}_{t-k:t-1})$$

1st order Markov process
(transition model $\boldsymbol{P}(\boldsymbol{X}_t \mid \boldsymbol{X}_{t-1})$)

* Photo from Wikipedia (https://en.wikipedia.org/wiki/Andrey_Markov).

II. Transition Model

Specifies the probability distribution over the latest state variables given the previous values:

$$\boldsymbol{P}(\boldsymbol{X}_t \mid \boldsymbol{X}_{0:t-1})$$

Andrey Andreyevich Markov

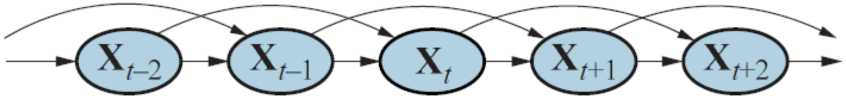
Markov assumption The current state depends on only a finite fixed number of previous states.

Markov chains are processes satisfying this assumption.

kth order Markov chain (or *Markov process*):

$$\boldsymbol{P}(\boldsymbol{X}_t \mid \boldsymbol{X}_{0:t-1}) = \boldsymbol{P}(\boldsymbol{X}_t \mid \boldsymbol{X}_{t-k:t-1})$$

1st order Markov process
(transition model $\boldsymbol{P}(\boldsymbol{X}_t \mid \boldsymbol{X}_{t-1})$)



2nd order Markov process
(transition model $\boldsymbol{P}(\boldsymbol{X}_t \mid \boldsymbol{X}_{t-1}, \boldsymbol{X}_{t-2})$)

* Photo from Wikipedia (https://en.wikipedia.org/wiki/Andrey_Markov).

Time Homogeneity & Sensor Assumptions

Time-Homogeneous Process Assumption

Changes in the world is governed by laws that do not change over time.

Time Homogeneity & Sensor Assumptions

Time-Homogeneous Process Assumption

Changes in the world is governed by laws that do not change over time.

E.g., in the umbrella world, $\boldsymbol{P}(R_t \mid R_{t-1})$ is the same for all t.

Time Homogeneity & Sensor Assumptions

Time-Homogeneous Process Assumption

Changes in the world is governed by laws that do not change over time.

E.g., in the umbrella world, $\boldsymbol{P}(R_t \mid R_{t-1})$ is the same for all t.

Sensor Markov Assumption

Current sensor values are generated by the current state only.

Time Homogeneity & Sensor Assumptions

Time-Homogeneous Process Assumption

Changes in the world is governed by laws that do not change over time.

E.g., in the umbrella world, $\mathbf{P}(R_t \mid R_{t-1})$ is the same for all t.

Sensor Markov Assumption

Current sensor values are generated by the current state only.

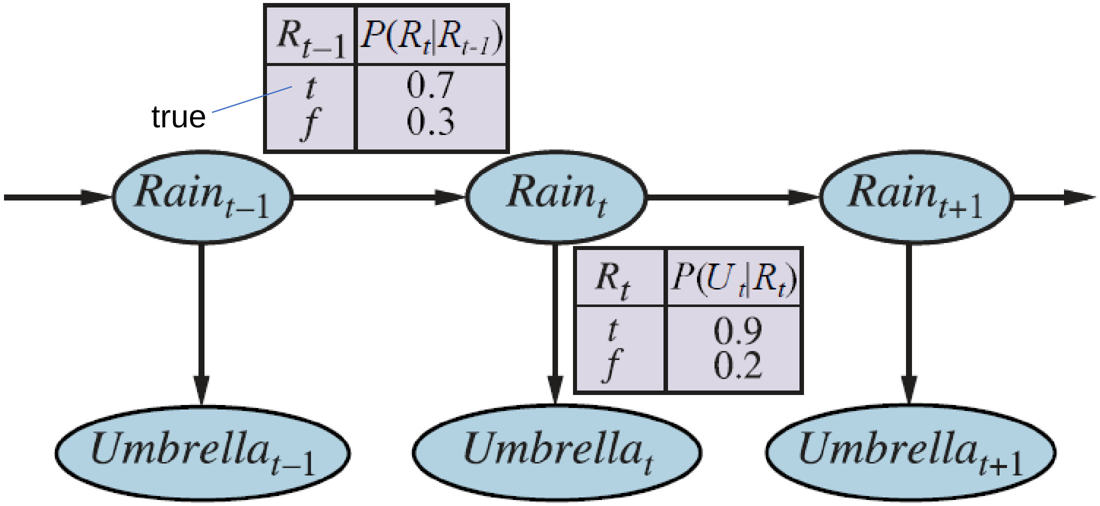
$$\mathbf{P}(\mathbf{E}_t \mid \mathbf{X}_{0:t}, \mathbf{E}_{1:t-1}) = \mathbf{P}(\mathbf{E}_t \mid \mathbf{X}_t)$$

sensor/observation
model

Bayes Net for the Umbrella World

time // t is overloaded

R_t = *true*



true

R_{t-1}	$P(R_t \| R_{t-1})$
t	0.7
f	0.3

R_t	$P(U_t \| R_t)$
t	0.9
f	0.2

Transition model:

$P(Rain_t \mid Rain_{t-1})$

Sensor model:

$P(Umbrella_t \mid Rain_t)$

1^{st} order Markov Process

The state (*Rain*) causes the sensor to take on a particular value (*Umbrella*).

Complete Joint Distribution

Given the prior distribution $\boldsymbol{P}(\boldsymbol{X}_0)$ at time 0, we have the complete joint distribution:

Complete Joint Distribution

Given the prior distribution $\boldsymbol{P}(\boldsymbol{X}_0)$ at time 0, we have the complete joint distribution:

$$\boldsymbol{P}(\boldsymbol{X}_{0:t}, \boldsymbol{E}_{1:t})$$

Complete Joint Distribution

Given the prior distribution $\boldsymbol{P}(\boldsymbol{X}_0)$ at time 0, we have the complete joint distribution:

$$\boldsymbol{P}(\boldsymbol{X}_{0:t}, \boldsymbol{E}_{1:t}) = \boldsymbol{P}(\boldsymbol{X}_0, \boldsymbol{X}_1, \boldsymbol{E}_1, \ldots, \boldsymbol{X}_t, \boldsymbol{E}_t)$$

Complete Joint Distribution

Given the prior distribution $\boldsymbol{P}(\boldsymbol{X}_0)$ at time 0, we have the complete joint distribution:

$$\begin{aligned}\boldsymbol{P}(\boldsymbol{X}_{0:t}, \boldsymbol{E}_{1:t}) &= \boldsymbol{P}(\boldsymbol{X}_0, \boldsymbol{X}_1, \boldsymbol{E}_1, \dots, \boldsymbol{X}_t, \boldsymbol{E}_t) \\ &= \boldsymbol{P}(\boldsymbol{X}_0)(\boldsymbol{P}(\boldsymbol{X}_1 | \boldsymbol{X}_0)\boldsymbol{P}(\boldsymbol{E}_1 \mid \boldsymbol{X}_1)) \cdots (\boldsymbol{P}(\boldsymbol{X}_t \mid \boldsymbol{X}_{t-1})\boldsymbol{P}(\boldsymbol{E}_t \mid \boldsymbol{X}_t))\end{aligned}$$

Complete Joint Distribution

Given the prior distribution $\boldsymbol{P}(\boldsymbol{X}_0)$ at time 0, we have the complete joint distribution:

$$\boldsymbol{P}(\boldsymbol{X}_{0:t}, \boldsymbol{E}_{1:t}) = \boldsymbol{P}(\boldsymbol{X}_0, \boldsymbol{X}_1, \boldsymbol{E}_1, \ldots, \boldsymbol{X}_t, \boldsymbol{E}_t)$$

$$= \boldsymbol{P}(\boldsymbol{X}_0)(\boldsymbol{P}(\boldsymbol{X}_1 | \boldsymbol{X}_0)\boldsymbol{P}(\boldsymbol{E}_1 | \boldsymbol{X}_1)) \cdots (\boldsymbol{P}(\boldsymbol{X}_t | \boldsymbol{X}_{t-1})\boldsymbol{P}(\boldsymbol{E}_t | \boldsymbol{X}_t))$$

$$= \boldsymbol{P}(\boldsymbol{X}_0) \prod_{\boldsymbol{i}=\boldsymbol{1}}^{\boldsymbol{t}} (\boldsymbol{P}(\boldsymbol{X}_i | \boldsymbol{X}_{i-1})\boldsymbol{P}(\boldsymbol{E}_i | \boldsymbol{X}_i))$$

Complete Joint Distribution

Given the prior distribution $\boldsymbol{P}(\boldsymbol{X}_0)$ at time 0, we have the complete joint distribution:

$$\boldsymbol{P}(\boldsymbol{X}_{0:t}, \boldsymbol{E}_{1:t}) = \boldsymbol{P}(\boldsymbol{X}_0, \boldsymbol{X}_1, \boldsymbol{E}_1, \ldots, \boldsymbol{X}_t, \boldsymbol{E}_t)$$

$$= \boldsymbol{P}(\boldsymbol{X}_0)(\boldsymbol{P}(\boldsymbol{X}_1 | \boldsymbol{X}_0)\boldsymbol{P}(\boldsymbol{E}_1 | \boldsymbol{X}_1)) \cdots (\boldsymbol{P}(\boldsymbol{X}_t | \boldsymbol{X}_{t-1})\boldsymbol{P}(\boldsymbol{E}_t | \boldsymbol{X}_t))$$

$$= \underbrace{\boldsymbol{P}(\boldsymbol{X}_0)}_{\text{initial state model}} \prod_{\boldsymbol{i}=\boldsymbol{1}}^{\boldsymbol{t}} (\boldsymbol{P}(\boldsymbol{X}_i | \boldsymbol{X}_{i-1})\boldsymbol{P}(\boldsymbol{E}_i | \boldsymbol{X}_i))$$

Complete Joint Distribution

Given the prior distribution $\boldsymbol{P}(\boldsymbol{X}_0)$ at time 0, we have the complete joint distribution:

$$\boldsymbol{P}(\boldsymbol{X}_{0:t}, \boldsymbol{E}_{1:t}) = \boldsymbol{P}(\boldsymbol{X}_0, \boldsymbol{X}_1, \boldsymbol{E}_1, \ldots, \boldsymbol{X}_t, \boldsymbol{E}_t)$$

$$= \boldsymbol{P}(\boldsymbol{X}_0)(\boldsymbol{P}(\boldsymbol{X}_1 | \boldsymbol{X}_0)\boldsymbol{P}(\boldsymbol{E}_1 | \boldsymbol{X}_1)) \cdots (\boldsymbol{P}(\boldsymbol{X}_t | \boldsymbol{X}_{t-1})\boldsymbol{P}(\boldsymbol{E}_t | \boldsymbol{X}_t))$$

$$= \underbrace{\boldsymbol{P}(\boldsymbol{X}_0)}_{\text{initial state model}} \prod_{\boldsymbol{i}=\boldsymbol{1}}^{\boldsymbol{t}} (\underbrace{\boldsymbol{P}(\boldsymbol{X}_i | \boldsymbol{X}_{i-1})}_{\text{transition model}}\boldsymbol{P}(\boldsymbol{E}_i | \boldsymbol{X}_i))$$

Complete Joint Distribution

Given the prior distribution $\boldsymbol{P}(\boldsymbol{X}_0)$ at time 0, we have the complete joint distribution:

$$
\begin{aligned}
\boldsymbol{P}(\boldsymbol{X}_{0:t}, \boldsymbol{E}_{1:t}) &= \boldsymbol{P}(\boldsymbol{X}_0, \boldsymbol{X}_1, \boldsymbol{E}_1, \ldots, \boldsymbol{X}_t, \boldsymbol{E}_t) \\
&= \boldsymbol{P}(\boldsymbol{X}_0)(\boldsymbol{P}(\boldsymbol{X}_1 \mid \boldsymbol{X}_0)\boldsymbol{P}(\boldsymbol{E}_1 \mid \boldsymbol{X}_1)) \cdots (\boldsymbol{P}(\boldsymbol{X}_t \mid \boldsymbol{X}_{t-1})\boldsymbol{P}(\boldsymbol{E}_t \mid \boldsymbol{X}_t)) \\
&= \underbrace{\boldsymbol{P}(\boldsymbol{X}_0)}_{\text{initial state model}} \prod_{\boldsymbol{i}=\boldsymbol{1}}^{\boldsymbol{t}} (\underbrace{\boldsymbol{P}(\boldsymbol{X}_i \mid \boldsymbol{X}_{i-1})}_{\text{transition model}}\underbrace{\boldsymbol{P}(\boldsymbol{E}_i \mid \boldsymbol{X}_i)}_{\text{sensor model}})
\end{aligned}
$$

Complete Joint Distribution

Given the prior distribution $\boldsymbol{P}(\boldsymbol{X}_0)$ at time 0, we have the complete joint distribution:

$$\boldsymbol{P}(\boldsymbol{X}_{0:t},\boldsymbol{E}_{1:t}) = \boldsymbol{P}(\boldsymbol{X}_0,\boldsymbol{X}_1,\boldsymbol{E}_1,\ldots,\boldsymbol{X}_t,\boldsymbol{E}_t)$$

$$= \boldsymbol{P}(\boldsymbol{X}_0)(\boldsymbol{P}(\boldsymbol{X}_1 \mid \boldsymbol{X}_0)\boldsymbol{P}(\boldsymbol{E}_1 \mid \boldsymbol{X}_1))\cdots(\boldsymbol{P}(\boldsymbol{X}_t \mid \boldsymbol{X}_{t-1})\boldsymbol{P}(\boldsymbol{E}_t \mid \boldsymbol{X}_t))$$

$$= \underbrace{\boldsymbol{P}(\boldsymbol{X}_0)}_{\text{initial state model}} \prod_{\boldsymbol{i=1}}^{\boldsymbol{t}} (\underbrace{\boldsymbol{P}(\boldsymbol{X}_i \mid \boldsymbol{X}_{i-1})}_{\text{transition model}}\underbrace{\boldsymbol{P}(\boldsymbol{E}_i \mid \boldsymbol{X}_i)}_{\text{sensor model}})$$

♠ Such a model cannot be represented by a standard Bayes net which requires a finite set of variables.

Complete Joint Distribution

Given the prior distribution $\boldsymbol{P}(\boldsymbol{X}_0)$ at time 0, we have the complete joint distribution:

$$\boldsymbol{P}(\boldsymbol{X}_{0:t}, \boldsymbol{E}_{1:t}) = \boldsymbol{P}(\boldsymbol{X}_0, \boldsymbol{X}_1, \boldsymbol{E}_1, \ldots, \boldsymbol{X}_t, \boldsymbol{E}_t)$$

$$= \boldsymbol{P}(\boldsymbol{X}_0)(\boldsymbol{P}(\boldsymbol{X}_1 | \boldsymbol{X}_0)\boldsymbol{P}(\boldsymbol{E}_1 | \boldsymbol{X}_1)) \cdots (\boldsymbol{P}(\boldsymbol{X}_t | \boldsymbol{X}_{t-1})\boldsymbol{P}(\boldsymbol{E}_t | \boldsymbol{X}_t))$$

$$= \underbrace{\boldsymbol{P}(\boldsymbol{X}_0)}_{\text{initial state model}} \prod_{\boldsymbol{i}=\boldsymbol{1}}^{\boldsymbol{t}} (\underbrace{\boldsymbol{P}(\boldsymbol{X}_i | \boldsymbol{X}_{i-1})}_{\text{transition model}} \underbrace{\boldsymbol{P}(\boldsymbol{E}_i | \boldsymbol{X}_i)}_{\text{sensor model}})$$

♠ Such a model cannot be represented by a standard Bayes net which requires a finite set of variables.

- ♦ Discrete time models can handle an infinite set of variables due to
  - ♣ use of integer indices
  - ♣ use of implicit universal quantification to define sensor and transition models

III. Filtering & Prediction

Belief state: $\boldsymbol{P}(\boldsymbol{X}_t \mid \boldsymbol{e}_{1:t})$, the posterior distribution over the most recent state give all evidence to date.

III. Filtering & Prediction

Belief state: $\boldsymbol{P}(\boldsymbol{X}_t \mid \boldsymbol{e}_{1:t})$, the posterior distribution over the most recent state give all evidence to date.

◆ *Filtering* (or *state estimation*) is the task of computing $\boldsymbol{P}(\boldsymbol{X}_t \mid \boldsymbol{e}_{1:t})$.

Keep track of the current state as time evolves (i.e., as t increases).

III. Filtering & Prediction

Belief state: $\boldsymbol{P}(\boldsymbol{X}_t \mid \boldsymbol{e}_{1:t})$, the posterior distribution over the most recent state give all evidence to date.

- *Filtering* (or *state estimation*) is the task of computing $\boldsymbol{P}(\boldsymbol{X}_t \mid \boldsymbol{e}_{1:t})$.

  Keep track of the current state as time evolves (i.e., as t increases).

  E.g., the probability of rain today, given all the umbrella observations made so far.

III. Filtering & Prediction

Belief state: $\boldsymbol{P}(\boldsymbol{X}_t \mid \boldsymbol{e}_{1:t})$, the posterior distribution over the most recent state give all evidence to date.

- *Filtering* (or *state estimation*) is the task of computing $\boldsymbol{P}(\boldsymbol{X}_t \mid \boldsymbol{e}_{1:t})$.

  Keep track of the current state as time evolves (i.e., as t increases).

  E.g., the probability of rain today, given all the umbrella observations made so far.

- *Prediction* is the task of computing $\boldsymbol{P}(\boldsymbol{X}_{t+k} \mid \boldsymbol{e}_{1:t})$ for some $k > 0$.

  Compute the posterior distribution over a future state, given all the evidence to date.

III. Filtering & Prediction

Belief state: $\boldsymbol{P}(\boldsymbol{X}_t \mid \boldsymbol{e}_{1:t})$, the posterior distribution over the most recent state give all evidence to date.

- *Filtering* (or *state estimation*) is the task of computing $\boldsymbol{P}(\boldsymbol{X}_t \mid \boldsymbol{e}_{1:t})$.

  Keep track of the current state as time evolves (i.e., as t increases).

  E.g., the probability of rain today, given all the umbrella observations made so far.

- *Prediction* is the task of computing $\boldsymbol{P}(\boldsymbol{X}_{t+k} \mid \boldsymbol{e}_{1:t})$ for some $k > 0$.

  Compute the posterior distribution over a future state, given all the evidence to date.

  E.g., the probability of rain three days from now.

Smoothing & Most Likely Explanation

- *Smoothing* is the task of computing $\mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{1:t})$ for some k, $0 \le k < t$.

Smoothing & Most Likely Explanation

- *Smoothing* is the task of computing $\boldsymbol{P}(\boldsymbol{X}_k \mid \boldsymbol{e}_{1:t})$ for some k, $0 \leq k < t$.
  - Compute the posterior distribution over a past state, given all the evidence to date.

Smoothing & Most Likely Explanation

- *Smoothing* is the task of computing $\boldsymbol{P}(\boldsymbol{X}_k \mid \boldsymbol{e}_{1:t})$ for some k, $0 \leq k < t$.

  Compute the posterior distribution over a past state, given all the evidence to date.

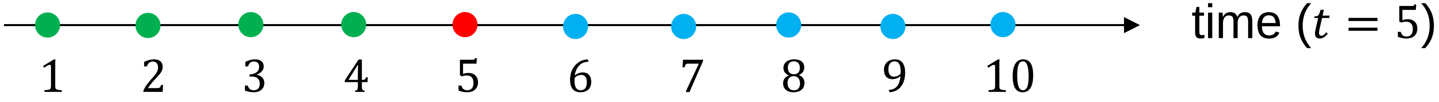
  E.g., the probability that it rained last Wednesday.

Smoothing & Most Likely Explanation

- *Smoothing* is the task of computing $\boldsymbol{P}(\boldsymbol{X}_k \mid \boldsymbol{e}_{1:t})$ for some k, $0 \le k < t$.

  Compute the posterior distribution over a past state, given all the evidence to date.

  E.g., the probability that it rained last Wednesday.

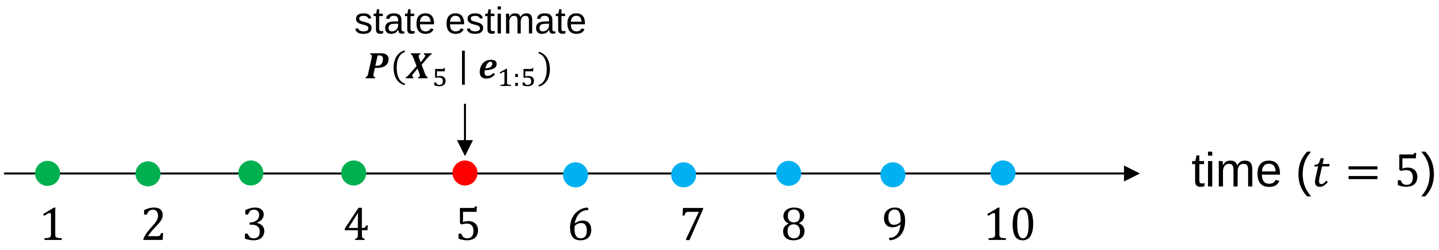


Smoothing & Most Likely Explanation

- *Smoothing* is the task of computing $\boldsymbol{P}(\boldsymbol{X}_k \mid \boldsymbol{e}_{1:t})$ for some k, $0 \le k < t$.

  Compute the posterior distribution over a past state, given all the evidence to date.

  E.g., the probability that it rained last Wednesday.



state estimate

$\boldsymbol{P}(\boldsymbol{X}_5 \mid \boldsymbol{e}_{1:5})$

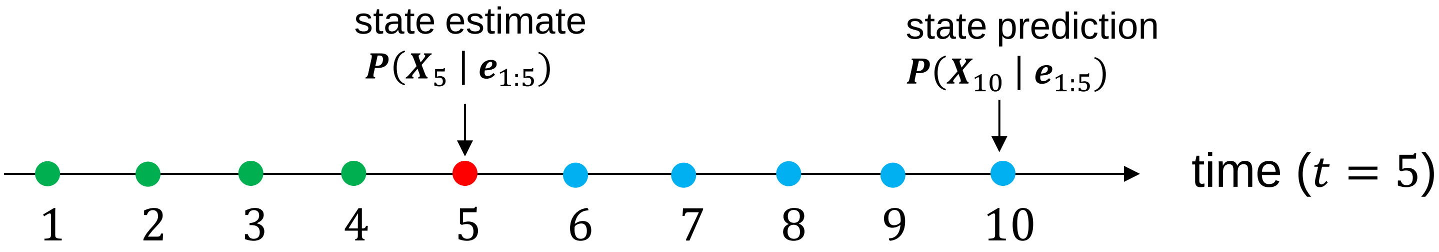
time ($t = 5$)

Smoothing & Most Likely Explanation

- *Smoothing* is the task of computing $\boldsymbol{P}(\boldsymbol{X}_k \mid \boldsymbol{e}_{1:t})$ for some k, $0 \le k < t$.

  Compute the posterior distribution over a past state, given all the evidence to date.

  E.g., the probability that it rained last Wednesday.



state estimate

$\boldsymbol{P}(\boldsymbol{X}_5 \mid \boldsymbol{e}_{1:5})$

state prediction

$\boldsymbol{P}(\boldsymbol{X}_{10} \mid \boldsymbol{e}_{1:5})$

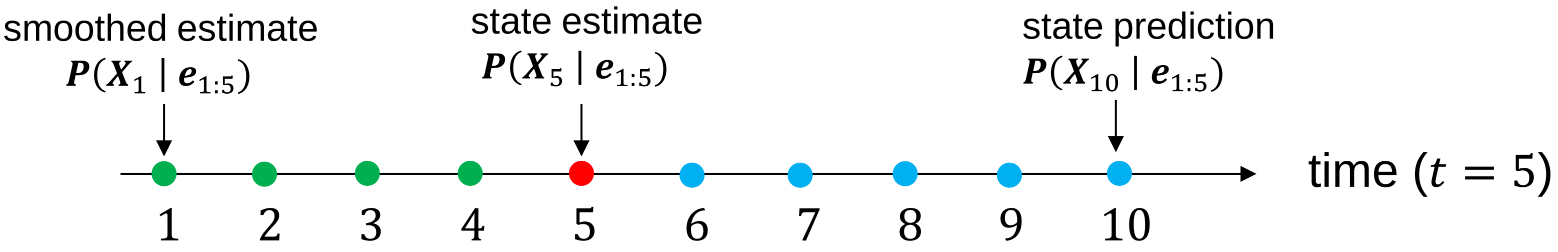
time ($t = 5$)

Smoothing & Most Likely Explanation

- *Smoothing* is the task of computing $\boldsymbol{P}(\boldsymbol{X}_k \mid \boldsymbol{e}_{1:t})$ for some k, $0 \le k < t$.

  Compute the posterior distribution over a past state, given all the evidence to date.

  E.g., the probability that it rained last Wednesday.

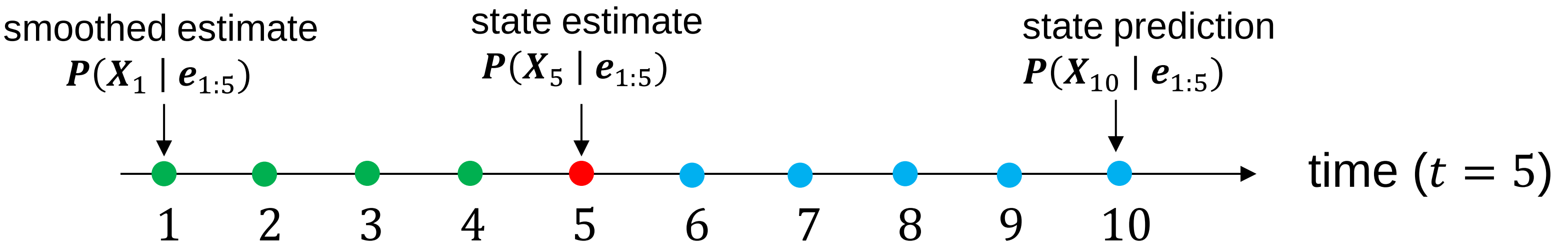


Smoothing & Most Likely Explanation

- *Smoothing* is the task of computing $\boldsymbol{P}(\boldsymbol{X}_k \mid \boldsymbol{e}_{1:t})$ for some k, $0 \leq k < t$.

  Compute the posterior distribution over a past state, given all the evidence to date.

  E.g., the probability that it rained last Wednesday.



- *Most likely explanation* is the task of computing the sequence of states $\boldsymbol{x}_{1:t}$ to maximize $\boldsymbol{P}(\boldsymbol{X}_{1:t} \mid \boldsymbol{e}_{1:t})$.

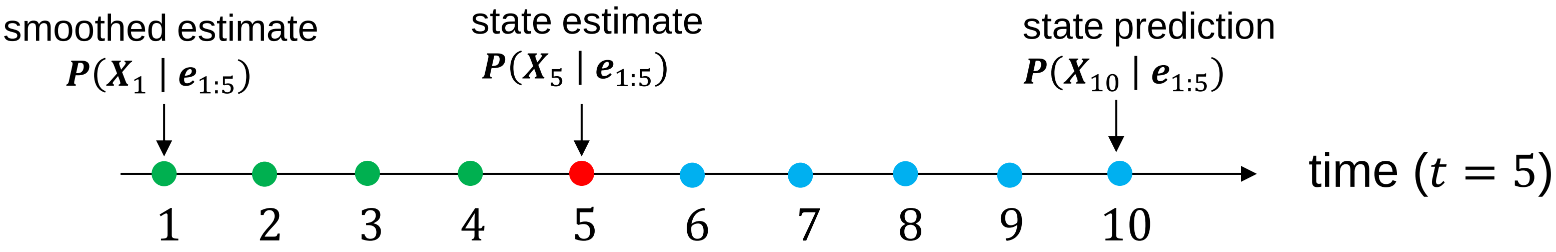
  Given a sequence of observations, find the sequence of states that is mostly likely to have generated those observations.

Smoothing & Most Likely Explanation

- *Smoothing* is the task of computing $\mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{1:t})$ for some k, $0 \leq k < t$.

  Compute the posterior distribution over a past state, given all the evidence to date.

  E.g., the probability that it rained last Wednesday.



smoothed estimate $\mathbf{P}(\mathbf{X}_1 \mid \mathbf{e}_{1:5})$

state estimate $\mathbf{P}(\mathbf{X}_5 \mid \mathbf{e}_{1:5})$

state prediction $\mathbf{P}(\mathbf{X}_{10} \mid \mathbf{e}_{1:5})$

1 2 3 4 5 6 7 8 9 10

time ($t = 5$)

- *Most likely explanation* is the task of computing the sequence of states $\mathbf{x}_{1:t}$ to maximize $\mathbf{P}(\mathbf{X}_{1:t} \mid \mathbf{e}_{1:t})$.

  Given a sequence of observations, find the sequence of states that is mostly likely to have generated those observations.

  E.g., If the umbrella appears on each of the first three days and is absent on the fourth, then the most likely explanation is that it rained on the first three days and did not rain on the fourth.

Filtering

To be efficient (so usable in a real time scenario), a filtering algorithm

♦ needs to maintain current state estimate and update it on the fly, and

♠ should not go back over the entire history

Filtering

To be efficient (so usable in a real time scenario), a filtering algorithm

- ♦ needs to maintain current state estimate and update it on the fly, and
- ♠ should not go back over the entire history

Recursive estimation:

$$\underbrace{\boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t+1})}_{\text{state estimate at } t+1} = f(\boldsymbol{e}_{t+1}, \underbrace{\boldsymbol{P}(\boldsymbol{X}_t \mid \boldsymbol{e}_{1:t})}_{\text{state estimate at } t})$$

Filtering

To be efficient (so usable in a real time scenario), a filtering algorithm

- ♦ needs to maintain current state estimate and update it on the fly, and
- ♠ should not go back over the entire history

Recursive estimation:

$$\underbrace{\boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t+1})}_{\text{state estimate at } t+1} = f(\boldsymbol{e}_{t+1}, \underbrace{\boldsymbol{P}(\boldsymbol{X}_t \mid \boldsymbol{e}_{1:t})}_{\text{state estimate at } t})$$

a) Projects the current state distribution forward from t to $t+1$.

b) Updates the projected estimate using the new evidence $\boldsymbol{e}_{t+1}$.

Filtering

To be efficient (so usable in a real time scenario), a filtering algorithm

- ♦ needs to maintain current state estimate and update it on the fly, and
- ♠ should not go back over the entire history

Recursive estimation:

$$\underbrace{\boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t+1})}_{\text{state estimate at } t+1} = f(\boldsymbol{e}_{t+1}, \underbrace{\boldsymbol{P}(\boldsymbol{X}_t \mid \boldsymbol{e}_{1:t})}_{\text{state estimate at } t})$$

a) Projects the current state distribution forward from t to $t+1$.

b) Updates the projected estimate using the new evidence $\boldsymbol{e}_{t+1}$.

$$\boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t+1}) = \boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t}, \boldsymbol{e}_{t+1}) \qquad \text{(Dividing up the evidence)}$$

Filtering

To be efficient (so usable in a real time scenario), a filtering algorithm

- ♦ needs to maintain current state estimate and update it on the fly, and
- ♠ should not go back over the entire history

Recursive estimation:

$$\boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t+1}) = f(\boldsymbol{e}_{t+1}, \boldsymbol{P}(\boldsymbol{X}_t \mid \boldsymbol{e}_{1:t}))$$

state estimate at $t+1$ state estimate at t

a) Projects the current state distribution forward from t to $t+1$.

b) Updates the projected estimate using the new evidence $\boldsymbol{e}_{t+1}$.

$$\boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t+1}) = \boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t}, \boldsymbol{e}_{t+1}) \quad \text{(Dividing up the evidence)}$$

$$= \alpha \boldsymbol{P}(\boldsymbol{e}_{t+1} \mid \boldsymbol{X}_{t+1}, \boldsymbol{e}_{1:t})\ \boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t}) \quad \text{(Bayes' rule, given } \boldsymbol{e}_{1:t}\text{)}$$

Filtering

To be efficient (so usable in a real time scenario), a filtering algorithm

- ♦ needs to maintain current state estimate and update it on the fly, and
- ♠ should not go back over the entire history

Recursive estimation:

$$\underbrace{\boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t+1})}_{\text{state estimate at } t+1} = f(\boldsymbol{e}_{t+1}, \underbrace{\boldsymbol{P}(\boldsymbol{X}_t \mid \boldsymbol{e}_{1:t})}_{\text{state estimate at } t})$$

a) Projects the current state distribution forward from t to $t+1$.

b) Updates the projected estimate using the new evidence $\boldsymbol{e}_{t+1}$.

$$\boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t+1}) = \boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t}, \boldsymbol{e}_{t+1}) \qquad \text{(Dividing up the evidence)}$$

$$= \underbrace{\alpha}_{\text{normalizing constant}} \boldsymbol{P}(\boldsymbol{e}_{t+1} \mid \boldsymbol{X}_{t+1}, \boldsymbol{e}_{1:t})\ \boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t}) \qquad \text{(Bayes' rule, given } \boldsymbol{e}_{1:t}\text{)}$$

Filtering

To be efficient (so usable in a real time scenario), a filtering algorithm

♦ needs to maintain current state estimate and update it on the fly, and

♠ should not go back over the entire history

Recursive estimation:

$$\underbrace{\boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t+1})}_{\text{state estimate at } t+1} = f(\boldsymbol{e}_{t+1}, \underbrace{\boldsymbol{P}(\boldsymbol{X}_t \mid \boldsymbol{e}_{1:t})}_{\text{state estimate at } t})$$

a) Projects the current state distribution forward from t to $t+1$.

b) Updates the projected estimate using the new evidence $\boldsymbol{e}_{t+1}$.

$$\boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t+1}) = \boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t}, \boldsymbol{e}_{t+1}) \quad \text{(Dividing up the evidence)}$$

$$= \alpha \boldsymbol{P}(\boldsymbol{e}_{t+1} \mid \boldsymbol{X}_{t+1}, \boldsymbol{e}_{1:t})\, \boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t}) \quad \text{(Bayes' rule, given } \boldsymbol{e}_{1:t}\text{)}$$

normalizing constant

$$= \alpha \boldsymbol{P}(\boldsymbol{e}_{t+1} \mid \boldsymbol{X}_{t+1})\, \boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t}) \quad \text{(by the sensor Markov assumption)}$$

Filtering

To be efficient (so usable in a real time scenario), a filtering algorithm

♦ needs to maintain current state estimate and update it on the fly, and

♠ should not go back over the entire history

Recursive estimation:

$$\boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t+1}) = f(\boldsymbol{e}_{t+1}, \boldsymbol{P}(\boldsymbol{X}_t \mid \boldsymbol{e}_{1:t}))$$

state estimate at $t+1$ state estimate at t

a) Projects the current state distribution forward from t to $t+1$.

b) Updates the projected estimate using the new evidence $\boldsymbol{e}_{t+1}$.

$$\boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t+1}) = \boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t}, \boldsymbol{e}_{t+1}) \qquad \text{(Dividing up the evidence)}$$

$$= \alpha \boldsymbol{P}(\boldsymbol{e}_{t+1} \mid \boldsymbol{X}_{t+1}, \boldsymbol{e}_{1:t})\ \boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t}) \qquad \text{(Bayes' rule, given } \boldsymbol{e}_{1:t}\text{)}$$

normalizing constant

$$= \alpha \underbrace{\boldsymbol{P}(\boldsymbol{e}_{t+1} \mid \boldsymbol{X}_{t+1})}_{\text{update}}\ \underbrace{\boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t})}_{\text{prediction}} \qquad \text{(by the sensor Markov assumption)}$$

One-Step Prediction

$$\boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t+1}) = \alpha \boldsymbol{P}(\boldsymbol{e}_{t+1} \mid \boldsymbol{X}_{t+1})\ \boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t})$$

One-Step Prediction

$$\boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t+1}) = \alpha \boldsymbol{P}(\boldsymbol{e}_{t+1} \mid \boldsymbol{X}_{t+1})\ \boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t})$$

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

One-Step Prediction

$$\boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t+1}) = \alpha \boldsymbol{P}(\boldsymbol{e}_{t+1} \mid \boldsymbol{X}_{t+1})\ \boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t})$$

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

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

One-Step Prediction

$$\mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t+1}) = \alpha \mathbf{P}(\mathbf{e}_{t+1} \mid \mathbf{X}_{t+1})\ \mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t})$$

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

$$= \underbrace{\alpha \mathbf{P}(\mathbf{e}_{t+1} \mid \mathbf{X}_{t+1})}_{\text{sensor model}} \sum_{\mathbf{x}_t} \mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{x}_t) P(\mathbf{x}_t \mid \mathbf{e}_{1:t})$$

One-Step Prediction

$$\boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t+1}) = \alpha \boldsymbol{P}(\boldsymbol{e}_{t+1} \mid \boldsymbol{X}_{t+1})\ \boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t})$$

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

$$= \alpha \underbrace{\boldsymbol{P}(\boldsymbol{e}_{t+1} \mid \boldsymbol{X}_{t+1})}_{\text{sensor model}} \sum_{\boldsymbol{x}_t} \underbrace{\boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{x}_t)}_{\text{transition model}} P(\boldsymbol{x}_t \mid \boldsymbol{e}_{1:t})$$

One-Step Prediction

$$\boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t+1}) = \alpha \boldsymbol{P}(\boldsymbol{e}_{t+1} \mid \boldsymbol{X}_{t+1})\ \boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t})$$

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

$$= \alpha \underbrace{\boldsymbol{P}(\boldsymbol{e}_{t+1} \mid \boldsymbol{X}_{t+1})}_{\text{sensor model}} \sum_{\boldsymbol{x}_t} \underbrace{\boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{x}_t)}_{\text{transition model}} \underbrace{P(\boldsymbol{x}_t \mid \boldsymbol{e}_{1:t})}_{\text{recursion}}$$

One-Step Prediction

$$\mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t+1}) = \alpha \mathbf{P}(\mathbf{e}_{t+1} \mid \mathbf{X}_{t+1})\ \mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t})$$

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

$$= \alpha \underbrace{\mathbf{P}(\mathbf{e}_{t+1} \mid \mathbf{X}_{t+1})}_{\text{sensor model}} \boxed{\sum_{\mathbf{x}_t} \underbrace{\mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{x}_t)}_{\text{transition model}} \underbrace{P(\mathbf{x}_t \mid \mathbf{e}_{1:t})}_{\text{recursion}}}\ \text{prediction}$$

One-Step Prediction

$$\mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t+1}) = \alpha \mathbf{P}(\mathbf{e}_{t+1} \mid \mathbf{X}_{t+1})\ \mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t})$$

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

update

$$= \alpha \underbrace{\mathbf{P}(\mathbf{e}_{t+1} \mid \mathbf{X}_{t+1})}_{\text{sensor model}} \sum_{\mathbf{x}_t} \underbrace{\mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{x}_t)}_{\text{transition model}} \underbrace{P(\mathbf{x}_t \mid \mathbf{e}_{1:t})}_{\text{recursion}}$$

prediction

One-Step Prediction

$$\mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t+1}) = \alpha \mathbf{P}(\mathbf{e}_{t+1} \mid \mathbf{X}_{t+1})\ \mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t})$$

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

$$= \alpha \overset{\text{update}}{\underbrace{\boxed{\mathbf{P}(\mathbf{e}_{t+1} \mid \mathbf{X}_{t+1})}}_{\text{sensor model}}} \boxed{\sum_{\mathbf{x}_t} \underbrace{\mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{x}_t)}_{\text{transition model}} \underbrace{P(\mathbf{x}_t \mid \mathbf{e}_{1:t})}_{\text{recursion}}} \ \text{prediction}$$

$$\begin{cases} \mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t+1}) = \text{FORWARD}(\underbrace{\mathbf{P}(\mathbf{X}_t \mid \mathbf{e}_{1:t})}_{\text{“forward” message}}, \mathbf{e}_{t+1}) \\ \mathbf{P}(\mathbf{X}_0 \mid \mathbf{e}_{1:0}) = \mathbf{P}(\mathbf{X}_0) \end{cases}$$

// Initial estimate (prior)

One-Step Prediction

$$\mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t+1}) = \alpha \mathbf{P}(\mathbf{e}_{t+1} \mid \mathbf{X}_{t+1})\ \mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t})$$

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

update

$$= \underbrace{\alpha \mathbf{P}(\mathbf{e}_{t+1} \mid \mathbf{X}_{t+1})}_{\text{sensor model}} \sum_{\mathbf{x}_t} \underbrace{\mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{x}_t)}_{\text{transition model}} \underbrace{P(\mathbf{x}_t \mid \mathbf{e}_{1:t})}_{\text{recursion}}$$

prediction

$$\begin{cases} \mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t+1}) = \text{FORWARD}(\underbrace{\mathbf{P}(\mathbf{X}_t \mid \mathbf{e}_{1:t})}_{\text{“forward” message}}, \mathbf{e}_{t+1}) \\ \mathbf{P}(\mathbf{X}_0 \mid \mathbf{e}_{1:0}) = \mathbf{P}(\mathbf{X}_0) \end{cases}$$

// Initial estimate (prior)

$$\mathbf{P}(\mathbf{X}_0) \rightarrow \mathbf{P}(\mathbf{X}_1 \mid \mathbf{e}_{1:1}) \rightarrow \cdots \rightarrow \mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{1:k}) \rightarrow \mathbf{P}(\mathbf{X}_{k+1} \mid \mathbf{e}_{1:k+1}) \rightarrow \cdots$$

One-Step Prediction

$$\boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t+1}) = \alpha \boldsymbol{P}(\boldsymbol{e}_{t+1} \mid \boldsymbol{X}_{t+1})\, \boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t})$$

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

$$= \overset{\text{update}}{\boxed{\alpha \underbrace{\boldsymbol{P}(\boldsymbol{e}_{t+1} \mid \boldsymbol{X}_{t+1})}_{\text{sensor model}}}}\ \boxed{\sum_{\boldsymbol{x}_t} \underbrace{\boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{x}_t)}_{\text{transition model}} \underbrace{P(\boldsymbol{x}_t \mid \boldsymbol{e}_{1:t})}_{\text{recursion}}}\ \text{prediction}$$

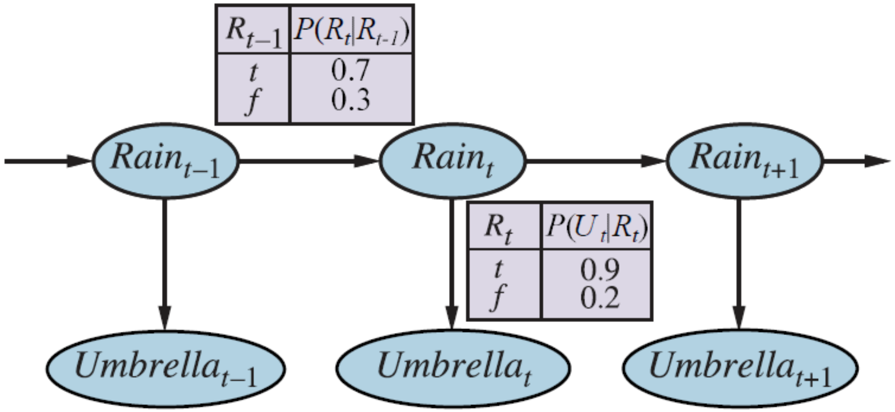
$$\begin{cases} \boldsymbol{P}(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t+1}) = \text{FORWARD}\big(\underbrace{\boldsymbol{P}(\boldsymbol{X}_t \mid \boldsymbol{e}_{1:t})}_{\text{“forward” message}}, \boldsymbol{e}_{t+1}\big) \\ \boldsymbol{P}(\boldsymbol{X}_0 \mid \boldsymbol{e}_{1:0}) = \boldsymbol{P}(\boldsymbol{X}_0) \end{cases}$$

// Initial estimate (prior)

$$\boldsymbol{P}(\boldsymbol{X}_0) \to \boldsymbol{P}(\boldsymbol{X}_1 \mid \boldsymbol{e}_{1:1}) \to \cdots \to \boldsymbol{P}(\boldsymbol{X}_k \mid \boldsymbol{e}_{1:k}) \to \boldsymbol{P}(\boldsymbol{X}_{k+1} \mid \boldsymbol{e}_{1:k+1}) \to \cdots$$

Time and space for the update at t must be *constant* in order to keep track of the current state distribution indefinitely.

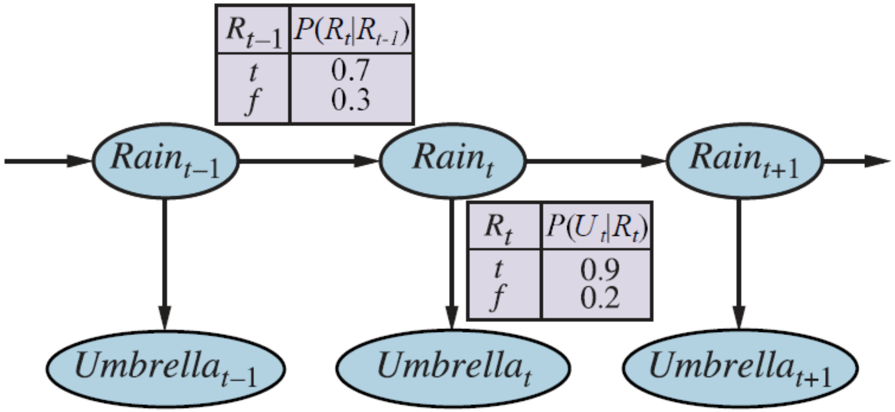
Filtering in the Umbrella World



Compute $\boldsymbol{P}(R_2 \mid u_{1:2})$ as follows:

$U_1 = true \wedge U_2 = true$

Filering in the Umbrella World



Compute $\boldsymbol{P}(R_2 \mid u_{1:2})$ as follows:

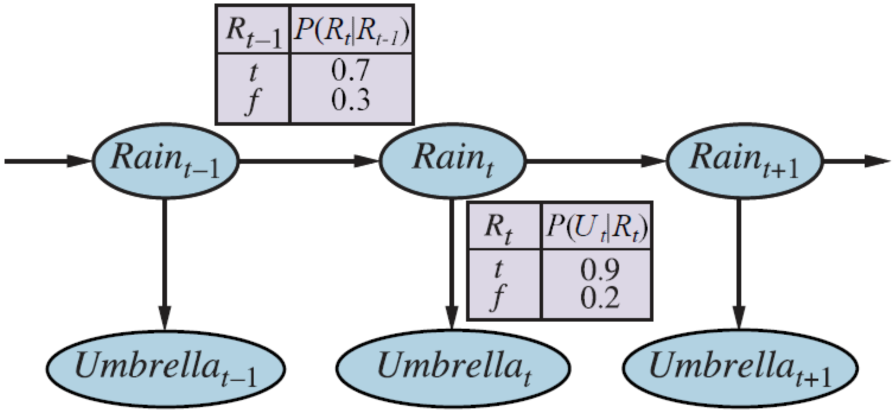
$U_1 = \textit{true} \wedge U_2 = \textit{true}$

- On day 0, no observation.

$$\boldsymbol{P}(R_0) = \langle 0.5, 0.5 \rangle$$

$$r_0, \neg r_0$$

Filtering in the Umbrella World



Compute $\boldsymbol{P}(R_2 \mid u_{1:2})$ as follows:

$U_1 = \mathit{true} \wedge U_2 = \mathit{true}$

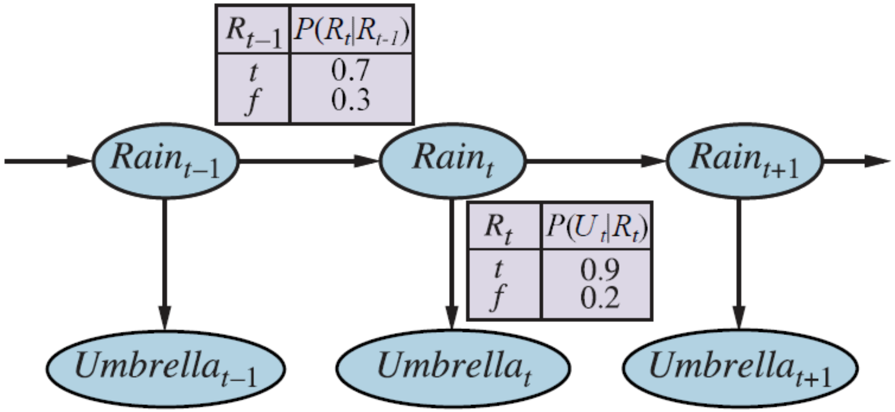
- On day 0, no observation.

$$\boldsymbol{P}(R_0) = \underset{r_0,\neg r_0}{\langle 0.5, 0.5 \rangle}$$

- On day 1, the umbrella appears.

$$u_1 \equiv (U_1 = \mathit{true})$$

Filtering in the Umbrella World



Compute $\boldsymbol{P}(R_2 \mid u_{1:2})$ as follows:

$U_1 = true \wedge U_2 = true$

- On day 0, no observation.

$$\boldsymbol{P}(R_0) = \langle \underset{r_0,}{0.5}, \underset{\neg r_0}{0.5} \rangle$$

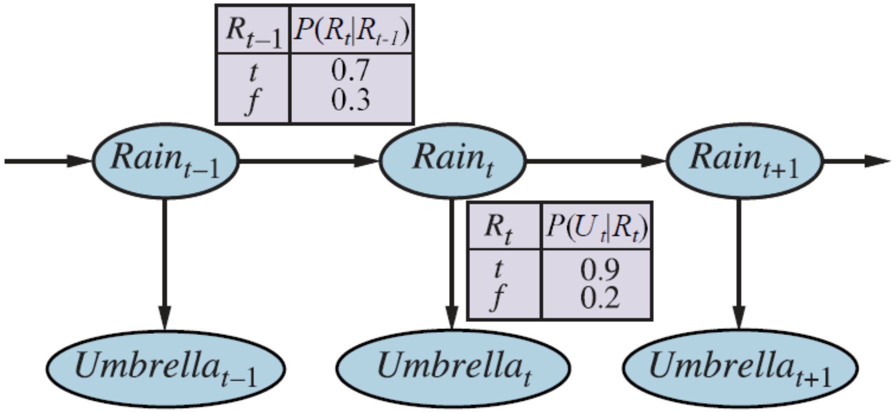
- On day 1, the umbrella appears.

$$u_1 \equiv (U_1 = true)$$

a) Prediction:

$$\boldsymbol{P}(R_1) = \sum_{r_0' \in \{r_0, \neg r_0\}} \boldsymbol{P}(R_1 \mid r_0') P(r_o') \qquad // \; r_0 \equiv (R_0 = true)$$

Filtering in the Umbrella World



Compute $\mathbf{P}(R_2 \mid u_{1:2})$ as follows:

$U_1 = true \wedge U_2 = true$

- On day 0, no observation.

$$\mathbf{P}(R_0) = \underset{r_0,\ \neg r_0}{\langle 0.5, 0.5 \rangle}$$

- On day 1, the umbrella appears.

$$u_1 \equiv (U_1 = true)$$

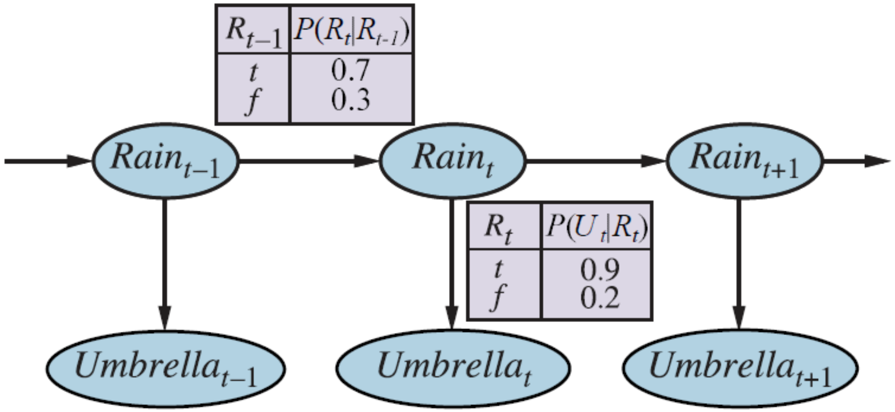
a) Prediction:

$$\mathbf{P}(R_1) = \sum_{r_0' \in \{r_0, \neg r_0\}} \mathbf{P}(R_1 \mid r_0') P(r_o')$$

// $r_0 \equiv (R_0 = true)$

$$= \langle 0.7, 1 - 0.7 \rangle \times 0.5 + \langle 0.3, 1 - 0.3 \rangle \times 0.5 = \langle 0.5, 0.5 \rangle$$

Filtering in the Umbrella World



Compute $\mathbf{P}(R_2 \mid u_{1:2})$ as follows:

$U_1 = true \wedge U_2 = true$

- On day 0, no observation.

$$\mathbf{P}(R_0) = \langle 0.5, 0.5 \rangle$$

$r_0, \neg r_0$

- On day 1, the umbrella appears.

$$u_1 \equiv (U_1 = true)$$

a) Prediction:

$$\mathbf{P}(R_1) = \sum_{r_0' \in \{r_0, \neg r_0\}} \mathbf{P}(R_1 \mid r_0') P(r_o')$$

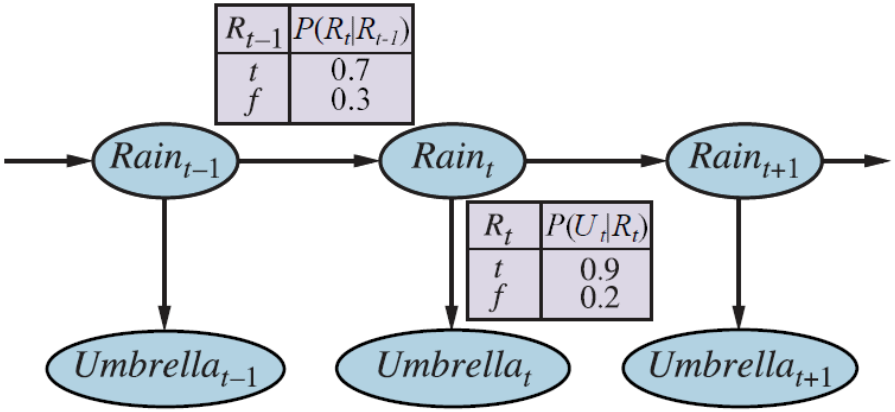
// $r_0 \equiv (R_0 = true)$

$$= \langle 0.7, 1 - 0.7 \rangle \times 0.5 + \langle 0.3, 1 - 0.3 \rangle \times 0.5 = \langle 0.5, 0.5 \rangle$$

b) Update with evidence for $t = 1$ followed by normalization:

$$\mathbf{P}(R_1 | \, u_1) = \alpha \mathbf{P}(u_1 \mid R_1) \mathbf{P}(R_1)$$

Filtering in the Umbrella World



Compute $\boldsymbol{P}(R_2 \mid u_{1:2})$ as follows:

$U_1 = true \wedge U_2 = true$

- On day 0, no observation.

$$\boldsymbol{P}(R_0) = \langle 0.5, 0.5 \rangle$$
$$r_0, \neg r_0$$

- On day 1, the umbrella appears.

$$u_1 \equiv (U_1 = true)$$

a) Prediction:

$$\boldsymbol{P}(R_1) = \sum_{r_0' \in \{r_0, \neg r_0\}} \boldsymbol{P}(R_1 \mid r_0') P(r_o') \qquad // \; r_0 \equiv (R_0 = true)$$

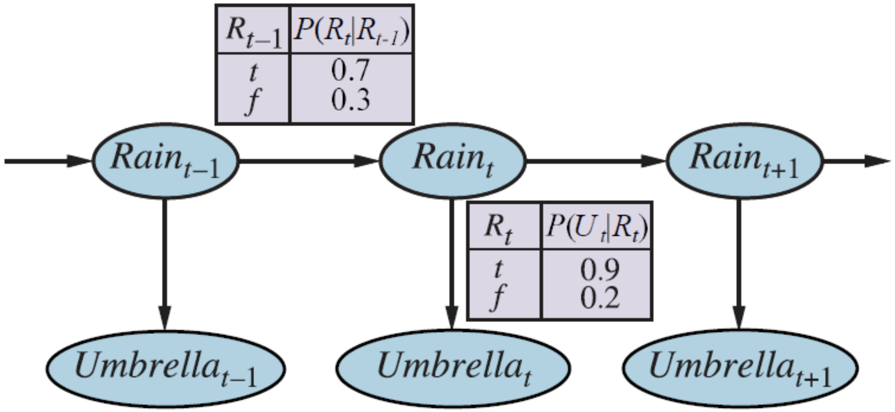
$$= \langle 0.7, 1-0.7 \rangle \times 0.5 + \langle 0.3, 1-0.3 \rangle \times 0.5 = \langle 0.5, 0.5 \rangle$$

b) Update with evidence for $t = 1$ followed by normalization:

$$\boldsymbol{P}(R_1 | \, u_1) = \alpha \boldsymbol{P}(u_1 \mid R_1) \boldsymbol{P}(R_1)$$

$$= \alpha \langle 0.9, 0.2 \rangle \, \langle 0.5, 0.5 \rangle$$

Filtering in the Umbrella World



R_{t-1}	$P(R_t \mid R_{t-1})$
t	0.7
f	0.3

R_t	$P(U_t \mid R_t)$
t	0.9
f	0.2

Compute $\boldsymbol{P}(R_2 \mid \underset{U_1 = true \wedge U_2 = true}{u_{1:2}})$ as follows:

- On day 0, no observation.

$$\boldsymbol{P}(R_0) = \underset{r_0,\, \neg r_0}{\langle 0.5, 0.5\rangle}$$

- On day 1, the umbrella appears.

$$u_1 \equiv (U_1 = true)$$

a) Prediction:

$$\boldsymbol{P}(R_1) = \sum_{r_0' \in \{r_0, \neg r_0\}} \boldsymbol{P}(R_1 \mid r_0') P(r_o') \qquad // \ r_0 \equiv (R_0 = true)$$

$$= \langle 0.7, 1 - 0.7\rangle \times 0.5 + \langle 0.3, 1 - 0.3\rangle \times 0.5 = \langle 0.5, 0.5\rangle$$

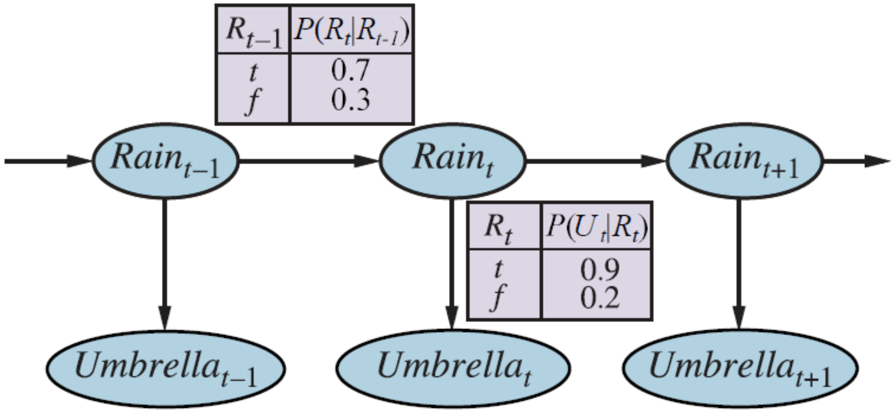
b) Update with evidence for $t = 1$ followed by normalization:

$$\boldsymbol{P}(R_1 \mid u_1) = \alpha \boldsymbol{P}(u_1 \mid R_1) \boldsymbol{P}(R_1)$$

$$= \alpha \langle 0.9, 0.2\rangle \langle 0.5, 0.5\rangle$$

$$= \alpha \langle 0.45, 0.1\rangle$$

Filtering in the Umbrella World



R_{t-1}	$P(R_t \mid R_{t-1})$
t	0.7
f	0.3

R_t	$P(U_t \mid R_t)$
t	0.9
f	0.2

Compute $\boldsymbol{P}(R_2 \mid u_{1:2})$ as follows:

$U_1 = \mathit{true} \wedge U_2 = \mathit{true}$

- On day 0, no observation.

$$\boldsymbol{P}(R_0) = \underset{r_0,\ \neg r_0}{\langle 0.5, 0.5 \rangle}$$

- On day 1, the umbrella appears.

$$u_1 \equiv (U_1 = \mathit{true})$$

a) Prediction:

$$\boldsymbol{P}(R_1) = \sum_{r_0' \in \{r_0, \neg r_0\}} \boldsymbol{P}(R_1 \mid r_0') P(r_o') \qquad // \ r_0 \equiv (R_0 = \mathit{true})$$

$$= \langle 0.7, 1 - 0.7 \rangle \times 0.5 + \langle 0.3, 1 - 0.3 \rangle \times 0.5 = \langle 0.5, 0.5 \rangle$$

b) Update with evidence for $t = 1$ followed by normalization:

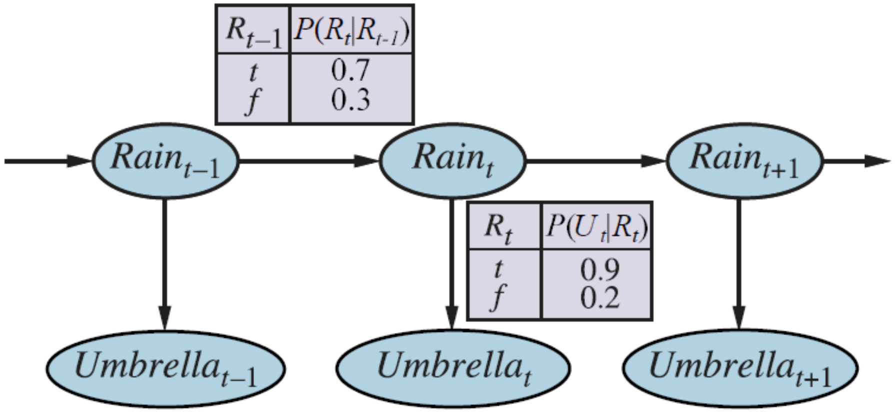
$$\boldsymbol{P}(R_1 \mid u_1) = \alpha \boldsymbol{P}(u_1 \mid R_1) \boldsymbol{P}(R_1)$$

$$= \alpha \langle 0.9, 0.2 \rangle \langle 0.5, 0.5 \rangle$$

$$= \alpha \langle 0.45, 0.1 \rangle$$

$$\approx \langle 0.818, 0.182 \rangle$$

Filtering in UW (cont'd)

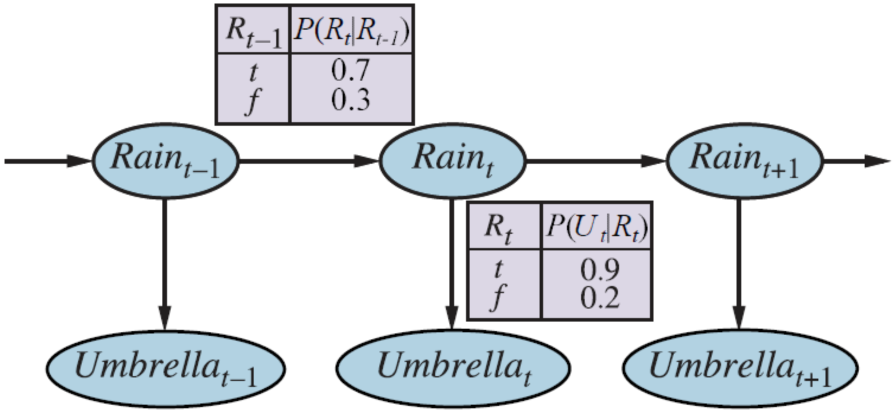


$$\boldsymbol{P}(R_1 \mid u_1) \approx \langle 0.818, 0.182 \rangle$$

- On day 2, the umbrella appears.

$$u_2 \equiv (U_2 = \mathit{true})$$

Filtering in UW (cont’d)



R_{t-1}	$P(R_t \mid R_{t-1})$
t	0.7
f	0.3

R_t	$P(U_t \mid R_t)$
t	0.9
f	0.2

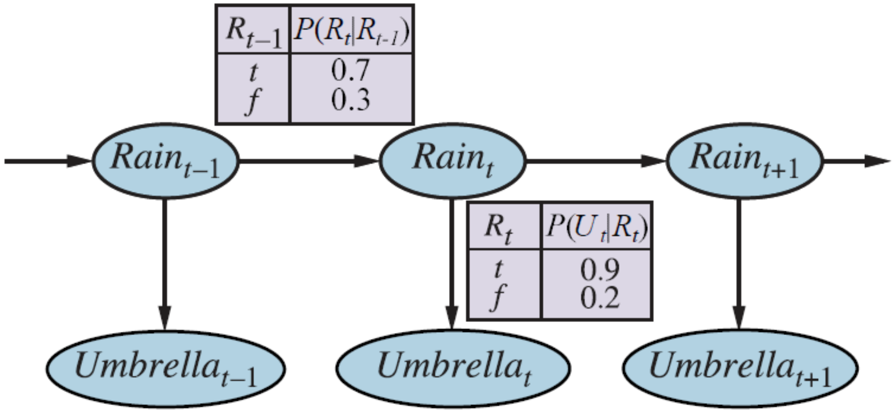
$$\boldsymbol{P}(R_1 \mid u_1) \approx \langle 0.818, 0.182 \rangle$$

- On day 2, the umbrella appears.

$$u_2 \equiv (U_2 = true)$$

a) Prediction: $\boldsymbol{P}(R_2 \mid u_1) = \sum_{r_1' \in \{r_1, \neg r_1\}} \boldsymbol{P}(R_2 \mid r_1') P(r_1' \mid u_1)$ // $r_1 \equiv (R_1 = true)$

Filtering in UW (cont'd)



$$\boldsymbol{P}(R_1 \mid u_1) \approx \langle 0.818, 0.182\rangle$$

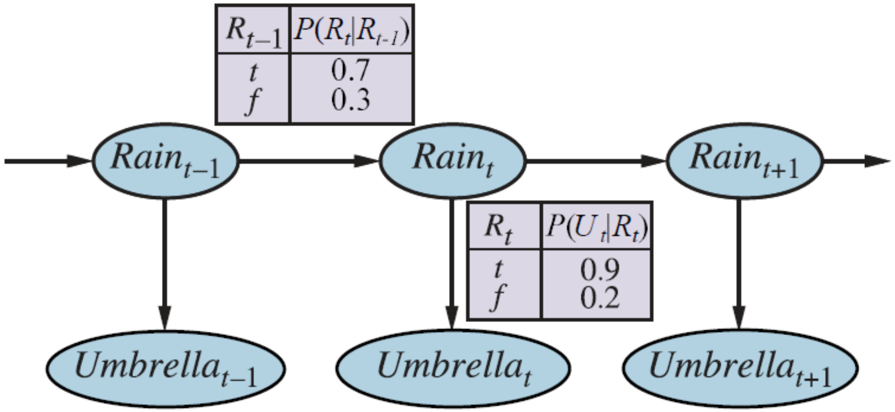
- On day 2, the umbrella appears.

$$u_2 \equiv (U_2 = \textit{true})$$

a) Prediction: $\boldsymbol{P}(R_2 \mid u_1) = \sum_{r_1' \in \{r_1, \neg r_1\}} \boldsymbol{P}(R_2 \mid r_1') P(r_1' \mid u_1)$ // $r_1 \equiv (R_1 = \textit{true})$

$$= \langle 0.7, 0.3\rangle \times 0.818 + \langle 0.3, 0.7\rangle \times 0.182 \approx \langle 0.627, 0.373\rangle$$

Filtering in UW (cont'd)



$$\boldsymbol{P}(R_1 \mid u_1) \approx \langle 0.818, 0.182 \rangle$$

- On day 2, the umbrella appears.

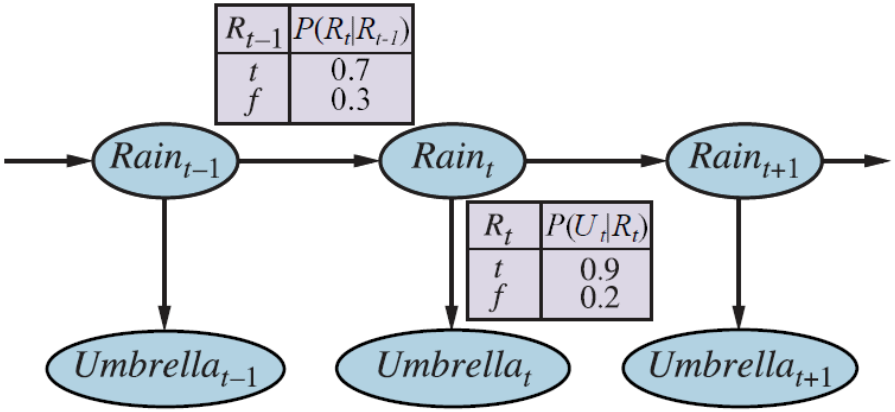
$$u_2 \equiv (U_2 = \mathit{true})$$

a) Prediction: $\boldsymbol{P}(R_2 \mid u_1) = \sum_{r_1' \in \{r_1, \neg r_1\}} \boldsymbol{P}(R_2 \mid r_1') P(r_1' \mid u_1)$ // $r_1 \equiv (R_1 = \mathit{true})$

$$= \langle 0.7, 0.3 \rangle \times 0.818 + \langle 0.3, 0.7 \rangle \times 0.182 \approx \langle 0.627, 0.373 \rangle$$

b) Update with evidence for $t = 2$: $\boldsymbol{P}(R_2 \mid u_1, u_2) = \alpha \boldsymbol{P}(u_2 \mid R_2) \boldsymbol{P}(R_2 \mid u_1)$

Filtering in UW (cont'd)



$$\boldsymbol{P}(R_1 \mid u_1) \approx \langle 0.818, 0.182 \rangle$$

- On day 2, the umbrella appears.

$$u_2 \equiv (U_2 = \mathit{true})$$

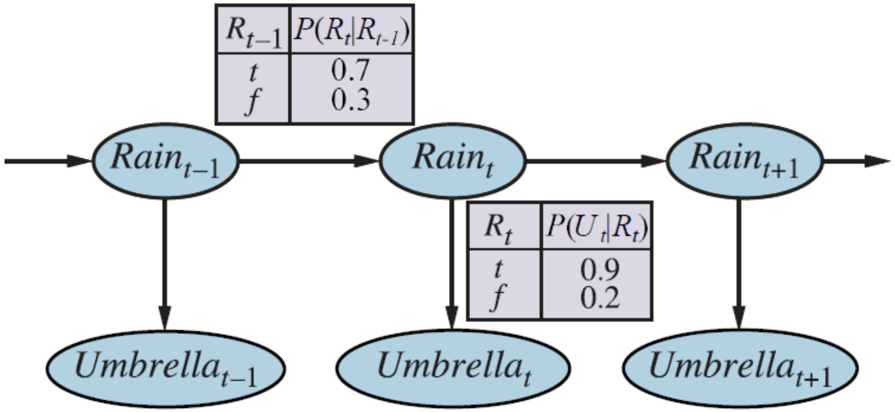
a) Prediction: $\boldsymbol{P}(R_2 \mid u_1) = \sum_{r_1' \in \{r_1, \neg r_1\}} \boldsymbol{P}(R_2 \mid r_1') P(r_1' \mid u_1)$ // $r_1 \equiv (R_1 = \mathit{true})$

$$= \langle 0.7, 0.3 \rangle \times 0.818 + \langle 0.3, 0.7 \rangle \times 0.182 \approx \langle 0.627, 0.373 \rangle$$

b) Update with evidence for $t = 2$: $\boldsymbol{P}(R_2 \mid u_1, u_2) = \alpha \boldsymbol{P}(u_2 \mid R_2) \boldsymbol{P}(R_2 \mid u_1)$

$$\approx \alpha \langle 0.9, 0.2 \rangle \langle 0.627, 0.373 \rangle$$

Filtering in UW (cont'd)



$$\boldsymbol{P}(R_1 \mid u_1) \approx \langle 0.818, 0.182 \rangle$$

- On day 2, the umbrella appears.

$$u_2 \equiv (U_2 = \mathit{true})$$

a) Prediction: $$\boldsymbol{P}(R_2 \mid u_1) = \sum_{r_1' \in \{r_1, \neg r_1\}} \boldsymbol{P}(R_2 \mid r_1') P(r_1' \mid u_1)$$ // $r_1 \equiv (R_1 = \mathit{true})$

$$= \langle 0.7, 0.3 \rangle \times 0.818 + \langle 0.3, 0.7 \rangle \times 0.182 \approx \langle 0.627, 0.373 \rangle$$

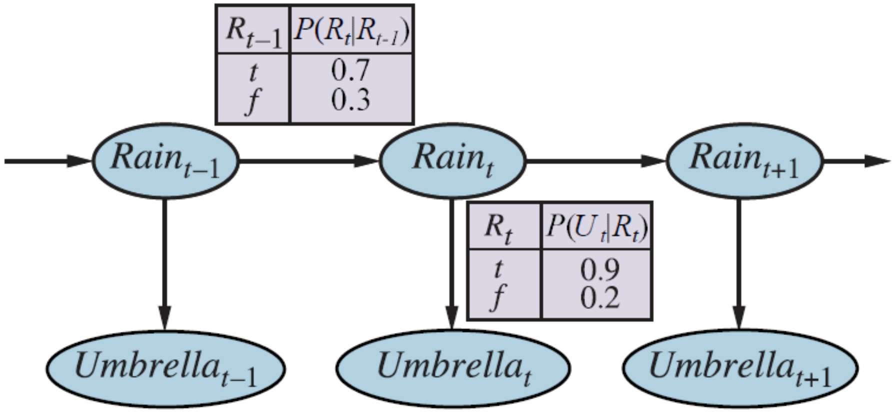
b) Update with evidence for $t = 2$: $$\boldsymbol{P}(R_2 \mid u_1, u_2) = \alpha \boldsymbol{P}(u_2 \mid R_2) \boldsymbol{P}(R_2 \mid u_1)$$

$$\approx \alpha \langle 0.9, 0.2 \rangle \langle 0.627, 0.373 \rangle$$

$$\approx \alpha \langle 0.565, 0.075 \rangle$$

$$\approx \langle 0.883, 0.117 \rangle$$

Filtering in UW (cont'd)



R_{t-1}	$P(R_t \mid R_{t-1})$
t	0.7
f	0.3

R_t	$P(U_t \mid R_t)$
t	0.9
f	0.2

$$\boldsymbol{P}(R_1 \mid u_1) \approx \langle 0.818, 0.182 \rangle$$

- On day 2, the umbrella appears.

$$u_2 \equiv (U_2 = \mathit{true})$$

a) Prediction: $$\boldsymbol{P}(R_2 \mid u_1) = \sum_{r_1' \in \{r_1, \neg r_1\}} \boldsymbol{P}(R_2 \mid r_1') P(r_1' \mid u_1) \qquad // \; r_1 \equiv (R_1 = \mathit{true})$$

$$= \langle 0.7, 0.3 \rangle \times 0.818 + \langle 0.3, 0.7 \rangle \times 0.182 \approx \langle 0.627, 0.373 \rangle$$

b) Update with evidence for $t = 2$: $$\boldsymbol{P}(R_2 \mid u_1, u_2) = \alpha \boldsymbol{P}(u_2 \mid R_2) \boldsymbol{P}(R_2 \mid u_1)$$

$$\approx \alpha \langle 0.9, 0.2 \rangle \langle 0.627, 0.373 \rangle$$

$$\approx \alpha \langle 0.565, 0.075 \rangle$$

$$\approx \langle 0.883, 0.117 \rangle$$

For Kalman filtering and its subroutine recursive least-squares (updating states out of a **continuum**), we refer to
https://faculty.sites.iastate.edu/jia/files/inline-files/kalman-filter.pdf
https://faculty.sites.iastate.edu/jia/files/inline-files/recursive-least-squares.pdf