Probabilistic Reasoning Over Time 2: Viterbi

3007/7059 Artificial Intelligence

Wei Zhang

School of Computer Science
University of Adelaide

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.

Prediction

- Also 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.

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

Prediction

- One-step Prediction:

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

- 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

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

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

Smoothing

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

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

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

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

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

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

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

Filtering (Forward) **Backward**

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

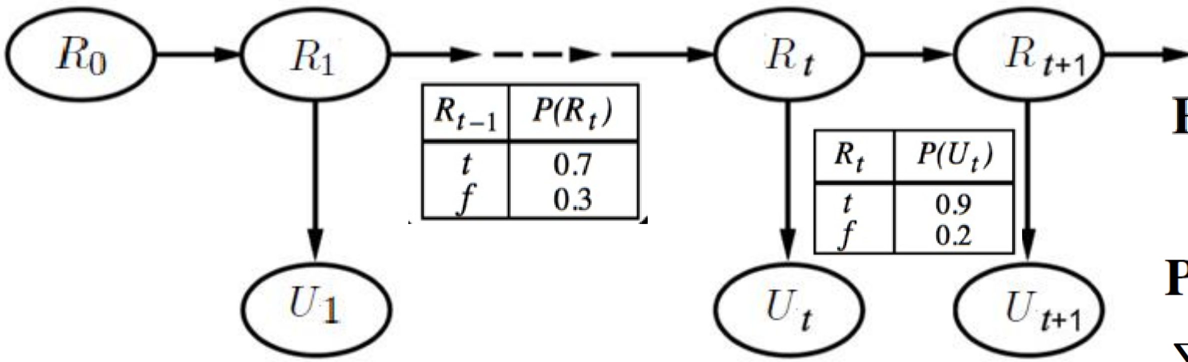
Initialize b:

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

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

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

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:

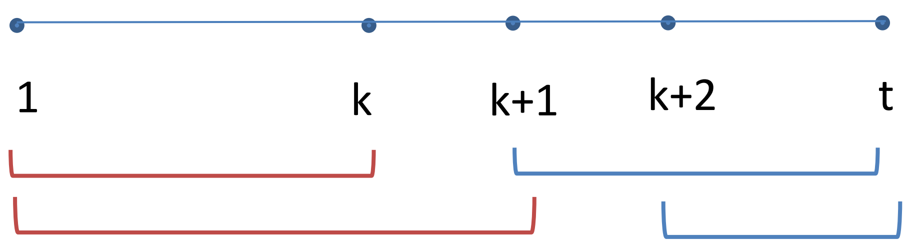
Was it raining outside at day 1, given the observation on day 1 and 2?

$$\mathbf{P}(R_1 \mid u_1, u_2) = \alpha\, \mathbf{P}(R_1 \mid u_1)\, \mathbf{P}(u_2 \mid R_1)$$
$$= \alpha\, \langle 0.818, 0.182 \rangle \sum_{r_2} P(u_2 \mid r_2) P(\ \mid r_2) \mathbf{P}(r_2 \mid R_1)$$
$$= \alpha\, \langle 0.818, 0.182 \rangle\, (0.9 \times 1 \times \langle 0.7, 0.3 \rangle + 0.2 \times 1 \times \langle 0.3, 0.7 \rangle)$$
$$= \alpha\, \langle 0.818, 0.182 \rangle\, \langle 0.69, 0.41 \rangle$$
$$\approx \langle 0.883, 0.117 \rangle$$

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 o smooth at a single time step with the observations $e_{1:t:}$ O(t), the whole sequence: O(t^2).

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

So far we learnt…

- Filtering

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

- Prediction

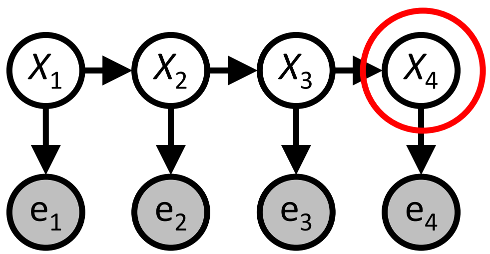
$$\mathbf{P}(\mathbf{X}_{t+k+1} \,|\, \mathbf{e}_{1:t})$$

- Smoothing

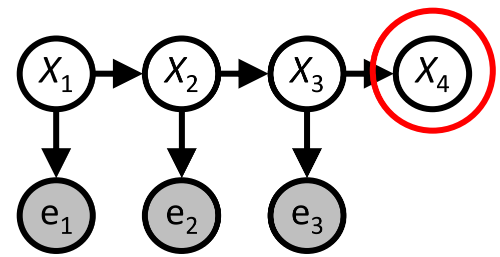
$$\mathbf{P}(\mathbf{X}_k \,|\, \mathbf{e}_{1:t}) \qquad 0 \le k < t$$

Inference tasks

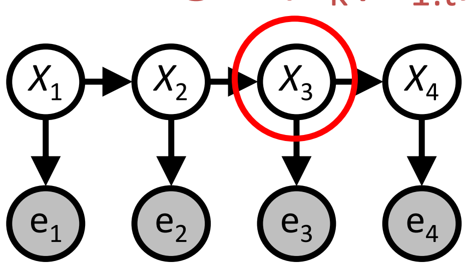
Filtering: $P(X_t | e1:t)$



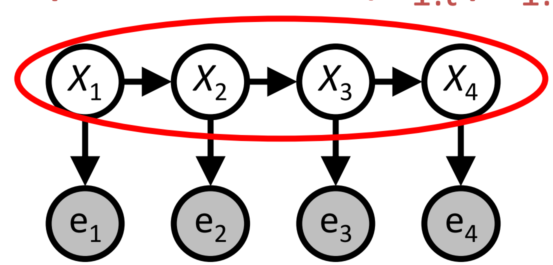
Prediction: $P(X_{t+k} | e_{1:t})$



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



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



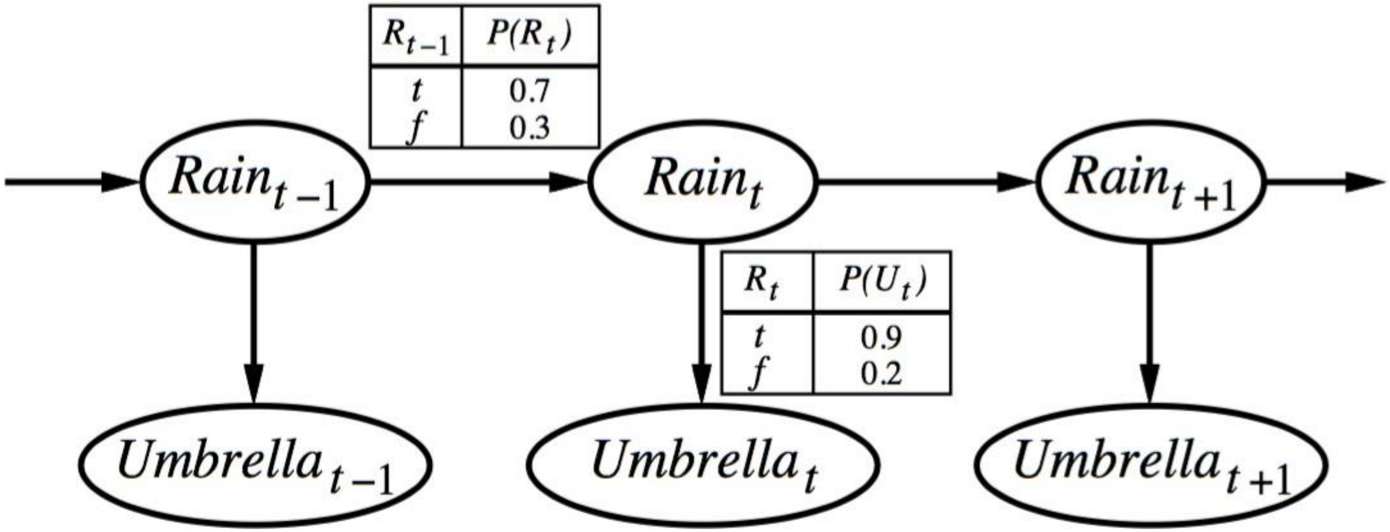
Viterbi Algorithm

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

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

  - Some applications
    - Speech recognition
    - Sequence tagging
    - …

The rain problem

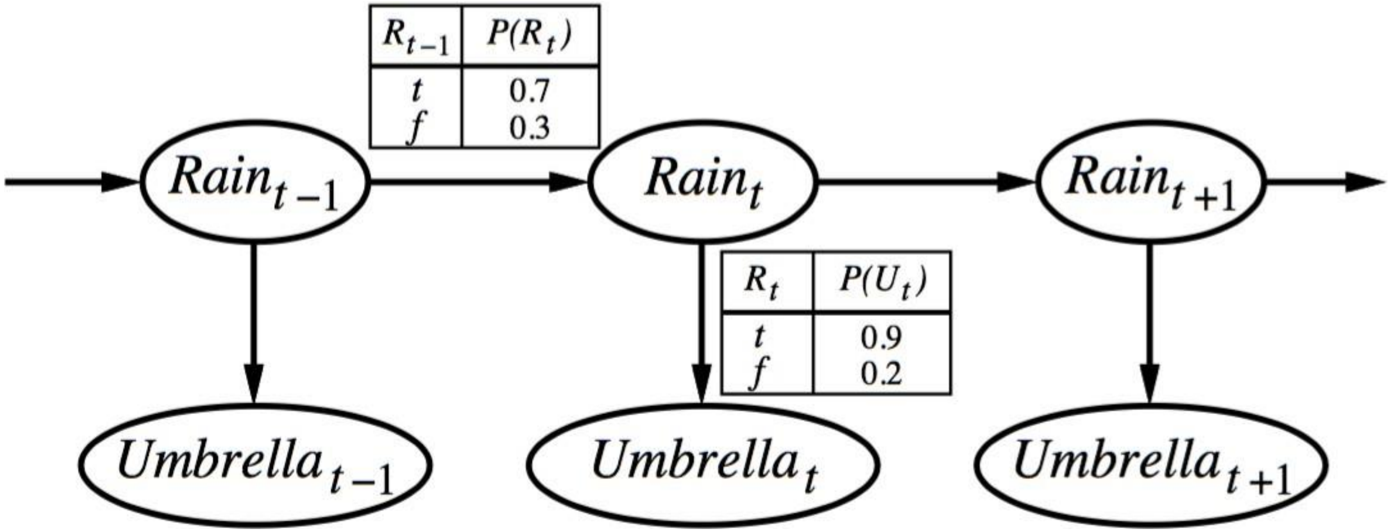


Umbrella sequence: [true, true, false, true, true]

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 x the transition probability x observation probability

How Viterbi works

- Memorize the Viterbi probability, $\max P_j(x_1, ..x_{t-1}, X_t = s_j | e_1, ..e_t)$ for 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*
$j=1, s_1$	$p(s_0)$ $v_1(0)$	$\max p(X_1=s_1 \mid e_1) = p(e_1 \mid X_1=s_1) \max_{j \text{ in } \{1,\ldots,N\}} [p(X_1=s_1 \mid X_0=s_j)\, v_j(0)]$
$j=2, s_2$	$p(s_0)$ $v_2(0)$	$\max p(X_1=s_2 \mid e_1) = p(e_1 \mid X_1=s_2) \max_{j \text{ in } \{1,\ldots,N\}} [p(X_1=s_2 \mid X_0=s_j)\, v_j(0)]$
…		…
$j=N, s_N$	$p(s_0)$ $v_N(0)$	$\max p(X_1=s_N \mid e_1) = p(e_1 \mid X_1=s_N) \max_{j \text{ in } \{1,\ldots,N\}} [p(X_1=s_N \mid X_0=s_j)\, v_j(0)]$

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 x the transition probability x observation probability

How Viterbi works

- Memorize the Viterbi probability, $\max P_j(x_1, ..x_{t-1}, X_t = s_j | e_1, ..e_t)$ for 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
$j=1, s_1$	$p(e_1\|s_1) \max_{j \text{ in } \{1,\ldots,N\}} [p(X_1=s_1\|X_0=s_j)\, v_j(0)]$ $v_1(1)$	$\max p(X_2=s_1, X_1\|e_1, e_2) = p(e_2\|s_1) \max_{j \text{ in } \{1,\ldots,N\}}[P(X_2=s_1\|X_1=s_j)\, v_j(1)]$ $v_1(2)$
$j=2, s_2$	$p(e_1\|s_2) \max_{j \text{ in } \{1,\ldots,N\}} [p(X_1=s_2\|X_0=s_j)\, v_j(0)]$ $v_2(1)$	$\max p(X_2=s_2, X_1\|e_1, e_2) = p(e_2\|s_2) \max_{j \text{ in } \{1,\ldots,N\}}[P(X_2=s_2\|X_1=s_j)\, v_j(1)]$ $v_2(2)$
…	…	
$j=N, s_N$	$p(e_1\|s_N) \max_{j \text{ in } \{1,\ldots,N\}} [p(X_1=s_N\|X_0=s_j)\, v_j(0)]$ $v_N(1)$	$\max p(X_2=s_N, X_1\|e_1, e_2) = p(e_2\|s_N) \max_{j \text{ in } \{1,\ldots,N\}}[P(X_2=s_N\|X_1=s_j)\, v_j(1)]$ $v_N(2)$

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 x the transition probability x observation probability

How Viterbi works

- Memorize the Viterbi probability, $\max P_j(x_1, ..x_{t-1}, X_t = s_j | e_1, ..e_t)$ for 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 $p(X_t=s_1, X_1,...,X_{t-1} \mid e_1, ..., e_t) = p(e_t \mid s_1) \max[P(X_t=s_1 \mid X_{t-1}=s_j)\, v_j(t-1)]$
j=2, s_2	...	$v_2(2)$	...	max $p(X_t=s_2, X_1,...,X_{t-1} \mid e_1, ..., e_t) = p(e_t \mid s_2) \max[P(X_t=s_2 \mid X_{t-1}=s_j)\, v_j(t-1)]$
...				
$j=N$, s_N		$v_N(2)$		max $p(X_t=s_N, X_1,...,X_{t-1} \mid e_1, ..., e_t) = p(e_t \mid s_N) \max[P(X_t=s_N \mid X_{t-1}=s_j)\, v_j(t-1)]$

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 x the transition probability x observation probability

How Viterbi works

- Memorize the Viterbi probability, $\max P_j(x_1, ..x_{t-1}, X_t = s_j | e_1, ..e_t)$ for 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.

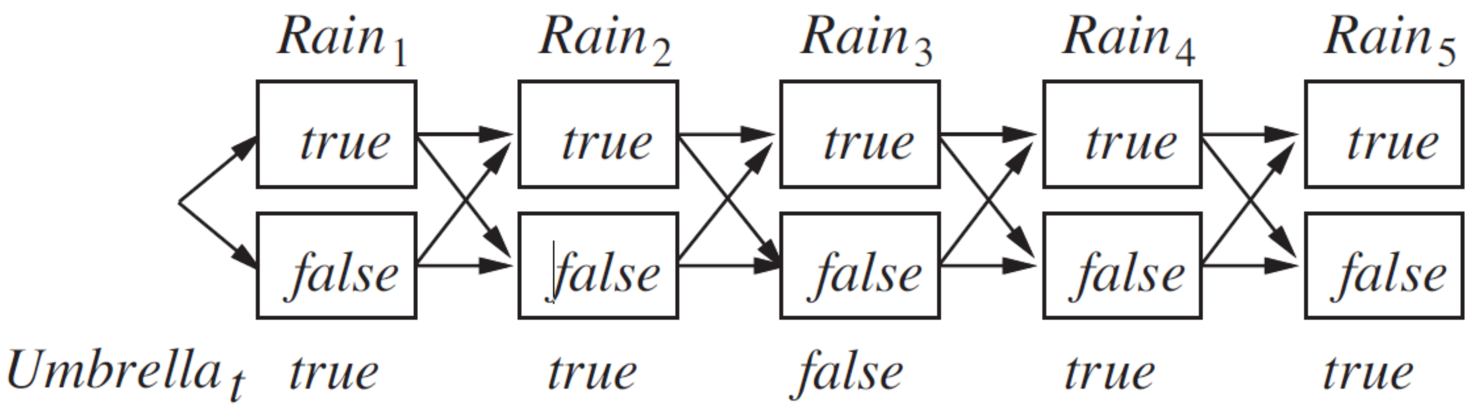
Time: T_2 *Time:* T_t , *Observation at* T_t *:* e_t

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

max

trellis

Viterbi Algorithm

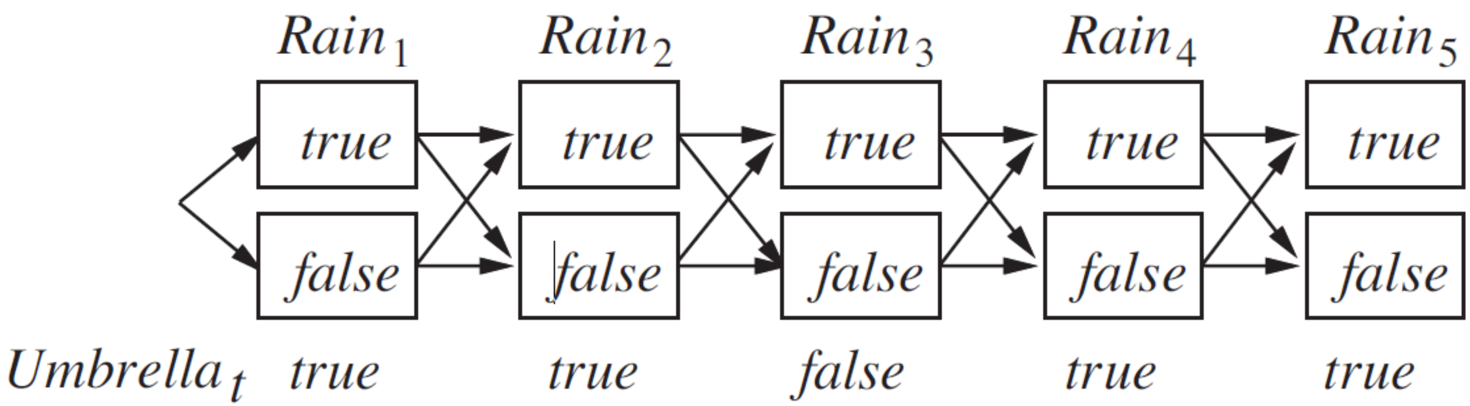


A state graph: each node is a possible state at each time step.

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



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

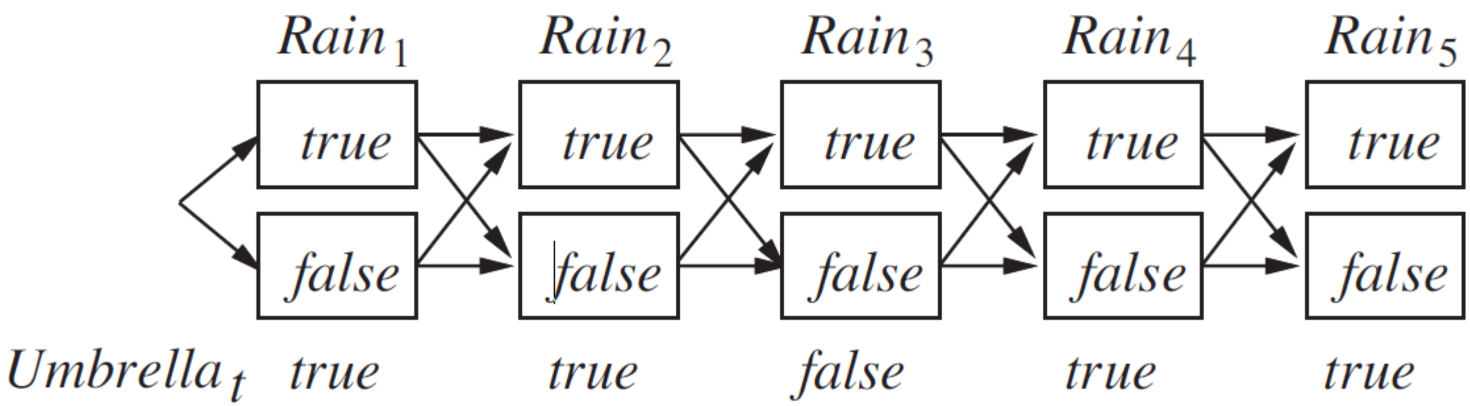
Recall Bayesian network's global semantics:

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

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

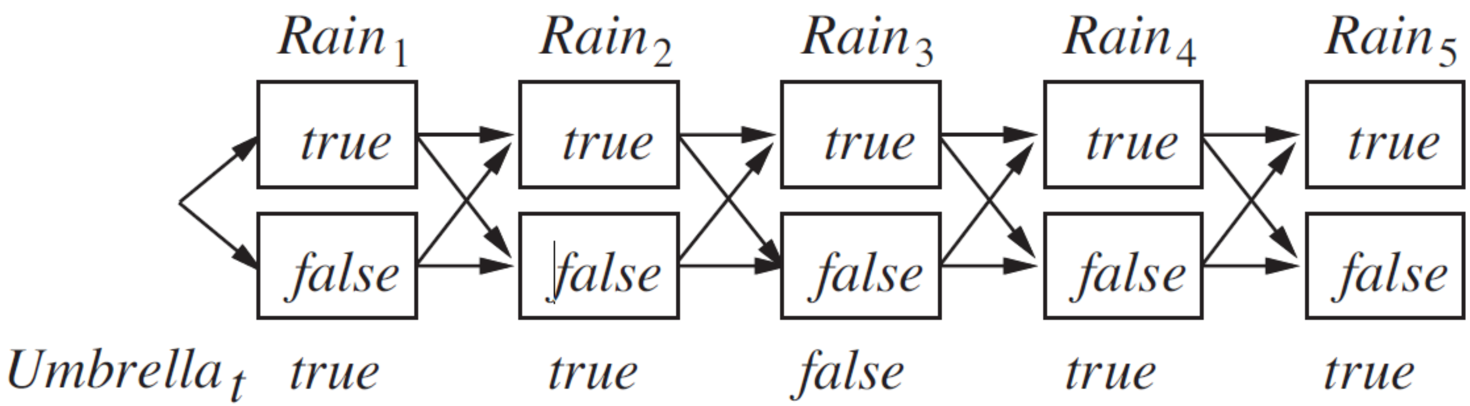


$$\mathbf{P}(\mathbf{x}_1, \ldots, \mathbf{x}_t, \mathbf{X}_{t+1} \mid \mathbf{e}_{1:t+1})$$

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

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

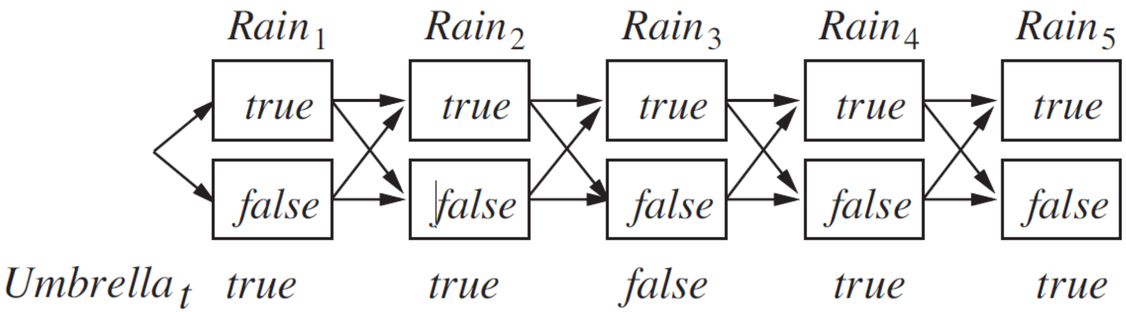
Viterbi Algorithm



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

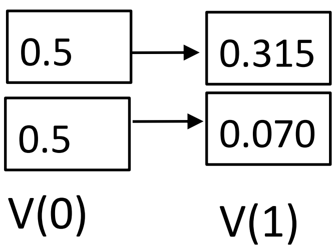
As we always find the max. so the computation could ignore alpha

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



$\max P(r1|u1)$

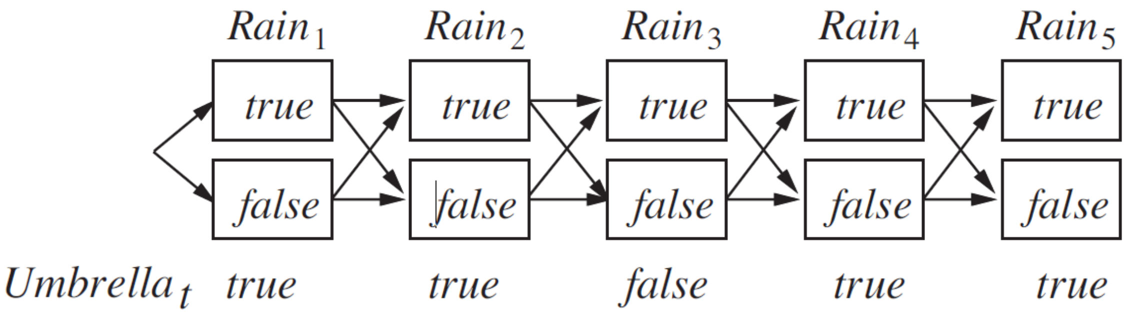
$\max P(\neg r_1|u1)$

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

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

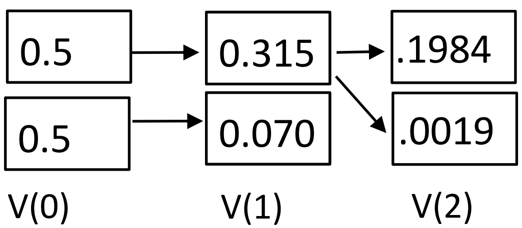
$$\max P(\mathbf{R}_1|u_1) = \mathbf{V}(1) = <0.315, 0.007>$$

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



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

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

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

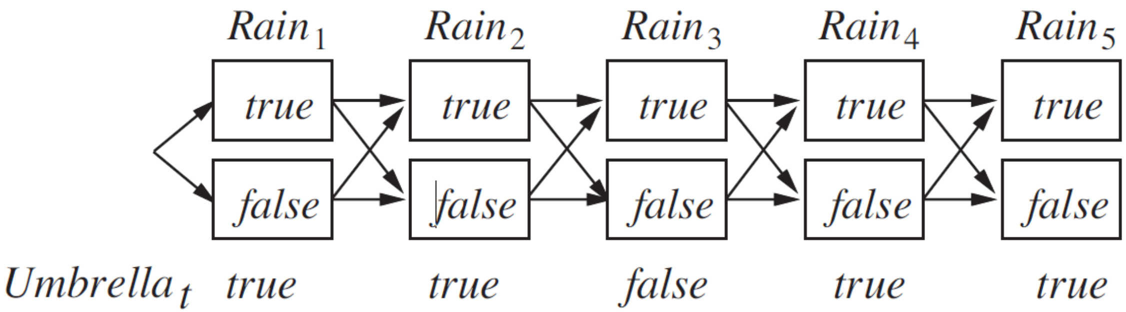
$P(r_1, r_2 | u_1, u_2)$

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

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

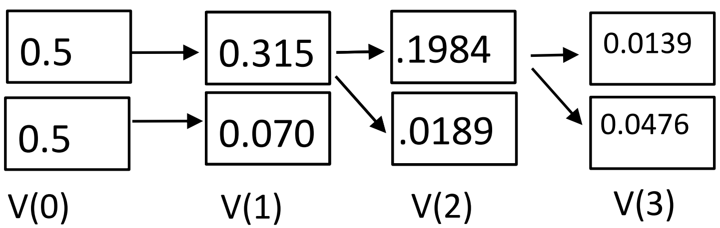
$$\max P(\mathbf{R}_1, \mathbf{R}_2 | u_1, u_2) = \mathbf{V}(2) = < 0.19845, 0.0189 >$$

Viterbi Algorithm



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

R_t	$P(U_t)$
t	0.9
f	0.2



$\max P(\mathbf{R_1}, \mathbf{R_2}, r_3 | u_1, u_2, \neg u_3)$

$\max P(\mathbf{R_1}, \mathbf{R_2}, \neg r_3 | u_1, u_2, \neg u_3)$

$$\begin{aligned}\max P(\mathbf{R_1}, \mathbf{R_2}, r_3 | u_1, u_2, \neg u_3) &= P(\neg u_3 | r_3) \max(P(r_3 | \mathbf{R_2}) \mathbf{V(2)}) \\ &= 0.1 * \max(< 0.7, 0.3 > * < 0.19845, 0.0189 >) \\ &= 0.1 * 0.7 * 0.19845 = 0.0138915\end{aligned}$$

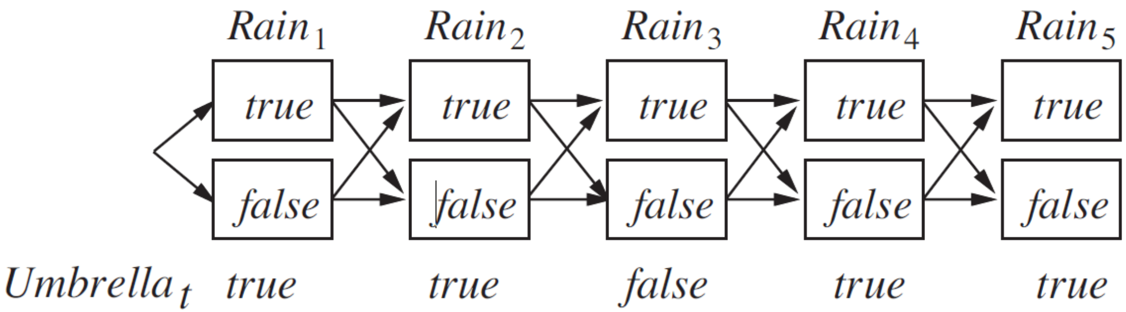
$P(r_1, r_2, r_3 | u_1, u_2, u_3)$

$$\begin{aligned}\max P(\mathbf{R_1}, \mathbf{R_2}, \neg r_3 | u_1, u_2, \neg u_3) &= P(\neg u_3 | \neg r_3) \max(P(\neg r_3 | \mathbf{R_2}) \mathbf{V(2)}) \\ &= 0.8 * \max(< 0.3, 0.7 > * < 0.19845, 0.0189 >) \\ &= 0.8 * 0.3 * 0.19845 = 0.047628\end{aligned}$$

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

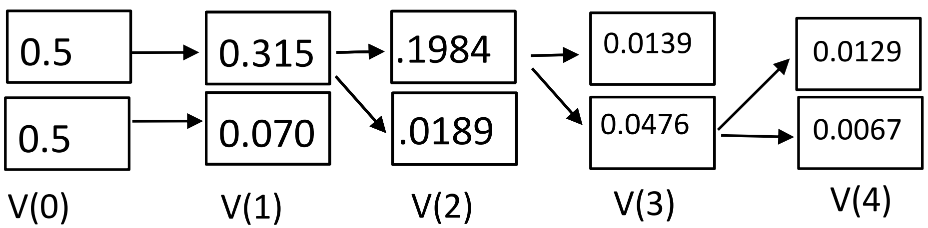
$$\max P(\mathbf{R_1}, \mathbf{R_2}, \mathbf{R_3} | u_1, u_2, \neg u_3) = \mathbf{V}(3) =< 0.0138915, 0.047628 >$$

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



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

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

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

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

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

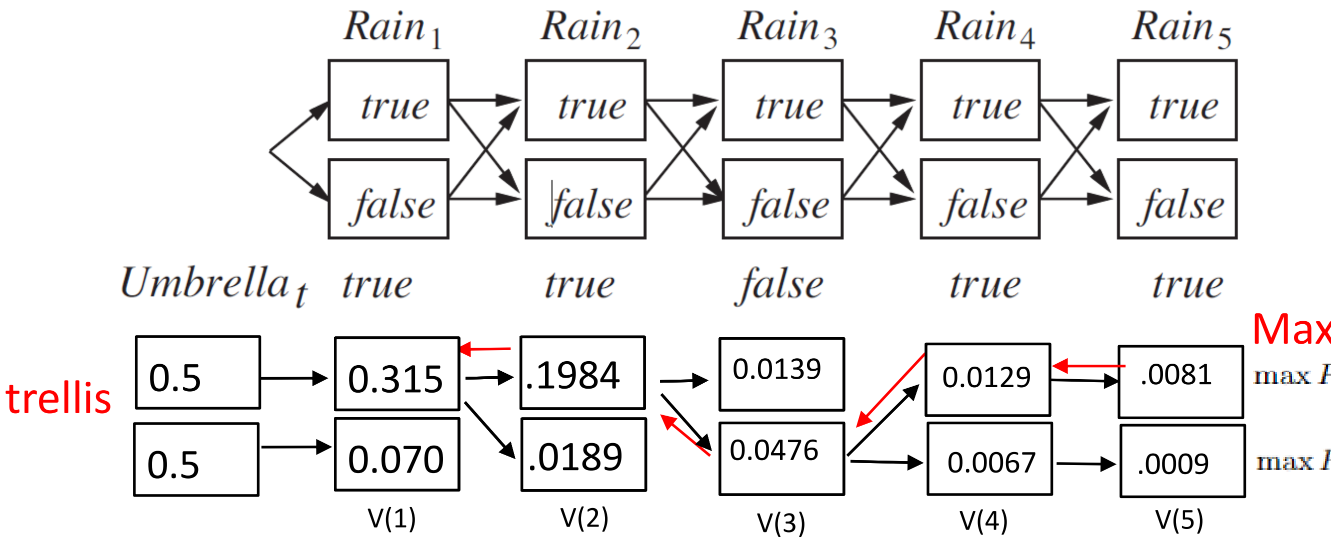
$P(r_1, r_2, \neg r_3, \neg r_4 | u_1, u_2, \neg u_3, \neg u_4)$

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

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



$\max P(\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3, \mathbf{R}_4, r_5 | u_1, u_2, \neg u_3, u_4, u_5)$

$\max P(\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3, \mathbf{R}_4, \neg r_5 | u_1, u_2, \neg u_3, u_4, u_5)$

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

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

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

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

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