CRICOS PROVIDER 00123M

Faculty of SET / School of Computer and Mathematical Sciences

COMP SCI 3007/7059/7659
Artificial Intelligence
Probability Reasoning Over Time 2 - Viterbi Algorithm

Acknowledgement of Country

We acknowledge and pay our respects to the Kaurna people, the traditional custodians whose ancestral lands we gather on.

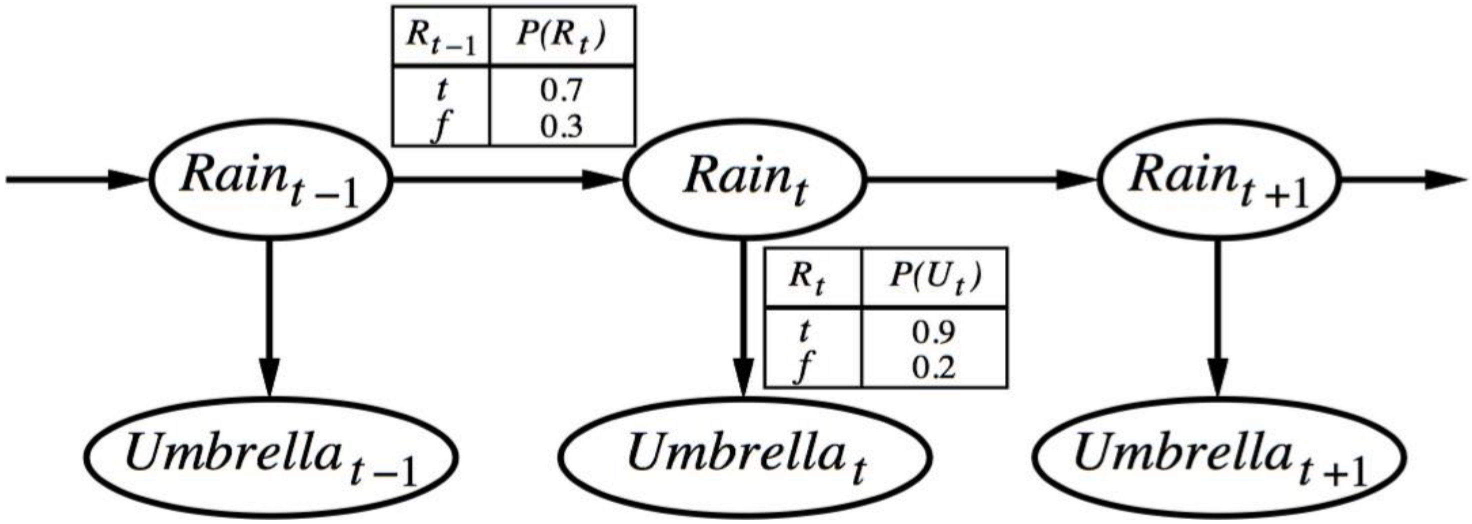
We acknowledge the deep feelings of attachment and relationship of the Kaurna people to the country and we respect and value their past, present and ongoing connection to the land and cultural beliefs.

Smoothing & Viterbi Algorithm

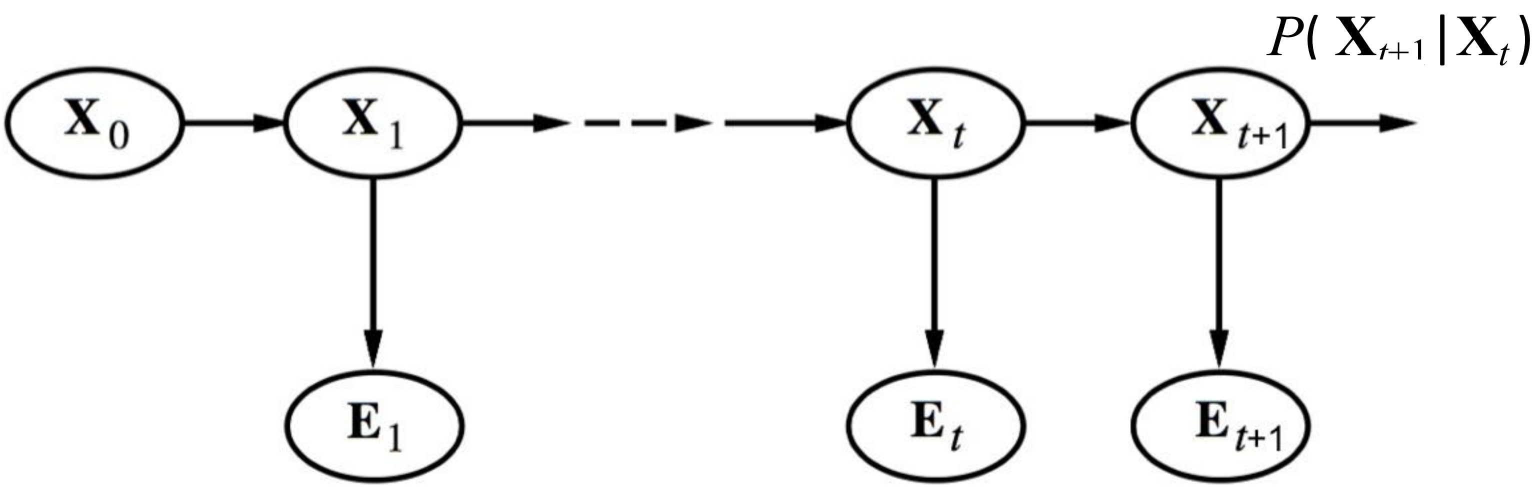
AIMA C15.2

Example: is it raining outside?

- A commonly used temporal model for this kind of problem: Hidden Markov Model (HMM)



The general case

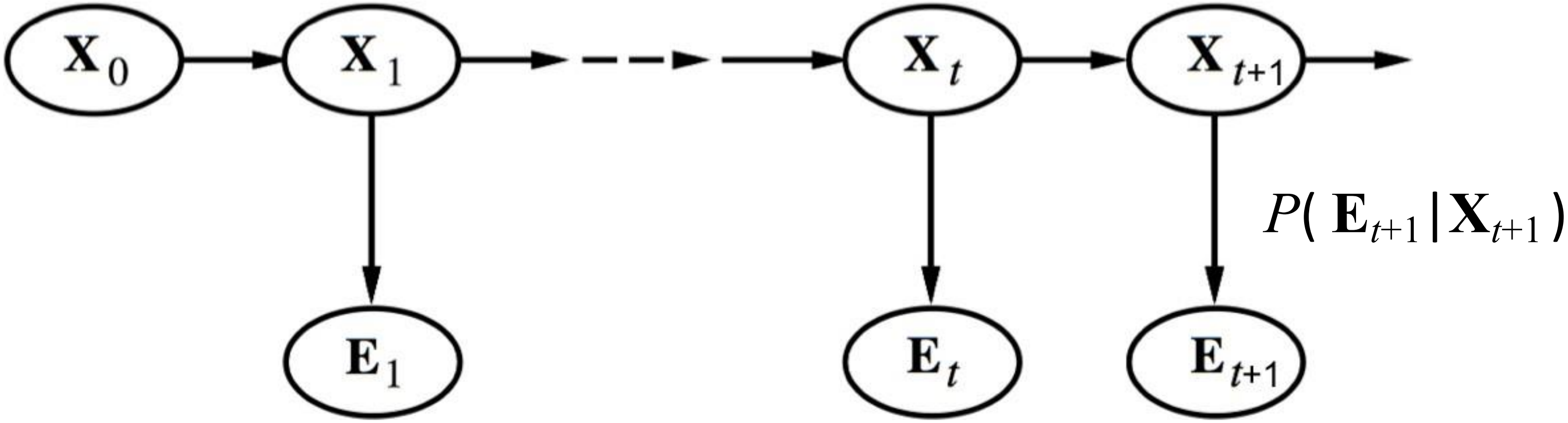


- State transition model:

$$P(\mathbf{X}_{t+1} | \mathbf{X}_0, \ldots, \mathbf{X}_t) = P(\mathbf{X}_{t+1} | \mathbf{X}_t)$$

- **First order Markov assumption**: the present state depends only on the immediate previous state.

The general case



- Observation/emission/sensor model

$$P(\mathbf{E}_{t+1} | \mathbf{X}_{0:t+1}, \mathbf{E}_{1:t}) = P(\mathbf{E}_{t+1} | \mathbf{X}_{t+1})$$

- Sensor Markov assumption: the probability of observing $\mathbf{E}_t$ depends only on the state $\mathbf{X}_t$.

*Note: $\mathbf{X}_{0:t} = \mathbf{X}_0, \mathbf{X}_1, \ldots, \mathbf{X}_t$

Filtering

- We have observed $\mathbf{e}_1, \ldots, \mathbf{e}_{t+1} = \mathbf{e}_{1:t+1}$. We wish to calculate

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

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

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

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

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

$$= \alpha\, \underbrace{\mathbf{P}(\mathbf{e}_{t+1} \mid \mathbf{X}_{t+1})}_{\text{Observation model}} \sum_{\mathbf{x}_t} \underbrace{\mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{x}_t)}_{\text{Transition model}} P(\mathbf{x}_t \mid \mathbf{e}_{1:t}) \quad \text{(Markov assumption).}$$

Filtering

- We have observed $\mathbf{e}_1, \ldots, \mathbf{e}_{t+1} = \mathbf{e}_{1:t+1}$. We wish to calculate

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

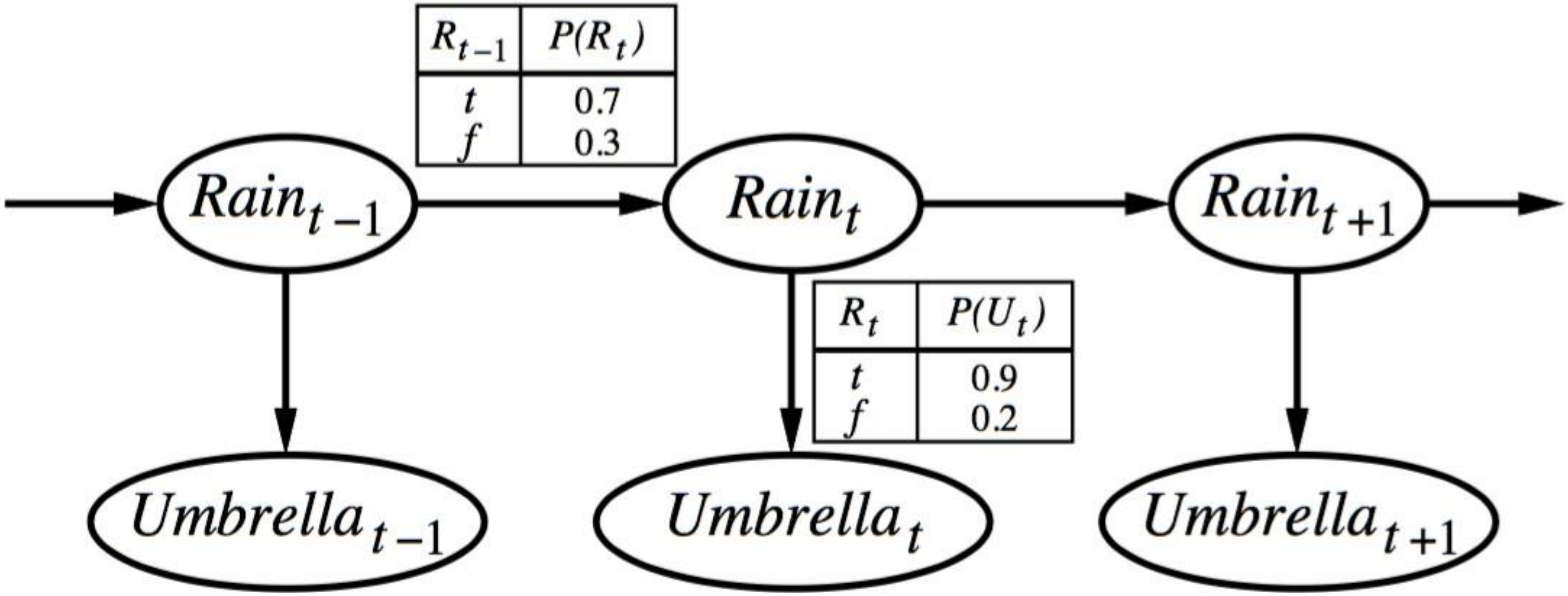
$$\begin{aligned}
\boxed{\mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t+1})} &= \mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t}, \mathbf{e}_{t+1}) \quad \text{(dividing up the evidence)} \\
&= \alpha\, \mathbf{P}(\mathbf{e}_{t+1} \mid \mathbf{X}_{t+1}, \mathbf{e}_{1:t})\, \mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t}) \quad \text{(using Bayes' rule)} \\
&= \alpha\, \mathbf{P}(\mathbf{e}_{t+1} \mid \mathbf{X}_{t+1})\, \underline{\mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t})} \quad \text{(by the sensor Markov assumption).} \\
&= \alpha\, \mathbf{P}(\mathbf{e}_{t+1} \mid \mathbf{X}_{t+1}) \sum_{\mathbf{x}_t} \mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{x}_t, \mathbf{e}_{1:t}) P(\mathbf{x}_t \mid \mathbf{e}_{1:t}) \\
&= \alpha\, \mathbf{P}(\mathbf{e}_{t+1} \mid \mathbf{X}_{t+1}) \underbrace{\sum_{\mathbf{x}_t} \mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{x}_t) \boxed{P(\mathbf{x}_t \mid \mathbf{e}_{1:t})}} \quad \text{(Markov assumption).}
\end{aligned}$$

Forward

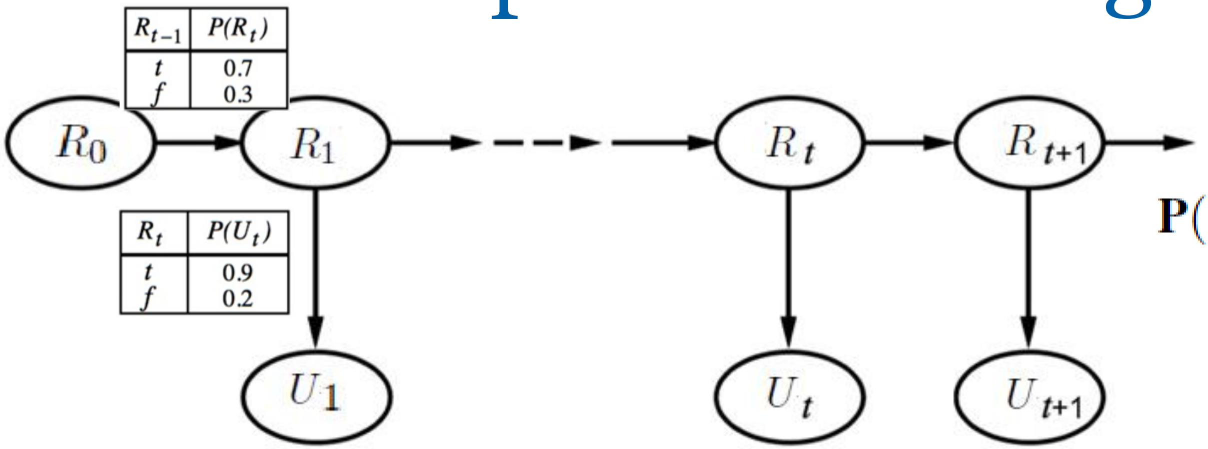
Calculating this is called prediction.

Example: is it raining outside?

- Form it as a first-order Markov process:



Example: is it raining outside?



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

- On day 0, we have no observations, only the security guard's prior beliefs; let's assume that consists of $\mathbf{P}(R_0) = \langle 0.5, 0.5 \rangle$.
- On day 1, the umbrella appears, so $U_1 = true$.

$$\mathbf{P}(R_1 \mid u_1) = \alpha\, \mathbf{P}(u_1 \mid R_1) \sum_{r_0} \mathbf{P}(R_1 \mid r_0) P(r_0)$$

$\langle P(r_1=t|r_0=t), P(r_1=f|r_0=t)\rangle$ $\langle P(r_1=t|r_0=f), P(r_1=f|\neg r_0=f)\rangle$

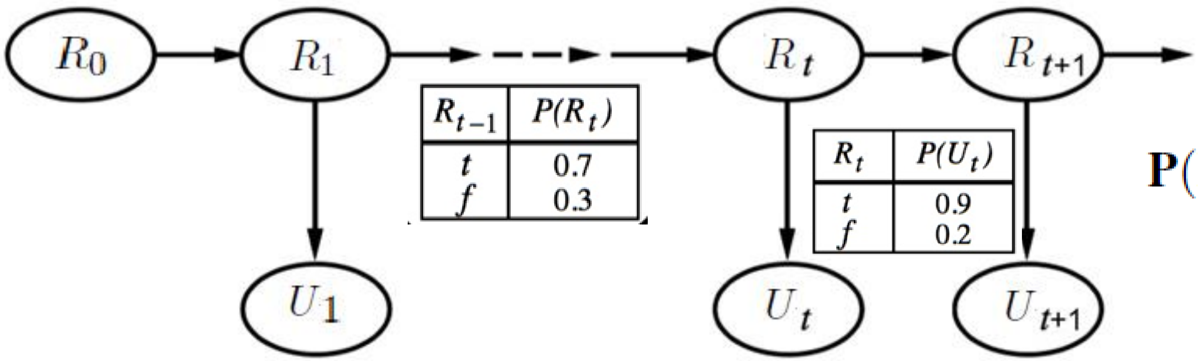
$$= \alpha \langle 0.9, 0.2 \rangle \left(\langle 0.7, 0.3 \rangle \times 0.5 + \langle 0.3, 0.7 \rangle \times 0.5 \right)$$

$\langle P(u_1=t|r_1=t), P(u_1=t|r_1=f)\rangle$ $P(r_0=t)$ $P(r_0=f)$

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

$$= \alpha \langle 0.45, 0.1 \rangle \approx \langle 0.818, 0.182 \rangle$$

Example: is it raining outside?



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

R_t	$P(U_t)$
t	0.9
f	0.2

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

- On day 2, the umbrella appears, so $U_2 = true$.

$$\mathbf{P}(R_2 \mid u_1, u_2) = \alpha\, \mathbf{P}(u_2 \mid R_2) \sum_{r_1} \mathbf{P}(R_2 \mid r_1) P(r_1 \mid u_1)$$

$$= \alpha \langle 0.9, 0.2 \rangle \left(\langle 0.7, 0.3 \rangle \times 0.818 + \langle 0.3, 0.7 \rangle \times 0.182 \right)$$

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

$$= \alpha \langle 0.565, 0.075 \rangle \approx \langle 0.883, 0.117 \rangle$$

Can keep on going as new observations are made.

Prediction

- We could see that the task of **prediction** can be seen simply as filtering without the addition of new evidence $\mathbf{e}_{t+1}$

- The **Filtering** process already incorporates a one-step prediction.

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

Filtering

- We have observed $\mathbf{e}_1, \ldots, \mathbf{e}_{t+1} = \mathbf{e}_{1:t+1}$. We wish to calculate

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

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

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

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

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

$$= \alpha\, \mathbf{P}(\mathbf{e}_{t+1} \mid \mathbf{X}_{t+1}) \underbrace{\sum_{\mathbf{x}_t} \mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{x}_t) \boxed{P(\mathbf{x}_t \mid \mathbf{e}_{1:t})}} \quad \text{(Markov assumption).}$$

Forward

Calculating this is called prediction.

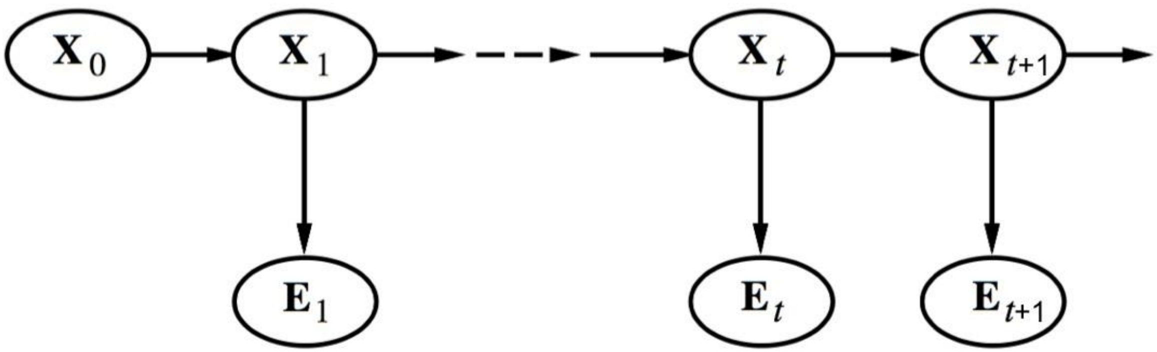
Prediction

- One-step Prediction:

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

- Prediction for *k* steps later:

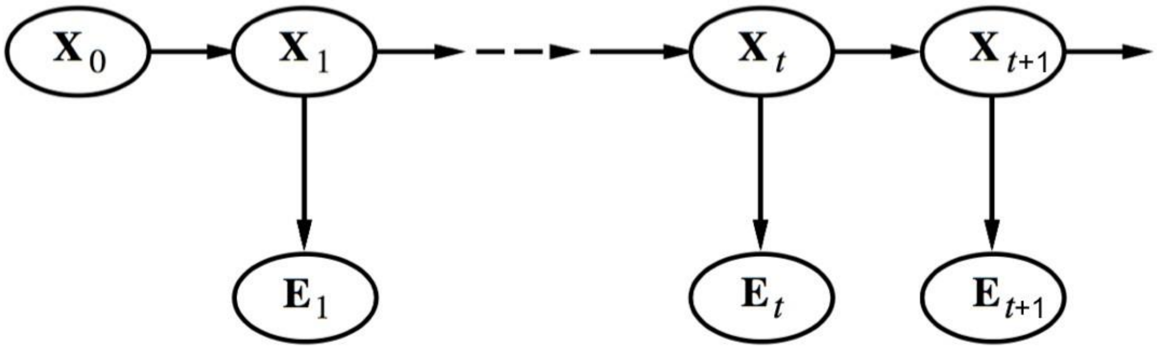
$$\mathbf{P}(\mathbf{X}_{t+k+1} \mid \mathbf{e}_{1:t}) = \sum_{\mathbf{x}_{t+k}} \mathbf{P}(\mathbf{X}_{t+k+1} \mid \mathbf{x}_{t+k}) P(\mathbf{x}_{t+k} \mid \mathbf{e}_{1:t})$$



Smoothing

- Smoothing computes the distribution over **past states** given evidence **up to** the present

$$\mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{1:t}) \text{ for } 0 \leq k < t$$



Smoothing

- Smoothing computes the distribution over **past states** given evidence **up to** the present

$$\mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{1:t}) \text{ for } 0 \le k < t$$

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

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

$$= \alpha\, \underbrace{\boxed{\mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{1:k})}}_{\text{Filtering (Forward)}} \underbrace{\mathbf{P}(\mathbf{e}_{k+1:t} \mid \mathbf{X}_k)}_{?} \quad \text{(conditional independence)}$$

ChatGPT prompt:

Show P(X_k|e_{1:k}, e_{k+1:t}) = \alpha P(X_k|e_{1:k})P(e_{k+1:t}|X_k, e_{1:k}) using Bayes' rule. The X here denotes hidden variables, k is an intermediate step between 0 and t, e denotes the obseveration variable

This is LaTex syntex, you can try overleaf

$$P(X_k | e_{1:t})$$

$$= P(\underbrace{X_k}_{A} | \underbrace{e_{1:k}}_{B}, \underbrace{e_{k+1:t}}_{C})$$

$$= \frac{P(A,B,C)}{P(B,C)} = \alpha P(A,B,C)$$

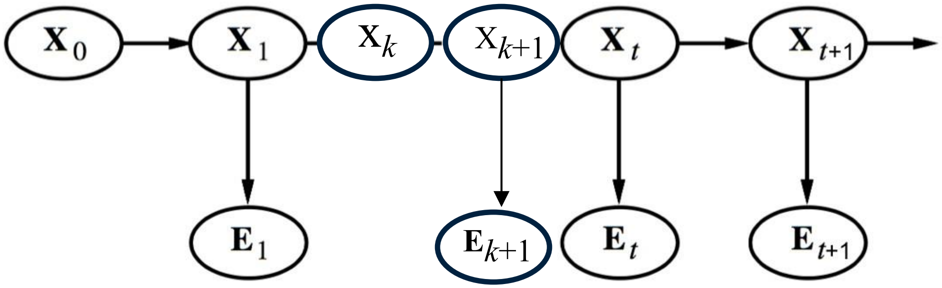
$$= \alpha P(A,B) \cdot P(C|A,B)$$

$$= \alpha P(A|B) \cdot P(B) \cdot P(C|A,B)$$

absorbed into

$$= \alpha P(A|B) \cdot P(C|A,B)$$

$$= \alpha P(X_k | e_{1:k}) \cdot P(e_{k+1:t} | X_k, e_{1:k})$$



$$\boxed{\mathbf{P}(\mathbf{e}_{k+1:t} \mid \mathbf{X}_k)}$$

$$= \sum_{\mathbf{x}_{k+1}} \mathbf{P}(\mathbf{e}_{k+1:t} \mid \mathbf{X}_k, \mathbf{x}_{k+1}) \mathbf{P}(\mathbf{x}_{k+1} \mid \mathbf{X}_k) \quad (\text{conditioning on } \mathbf{X}_{k+1})$$

$$= \sum_{\mathbf{x}_{k+1}} P(\mathbf{e}_{k+1:t} \mid \mathbf{x}_{k+1}) \mathbf{P}(\mathbf{x}_{k+1} \mid \mathbf{X}_k) \quad (\text{conditional independence})$$

(Sensor Markov assumption)

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

Observation model Transition model

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

(conditional independence of $\mathbf{e}_{k+1}$ and $\mathbf{e}_{k+2:t}$, given $\mathbf{X}_{k+1}$)

$$\mathbf{b}_{k+1:t} = \textsc{Backward}(\mathbf{b}_{k+2:t}, \mathbf{e}_{k+1})$$

Smoothing

- Smoothing computes the distribution over **past states** given evidence **up to** the present

$$\mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{1:t}) \text{ for } 0 \le k < t$$

$$\begin{aligned}
&= \mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{1:k}, \mathbf{e}_{k+1:t}) \\
&= \alpha\, \mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{1:k})\mathbf{P}(\mathbf{e}_{k+1:t} \mid \mathbf{X}_k, \mathbf{e}_{1:k}) \quad \text{(Bayes' rule)} \\
&= \alpha\, \boxed{\mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{1:k})}\boxed{\mathbf{P}(\mathbf{e}_{k+1:t} \mid \mathbf{X}_k)} \quad \text{(conditional independence)}
\end{aligned}$$

Filtering/Forward **Backward**

$$= \alpha\, \mathbf{f}_{1:k} \times \mathbf{b}_{k+1:t}$$

$$\mathbf{f}_{1:k+1} = \text{FORWARD}(\mathbf{f}_{1:k}, \mathbf{e}_{k+1})$$

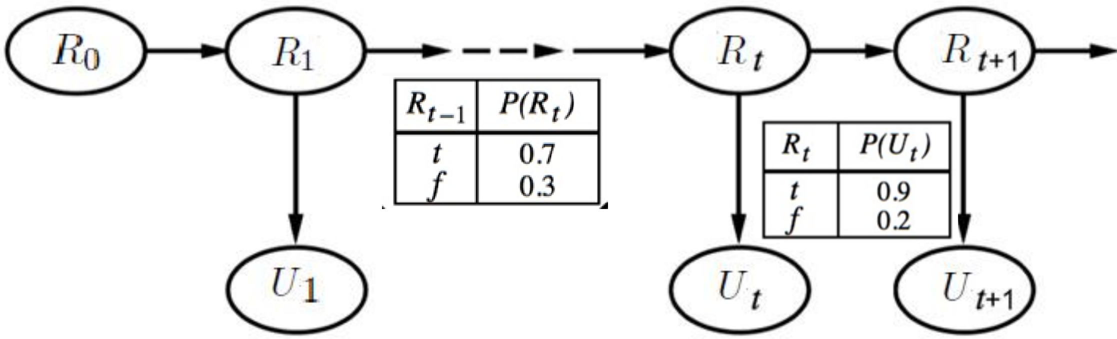
$$\mathbf{b}_{k+1:t} = \text{BACKWARD}(\mathbf{b}_{k+2:t}, \mathbf{e}_{k+1})$$

Initialize:

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

Because $\mathbf{e}_{t+1:t}$ is an empty sequence, the probability of observing it is 1.

Example: was it raining outside?



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

R_t	$P(U_t)$
t	0.9
f	0.2

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

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

Question:

Was it raining outside at day 1, given the observation on day 1 and 2?

$$\mathbf{P}(R_1 \mid u_1, u_2) = \alpha\, \mathbf{P}(R_1 \mid u_1)\, \mathbf{P}(u_2 \mid R_1)$$
$$= \alpha \, \langle 0.818, 0.182 \rangle \sum_{r_2} P(u_2 \mid r_2) P(\ \mid r_2) \mathbf{P}(r_2 \mid R_1)$$

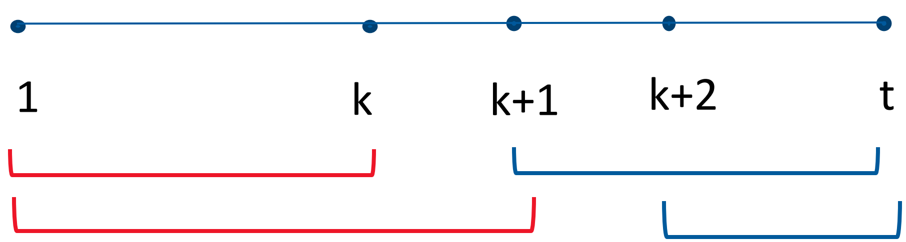
With k=1

$$= \alpha \, \langle 0.818, 0.182 \rangle \, (0.9 \times 1 \times \langle 0.7, 0.3 \rangle + 0.2 \times 1 \times \langle 0.3, 0.7 \rangle)$$
$$= \alpha \, \langle 0.818, 0.182 \rangle \, \langle 0.69, 0.41 \rangle$$
$$\approx \langle 0.883, 0.117 \rangle$$

Smoothing

$$\mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{1:t}) = \alpha\, \mathbf{f}_{1:k} \times \mathbf{b}_{k+1:t}$$

$$\mathbf{f}_{1:k+1} = \text{FORWARD}(\mathbf{f}_{1:k}, \mathbf{e}_{k+1}) \quad \mathbf{b}_{k+1:t} = \text{BACKWARD}(\mathbf{b}_{k+2:t}, \mathbf{e}_{k+1})$$



- Time complexity of smoothing at a single time step with the observations $e_{1:t}$: O(t), the whole sequence: O(t^2) in worst case.

Smoothing

```
function FORWARD-BACKWARD(ev, prior) returns a vector of probability distributions
  inputs: ev, a vector of evidence values for steps 1, ..., t
          prior, the prior distribution on the initial state, P(X_0)
  local variables: fv, a vector of forward messages for steps 0, ..., t
                   b, a representation of the backward message, initially all 1s
                   sv, a vector of smoothed estimates for steps 1, ..., t

  fv[0] ← prior
  for i = 1 to t do
     fv[i] ← FORWARD(fv[i − 1], ev[i])
  for i = t downto 1 do
     sv[i] ← NORMALIZE(fv[i] × b)
     b ← BACKWARD(b, ev[i])
  return sv
```

- Forward-Backward algorithm for smoothing the whole sequence: record the results of forward filtering over the whole sequence: O(t).

So far we learnt...

- Filtering

$$\mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{1:k})$$

- Prediction

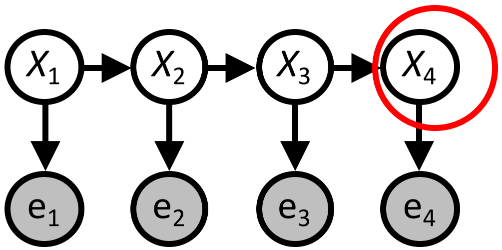
$$\mathbf{P}(\mathbf{X}_{t+k+1} \mid \mathbf{e}_{1:t})$$

- Smoothing

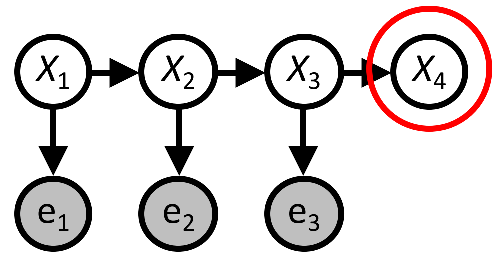
$$\mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{1:t}) \text{ for } 0 \leq k < t$$

Inference tasks

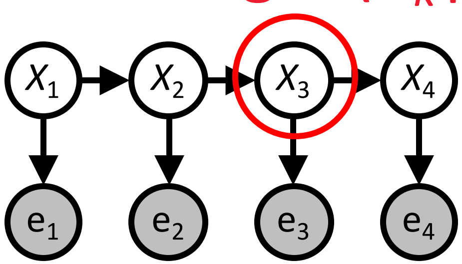
Filtering: $P(X_t | e_{1:t})$



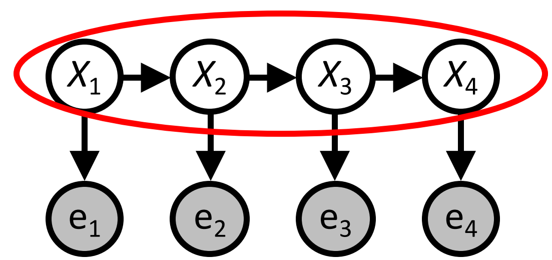
Prediction: $P(X_{t+k} | e_{1:t})$



Smoothing: $P(X_k | e_{1:t})$, $k<t$



Explanation: $P(X_{1:t} | e_{1:t})$



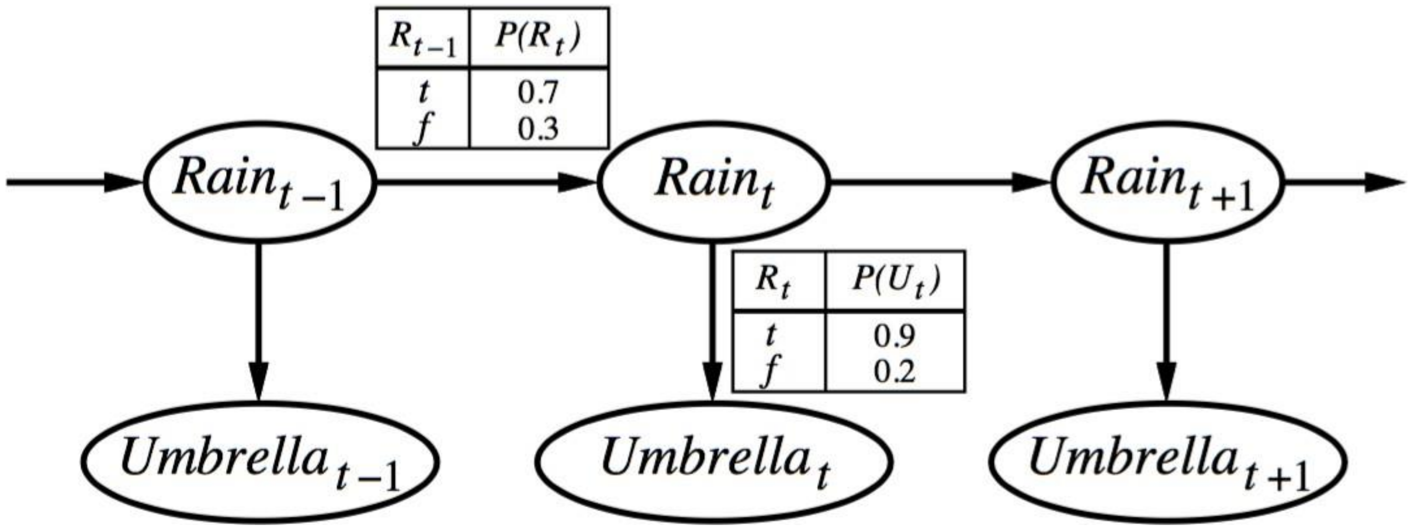
Viterbi Algorithm

- Finding the most likely sequence (i.e., Explanation)
  - Given a sequence of observations, the sequence of states that is *most* likely to have generated those observations.

$$\operatorname{argmax}_{\mathbf{x}_{1:t}} P(\mathbf{x}_{1:t} \mid \mathbf{e}_{1:t})$$

- Some applications
  - Speech recognition
  - Sequence tagging
  - …

The rain problem

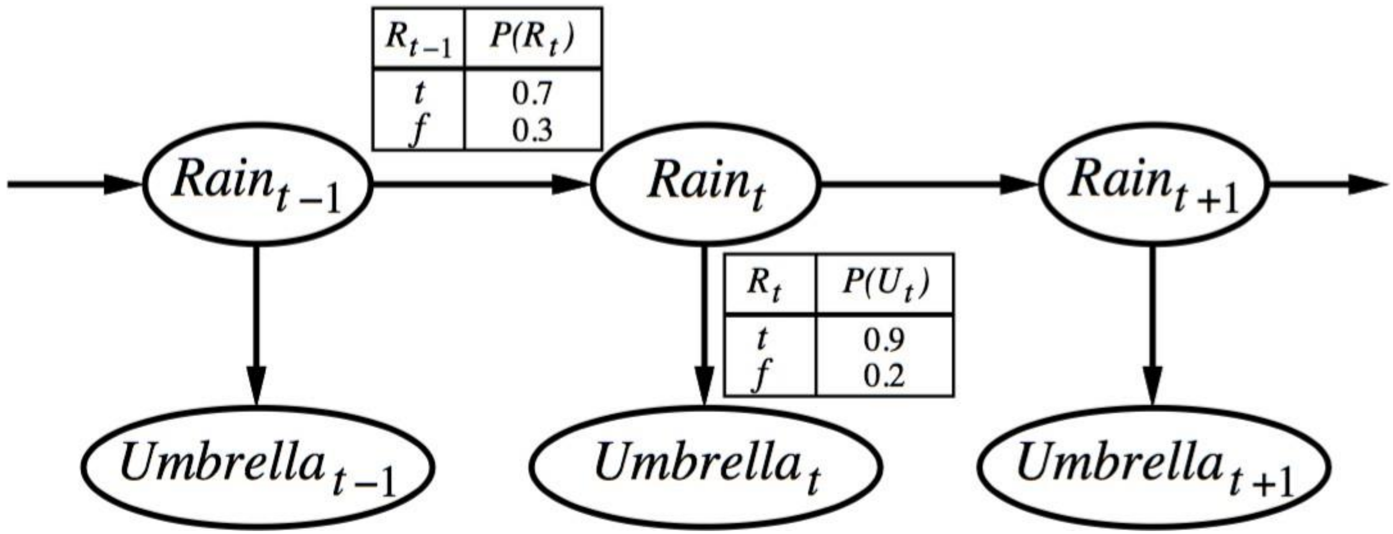


Umbrella sequence: [true, true, false, true, true]

What is the most likely weather sequence?

2^5 Sequences to examine

The rain problem



Umbrella sequence: [true, true, false, true, true]

What is the most likely weather sequence?

Use smoothing to find $\mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{1:t})$ for $0 \leq k < t$ and k=1,2,3,4,5 ?

why incorrect?

Viterbi Algorithm

Viterbi is a Dynamic Programming

- the probability of the best sequence reaching each state at time t, is the probability of best predecessors *multiply* the transition probability *multiply* observation probability

How Viterbi works

- Memorize the Viterbi probability, $\max P_j(x_1,..x_{t-1},X_t=s_j|e_1,..e_t)$ or j in [1,N], N is the number of all possible states (In the rain problem, N = 2, true and false).
- Initialization:

  Suppose the start state is s_0, which has equal probability to be $s_1, \ldots, s_N$.

Record which j leads to the maximum

	Time: T_0	Time: T_1 Observation: e_1
j=1, s_1	p(s_0) $v_1(0)$	max $p(X_1=s_1\|e_1) = p(e_1\|X_1=s_1) \max_{j \text{ in } \{1,\ldots,N\}} [p(X_1=s_1\|X_0=s_j)\, v_j(0)]$
j=2, s_2	p(s_0) $v_2(0)$	max $p(X_1=s_2\|e_1) = p(e_1\|X_1=s_2) \max_{j \text{ in } \{1,\ldots,N\}} [p(X_1=s_2\|X_0=s_j)\, v_j(0)]$
…		…
j=N, s_N	p(s_0) $v_N(0)$	max $p(X_1=s_N\|e_1) = p(e_1\|X_1=s_N) \max_{j \text{ in } \{1,\ldots,N\}} [p(X_1=s_N\|X_0=s_j)\, v_j(0)]$

transition probability

Observation probability

Viterbi Algorithm

Viterbi is a Dynamic Programming

- the probability of the best sequence reaching each state at time *t*, is the probability of best predecessors *multiply* the transition probability *multiply* observation probability

How Viterbi works

- Memorize the Viterbi probability, $\max P_j(x_1, ..x_{t-1}, X_t = s_j | e_1, ..e_t)$ or j in [1,*N*], *N* is the number of all possible states (In the rain problem, *N* = 2, true and false).
- T_2

Record which j leads to the maximum

	Time: T_1 Observation: e_1	T_2 , e_2
$j=1, s_1$	$p(e_1 \vert s_1) \max_{j \text{ in } \{1,\dots,N\}} [p(X_1=s_1 \vert X_0=s_j)\, v_j(0)]$ $v_1(1)$	$\max p(X_2=s_1, X_1 \vert e_1, e_2) = p(e_2 \vert s_1) \max_{j \text{ in } \{1,\dots,N\}}[P(X_2=s_1 \vert X_1=s_j)\, v_j(1)]$ $v_1(2)$
$j=2, s_2$	$p(e_1 \vert s_2) \max_{j \text{ in } \{1,\dots,N\}} [p(X_1=s_2 \vert X_0=s_j)\, v_j(0)]$ $v_2(1)$	$\max p(X_2=s_2, X_1 \vert e_1, e_2) = p(e_2 \vert s_2) \max_{j \text{ in } \{1,\dots,N\}}[P(X_2=s_2 \vert X_1=s_j)\, v_j(1)]$ $v_2(2)$
…	…	
$j=N, s_N$	$p(e_1 \vert s_N) \max_{j \text{ in } \{1,\dots,N\}} [p(X_1=s_N \vert X_0=s_j)\, v_j(0)]$ $v_N(1)$	$\max p(X_2=s_N, X_1 \vert e_1, e_2) = p(e_2 \vert s_N) \max_{j \text{ in } \{1,\dots,N\}}[P(X_2=s_N \vert X_1=s_j)\, v_j(1)]$ $v_N(2)$

Viterbi Algorithm

Viterbi is a Dynamic Programming

- the probability of the best sequence reaching each state at time *t*, is the probability of best predecessors *multiply* the transition probability *multiply* observation probability

How Viterbi works

- Memorize the Viterbi probability, $\max P_j(x_1,..x_{t-1},X_t=s_j|e_1,..e_t)$ or j in [1,*N*], *N* is the number of all possible states (In the rain problem, *N* = 2, true and false).
- T_t

Time: T_t , Observation at T_{t} : e_t — trellis

		T_1		
$j=1$, s_1		$v_1(2)$		$\max p(X_t=s_1, X_1,\ldots,X_{t-1} \mid e_1,\ldots,e_t) = p(e_t \mid s_1) \max[P(X_t=s_1 \mid X_{t-1}=s_j)\, v_j(t-1)]$
$j=2$, s_2	…	$v_2(2)$	…	$\max p(X_t=s_2, X_1,\ldots,X_{t-1} \mid e_1,\ldots,e_t) = p(e_t \mid s_2) \max[P(X_t=s_2 \mid X_{t-1}=s_j)\, v_j(t-1)]$
…				
$j=N$, s_N		$v_N(2)$		$\max p(X_t=s_N, X_1,\ldots,X_{t-1} \mid e_1,\ldots,e_t) = p(e_t \mid s_N) \max[P(X_t=s_N \mid X_{t-1}=s_j)\, v_j(t-1)]$

Viterbi Algorithm

Viterbi is a Dynamic Programming

- the probability of the best sequence reaching each state at time *t*, is the probability of best predecessors *multiply* the transition probability *multiply* observation probability

How Viterbi works

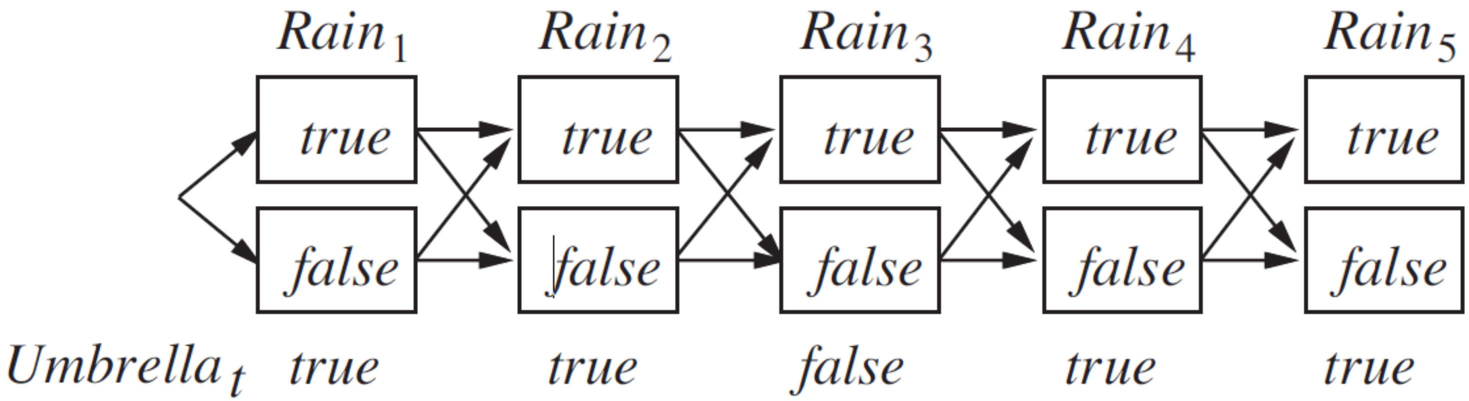
- Memorize the Viterbi probability, $\max P_j(x_1, ..x_{t-1}, X_t = s_j | e_1, ..e_t)$ or j in [1,*N*], *N* is the number of all possible states (In the rain problem, *N* = 2, true and false).
- Backtracing: go backwards to the recorded best predecessors, until the beginning.

		Time: T_2		*Time: T_t , Observation at T_{t} : e_t*
j=1, s_1		$v_1(2)$		$p(e_t \mid s_1) \max[P(X_t{=}s_1 \mid X_{t-1}{=}s_j)\, v_j(t\text{-}1)]$
j=2, s_2	…	$v_2(2)$	…	$p(e_t \mid s_2) \max[P(X_t{=}s_2 \mid X_{t-1}{=}s_j)\, v_j(t\text{-}1)]$
…				
j=*N*, s_N		$v_N(2)$		$p(e_t \mid s_N) \max[P(X_t{=}s_N \mid X_{t-1}{=}s_j)\, v_j(t\text{-}1)]$

max

trellis

Viterbi Algorithm

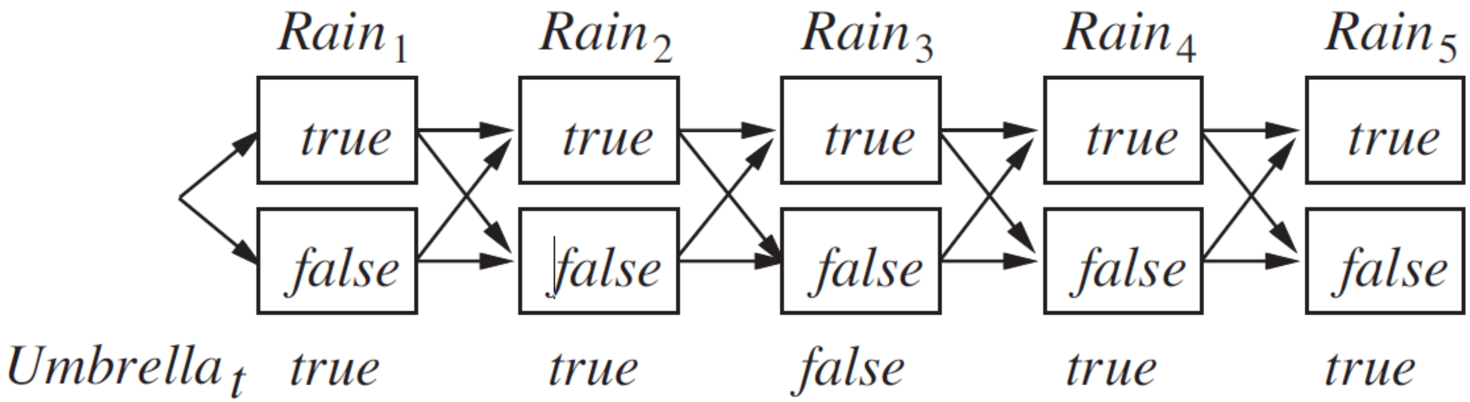


A state graph: each node is a possible state at each time step.

- Objective: finding the most likely path through this graph that generates the observation e.g., Umbrella sequence as [true, true, false, true, true].

$$\max_{\mathbf{x}_1 \ldots \mathbf{x}_t} \mathbf{P}(\mathbf{x}_1, \ldots, \mathbf{x}_t, \mathbf{X}_{t+1} \mid \mathbf{e}_{1:t+1})$$

Viterbi Algorithm



$$\max_{\mathbf{x}_1 \ldots \mathbf{x}_t} \mathbf{P}(\mathbf{x}_1, \ldots, \mathbf{x}_t, \mathbf{X}_{t+1} \mid \mathbf{e}_{1:t+1})$$

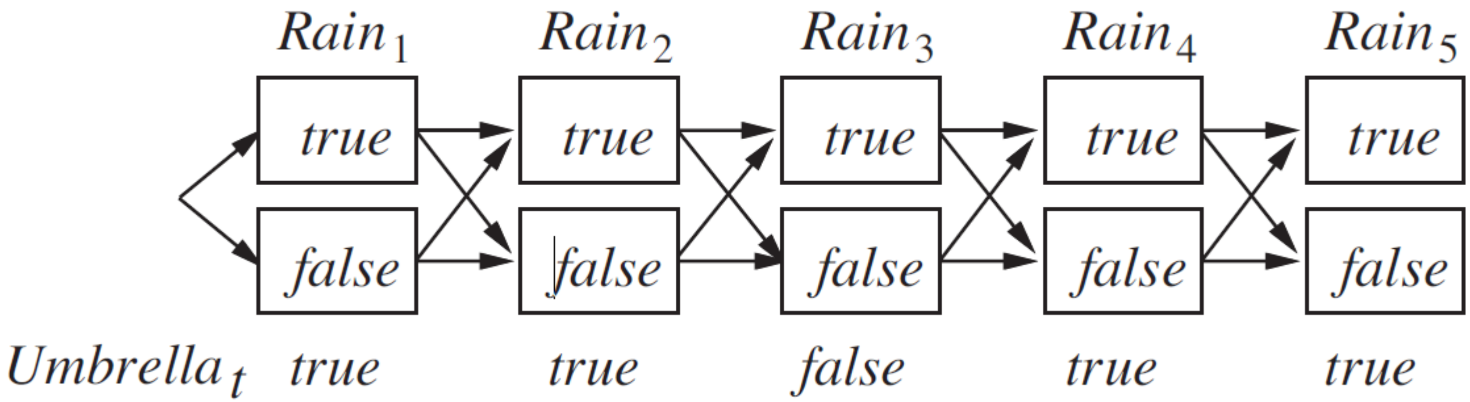
Recall Bayesian network's global semantics:

$$\mathbf{P}(\mathbf{X}_{0:t}, \mathbf{E}_{1:t}) = \mathbf{P}(\mathbf{X}_0) \prod_{i=1}^{t} \mathbf{P}(\mathbf{X}_i \mid \mathbf{X}_{i-1}) \mathbf{P}(\mathbf{E}_i \mid \mathbf{X}_i)$$

So we could find the relation between

$\mathbf{P}(\mathbf{x}_1, \ldots, \mathbf{x}_t, \mathbf{X}_{t+1} \mid \mathbf{e}_{1:t+1})$ and $P(\mathbf{x}_1, \ldots, \mathbf{x}_{t-1}, \mathbf{x}_t \mid \mathbf{e}_{1:t})$

Viterbi Algorithm



$$\mathbf{P}(\mathbf{x}_1, \ldots, \mathbf{x}_t, \mathbf{X}_{t+1} \mid \mathbf{e}_{1:t+1})$$

$$= \alpha \, \mathbf{P}(\mathbf{e}_{t+1} \mid \mathbf{X}_{t+1}) \, \mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{x}_t) P(\mathbf{x}_1, \ldots, \mathbf{x}_{t-1}, \mathbf{x}_t \mid \mathbf{e}_{1:t})$$

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

$$P(\underbrace{x_1,\dots x_t}_{A}, \underbrace{X_{t+1}}_{B} \,|\, e_{1:t+1}) \quad e_{1:t+1} \to \underbrace{e_{1:t}}_{C},\ \underbrace{e_{t+1}}_{D}$$

vs

$$\frac{P(e_{1:t})\, P(e_{t+1} \,|\, x_{t+1})\, P(X_{t+1} \,|\, x_t)\, P(x_1,\dots,x_{t-1},x_t \,|\, e_{1:t})}{P(e_{1:t+1})}$$

$$P(A,B \,|\, C,D) \quad \text{vs} \quad \frac{P(C)\cdot P(D|B)\cdot \underbrace{P(B|x_t)}_{\text{first order Markov}} P(A|C)}{P(C,D)}$$

$\Downarrow$ Bayes' rule

$$\frac{P(C,D|A,B)\cdot P(A,B)}{P(C,D)} \quad \text{vs} \quad \frac{P(C)\cdot P(D|B) P(B|A)\cdot P(A|C)}{P(C,D)}$$

$\Downarrow$

no dependence, no dependence

$$\frac{P(C|A,B)\cdot P(D|A,B)\cdot P(A,B)}{P(C,D)}$$

$\uparrow$ match

$\Downarrow$

$$\frac{P(C|A)\cdot P(D|B)\cdot P(A,B)}{P(C,D)}$$

local semantic

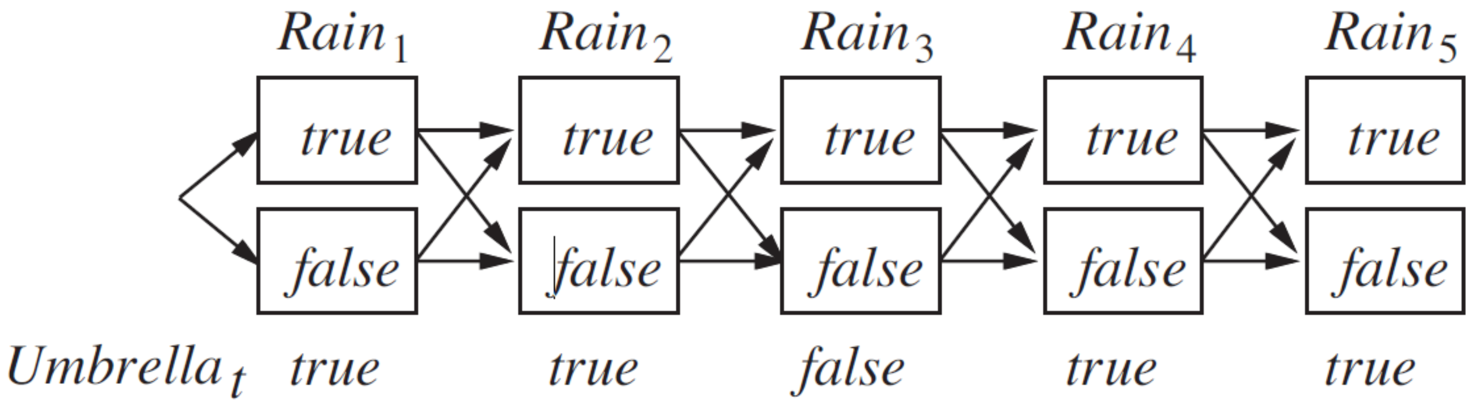
$\Downarrow$

$$\frac{\underbrace{P(C|A)}_{\text{Bayes' rule}}\cdot P(D|B)\cdot P(B|A)\cdot P(A)}{P(C,D)}$$

$\Downarrow$

$$\frac{\frac{P(A|C)\cdot P(C)}{\cancel{P(A)}}\cdot P(D|B) P(B|A) \cancel{P(A)}}{P(C,D)}$$

Viterbi Algorithm



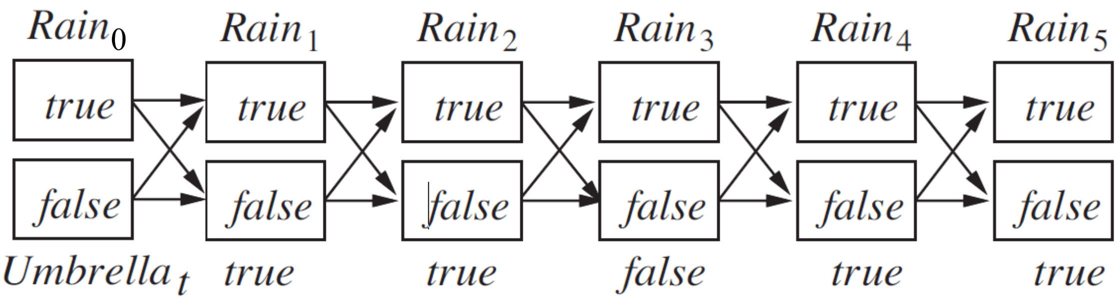
$$\max \mathbf{P}(\mathbf{x}_1, \ldots, \mathbf{x}_t, \mathbf{X}_{t+1} \mid \mathbf{e}_{1:t+1})$$
$$= \alpha \mathbf{P}(\mathbf{e}_{t+1} \mid \mathbf{X}_{t+1}) \max \left(\mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{x}_t) \, P(\mathbf{x}_1, \ldots, \mathbf{x}_{t-1}, \mathbf{x}_t \mid \mathbf{e}_{1:t}) \right)$$

As we always find the max, so the computation could ignore α

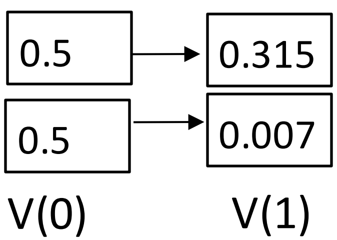
$$\alpha = P(\mathbf{e}_{1:t}) \, / \, P(\mathbf{e}_{1:t+1})$$

Viterbi Algorithm

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



$$
\mathbf{T} = \begin{array}{c} \\ R_{t-1} = t \\ R_{t-1} = f \end{array}
\begin{array}{c} \begin{array}{cc} R_t = t & R_t = f \end{array} \\ \begin{bmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{bmatrix} \end{array}
$$



$\max P(r1|u1)$

$\max P(\neg r_1|u1)$

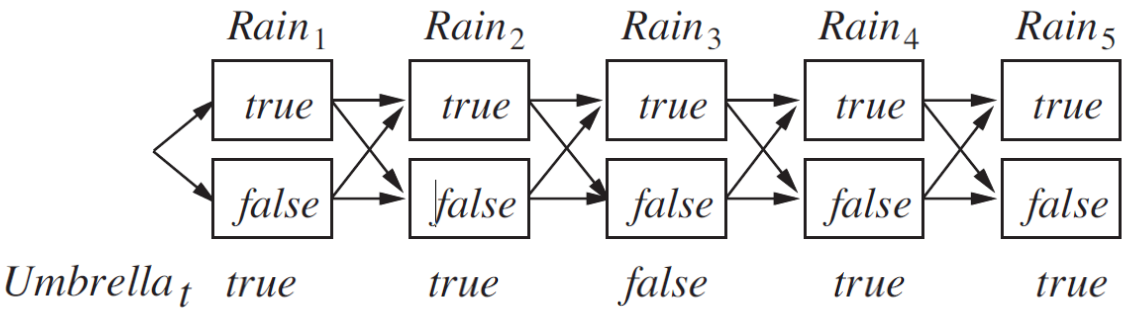
R_t	$P(U_t)$
t	0.9
f	0.2

$$
\begin{aligned}
\max P(r_1|u_1) &= P(u_1|r_1) \max P(r_1|\mathbf{R}_0)\mathbf{V}(\mathbf{0}) \\
&= P(u_1|r_1) \max\{P(r_1|r_0)P(r_0), P(r_1|\neg r_0)P(\neg r_0))\} \\
&= 0.9 \max\{0.7 * 0.5, 0.3 * 0.5\} \\
&= 0.9 * 0.7 * 0.5 = 0.315
\end{aligned}
$$

$$
\begin{aligned}
\max P(\neg r_1|u_1) &= P(u_1|\neg r_1) \max P(\neg r_1|\mathbf{R}_0)\mathbf{V}(\mathbf{0}) \\
&= P(u_1|\neg r_1) \max\{P(\neg r_1|r_0)P(r_0), P(\neg r_1|\neg r_0)P(\neg r_0))\} \\
&= 0.2 \max\{0.3 * 0.5, 0.7 * 0.5\} \\
&= 0.2 * 0.7 * 0.5 = 0.070
\end{aligned}
$$

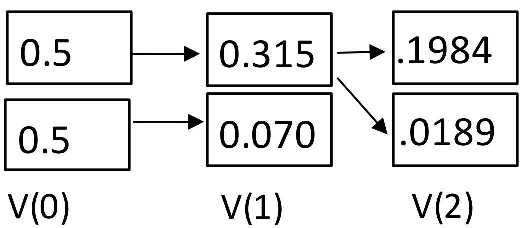
$$\max P(\mathbf{R}_1|u_1) = \mathbf{V}(1) =< 0.315, 0.007 >$$

Viterbi Algorithm



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

R_t	$P(U_t)$
t	0.9
f	0.2



$\max P(\mathbf{R}_1, r_2|u_1, u_2)$

$\max P(\mathbf{R}_1, \neg r_2|u_1, u_2)$

$$\begin{aligned}\max P(\mathbf{R}_1, r_2|u_1, u_2) &= P(u_2|r_2)\max(P(r_2|\mathbf{R}_1)\mathbf{V}(\mathbf{1}))\\ &= 0.9 * \max(< 0.7, 0.3 > * < 0.315, 0.007 >)\\ &= 0.9 * \max(0.7 * 0.315, 0.3 * 0.007)\\ &= 0.9 * 0.7 * 0.315 = 0.19845\end{aligned}$$

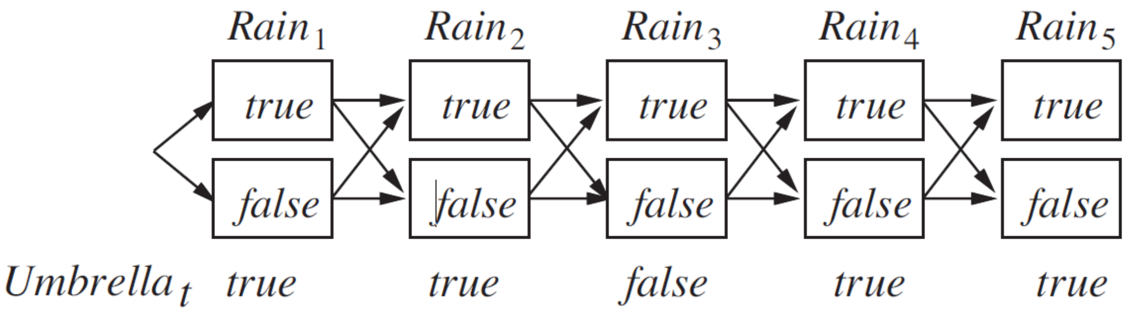
$P(r_1, r_2|u_1, u_2)$

$$\begin{aligned}\max P(\mathbf{R}_1, \neg r_2|u_1, u_2) &= P(u_2|\neg r_2)\max(P(\neg r_2|\mathbf{R}_1)\mathbf{V}(\mathbf{1}))\\ &= 0.2 * \max(< 0.3, 0.7 > * < 0.315, 0.007 >)\\ &= 0.2 * \max(0.3 * 0.315, 0.7 * 0.007)\\ &= 0.2 * 0.3 * 0.315 = 0.0189\end{aligned}$$

$P(r_1, \neg r_2|u_1, u_2)$

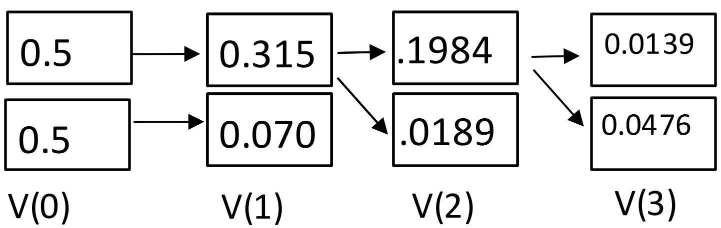
$$\max P(\mathbf{R}_1, \mathbf{R}_2|u_1, u_2) = \mathbf{V}(2) =< 0.19845, 0.0189 >$$

Viterbi Algorithm



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

R_t	$P(U_t)$	$P(U_f)$
t	0.9	0.1
f	0.2	0.8



$\max P(\mathbf{R}_1, \mathbf{R}_2, r_3 | u_1, u_2, \neg u_3)$

$\max P(\mathbf{R}_1, \mathbf{R}_2, \neg r_3 | u_1, u_2, \neg u_3)$

$$\begin{aligned}\max P(\mathbf{R}_1, \mathbf{R}_2, r_3 | u_1, u_2, \neg u_3) &= P(\neg u_3 | r_3) \max(P(r_3 | \mathbf{R}_2) \mathbf{V}(\mathbf{2})) \\ &= 0.1 * \max(< 0.7, 0.3 > * < 0.19845, 0.0189 >) \\ &= 0.1 * 0.7 * 0.19845 = 0.0138915\end{aligned}$$

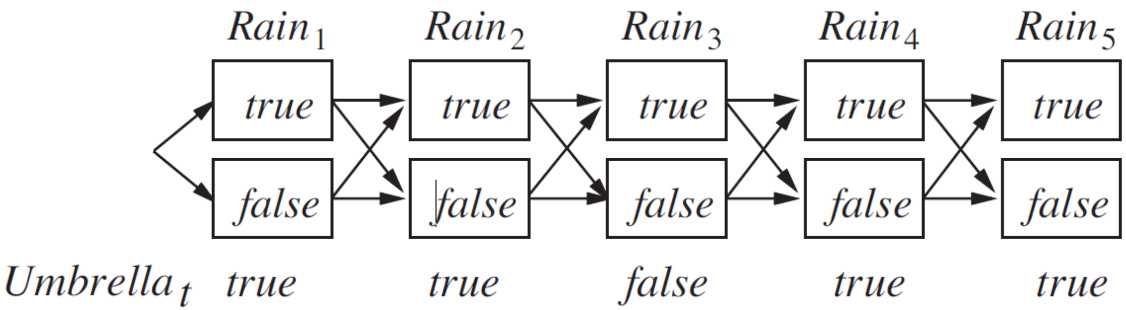
$P(r_1, r_2, r_3 | u_1, u_2, u_3)$

$$\begin{aligned}\max P(\mathbf{R}_1, \mathbf{R}_2, \neg r_3 | u_1, u_2, \neg u_3) &= P(\neg u_3 | \neg r_3) \max(P(\neg r_3 | \mathbf{R}_2) \mathbf{V}(\mathbf{2})) \\ &= 0.8 * \max(< 0.3, 0.7 > * < 0.19845, 0.0189 >) \\ &= 0.8 * 0.3 * 0.19845 = 0.047628\end{aligned}$$

$P(r_1, r_2, \neg r_3 | u_1, u_2, u_3)$

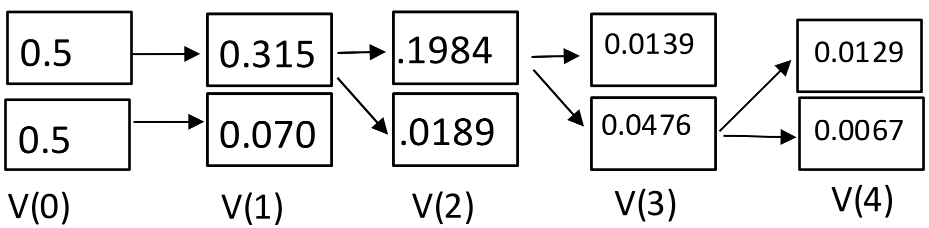
$$\max P(\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3 | u_1, u_2, \neg u_3) = \mathbf{V}(3) =< 0.0138915, 0.047628 >$$

Viterbi Algorithm



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

R_t	$P(U_t)$
t	0.9
f	0.2



$$\max P(\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3, r_4 | u_1, u_2, \neg u_3, u_4)$$

$$\max P(\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3, \neg r_4 | u_1, u_2, \neg u_3, u_4)$$

$$\begin{aligned} \max P(\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3, r_4 | u_1, u_2, \neg u_3, u_4) &= P(u_4 | r_4) \max(P(r_4 | \mathbf{R}_3) \mathbf{V}(\mathbf{3})) \\ &= 0.9 * \max(< 0.7, 0.3 > * < 0.0138915, 0.047628 >) \\ &= 0.9 * 0.3 * 0.047628 = 0.01285956 \end{aligned}$$

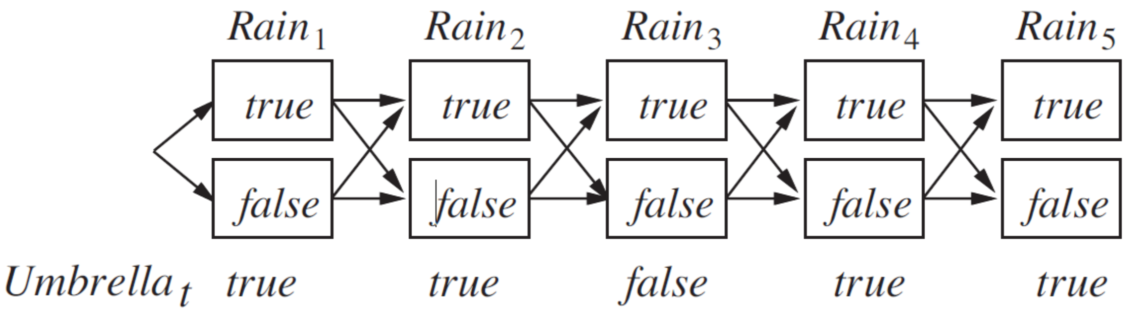
$$P(r_1, r_2, \neg r_3, r_4 | u_1, u_2, \neg u_3, u_4)$$

$$\begin{aligned} \max P(\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3, \neg r_4 | u_1, u_2, \neg u_3, u_4) &= P(u_4 | \neg r_4) \max(P(\neg r_4 | \mathbf{R}_3) \mathbf{V}(\mathbf{3})) \\ &= 0.2 * \max(< 0.3, 0.7 > * < 0.0138915, 0.047628 >) \\ &= 0.2 * 0.7 * 0.047628 = 0.00666792 \end{aligned}$$

$$P(r_1, r_2, \neg r_3, \neg r_4 | u_1, u_2, \neg u_3, \neg u_4)$$

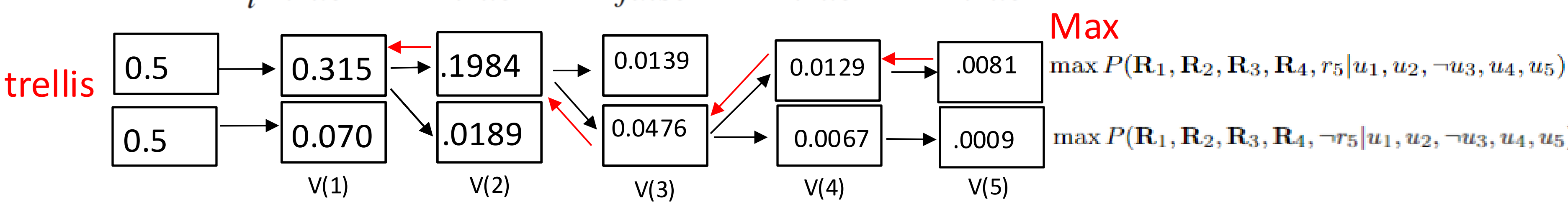
$$\max P(\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3, \mathbf{R}_4 | u_1, u_2, \neg u_3, u_4) = \mathbf{V}(\mathbf{4}) \;\; =< 0.01285956, 0.00666792 >$$

Viterbi Algorithm



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

R_t	$P(U_t)$
t	0.9
f	0.2



$$\max P(\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3, \mathbf{R}_4, r_5 | u_1, u_2, \neg u_3, u_4, u_5)$$
$$= P(u_5 | r_5) \max(P(r_5 | \mathbf{R}_4)\mathbf{V}(4))$$
$$= 0.9 * \max(< 0.7, 0.3 > * < 0.01285956, 0.00666792 >)$$
$$= 0.9 * 0.7 * 0.01285956 = 0.0081015228$$

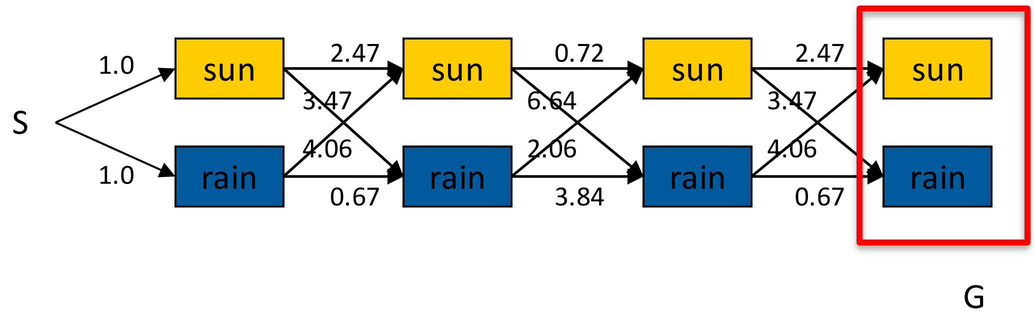
$$P(r_1, r_2, \neg r_3, r_4, r_5 | u_1, u_2, \neg u_3, u_4, u_5)$$

$$\max P(\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3, \mathbf{R}_4, \neg r_5 | u_1, u_2, \neg u_3, u_4, u_5)$$
$$= P(u_5 | \neg r_5) \max(P(\neg r_5 | \mathbf{R}_4)\mathbf{V}(4))$$
$$= 0.2 * \max(< 0.3, 0.7 > * < 0.01285956, 0.00666792 >)$$
$$= 0.2 * 0.7 * 0.00666792 = 0.0009335088$$

$$P(r_1, r_2, \neg r_3, \neg r_4, \neg r_5 | u_1, u_2, \neg u_3, u_4, u_5)$$

$$\max P(\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3, \mathbf{R}_4, \mathbf{R}_5 | u_1, u_2, \neg u_3, u_4, u_5) = \mathbf{V}(5) = < 0.0081015228, 0.0009335088 >$$

Viterbi in negative log space



W_{t-1}	$P(W_t \mid W_{t-1})$	
	sun	rain
sun	0.9	0.1
rain	0.3	0.7

W_t	$P(U_t \mid W_t)$	
	true	false
sun	0.2	0.8
rain	0.9	0.1

argmax of product of probabilities
= argmin of sum of negative log probabilities
= minimum-cost path

Viterbi is essentially breadth-first graph search
What about A*?