CS440/ECE448 Lecture 23: HMM Inference and the Viterbi Algorithm

Mark Hasegawa-Johnson, 3/2022

Louis-Leopold Boilly, Passer Payez, 1803. Public domain work of art, https://en.wikipedia.org/wiki/Umbrella

Outline

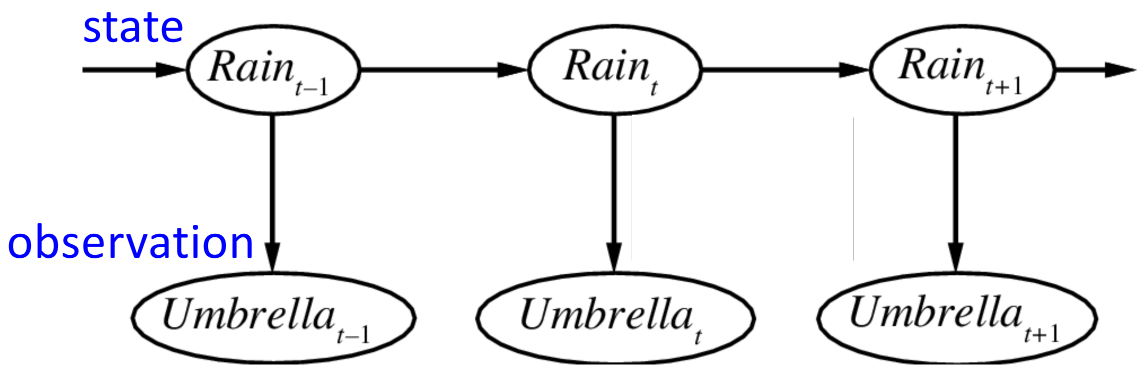
- Belief propagation
  - What is $P(Y_t|X_1 = x_1, \dots, X_T = x_T)$?
- Viterbi Algorithm
  - What is the most probable sequence $\{Y_1, \dots, Y_T\}$ given observations $\{X_1 = x_1, \dots, X_T = x_T\}$?

Example Scenario: UmbrellaWorld

Characters from the novel *Hammered* by Elizabeth Bear,
Scenario from chapter 15 of Russell & Norvig

Since he has read a lot about rain, Richard proposes a hidden Markov model:

- Rain on day t-1 (R_{t-1}) makes rain on day t (R_t) more likely.
- Elspeth usually brings her umbrella (U_t) on days when it rains (R_t), but not always.



Example Scenario: UmbrellaWorld

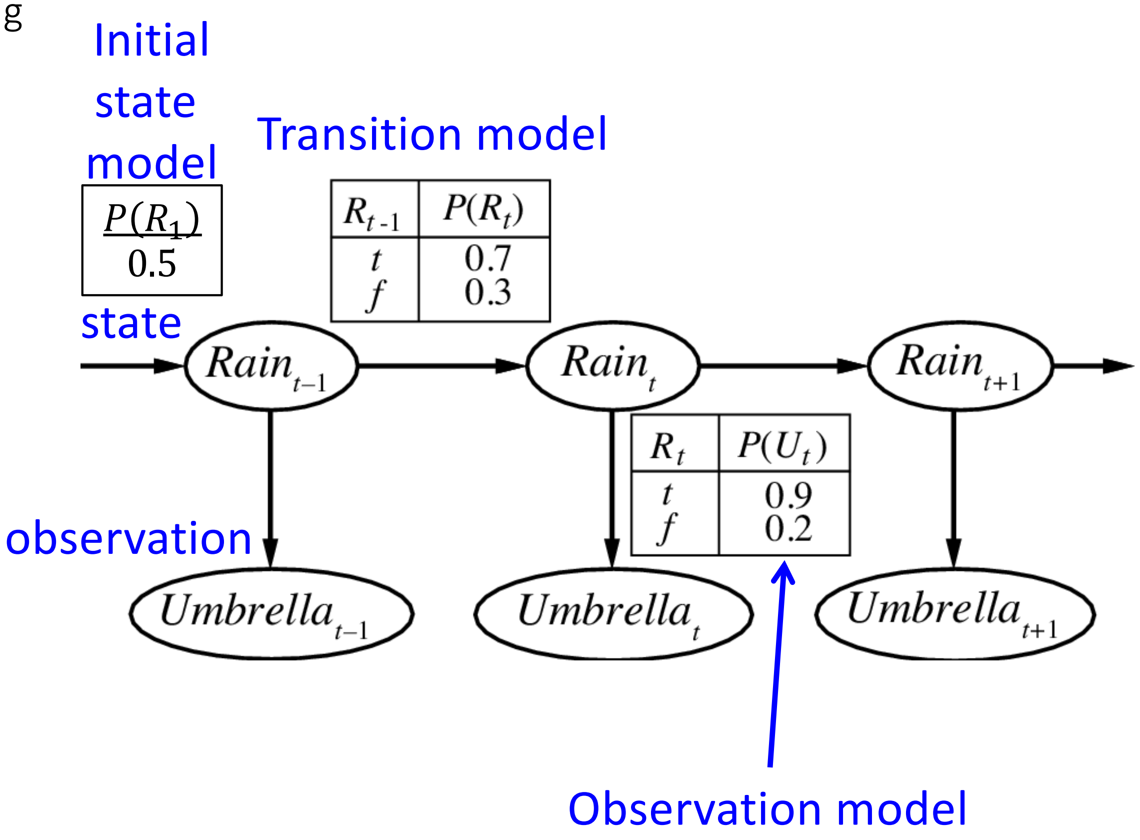
Characters from the novel *Hammered* by Elizabeth Bear,
Scenario from chapter 15 of Russell & Norvig

- Richard has no idea whether or not it's raining on day 1, so he assumes $P(R_1 = T) = 0.5$.
- Richard learns that the weather changes on 3 out of 10 days, thus

$$P(R_t = T|R_{t-1} = T) = 0.7$$
$$P(R_t = T|R_{t-1} = F) = 0.3$$

- He also learns that Elspeth sometimes forgets her umbrella when it's raining, and that she sometimes brings an umbrella when it's not raining. Specifically,

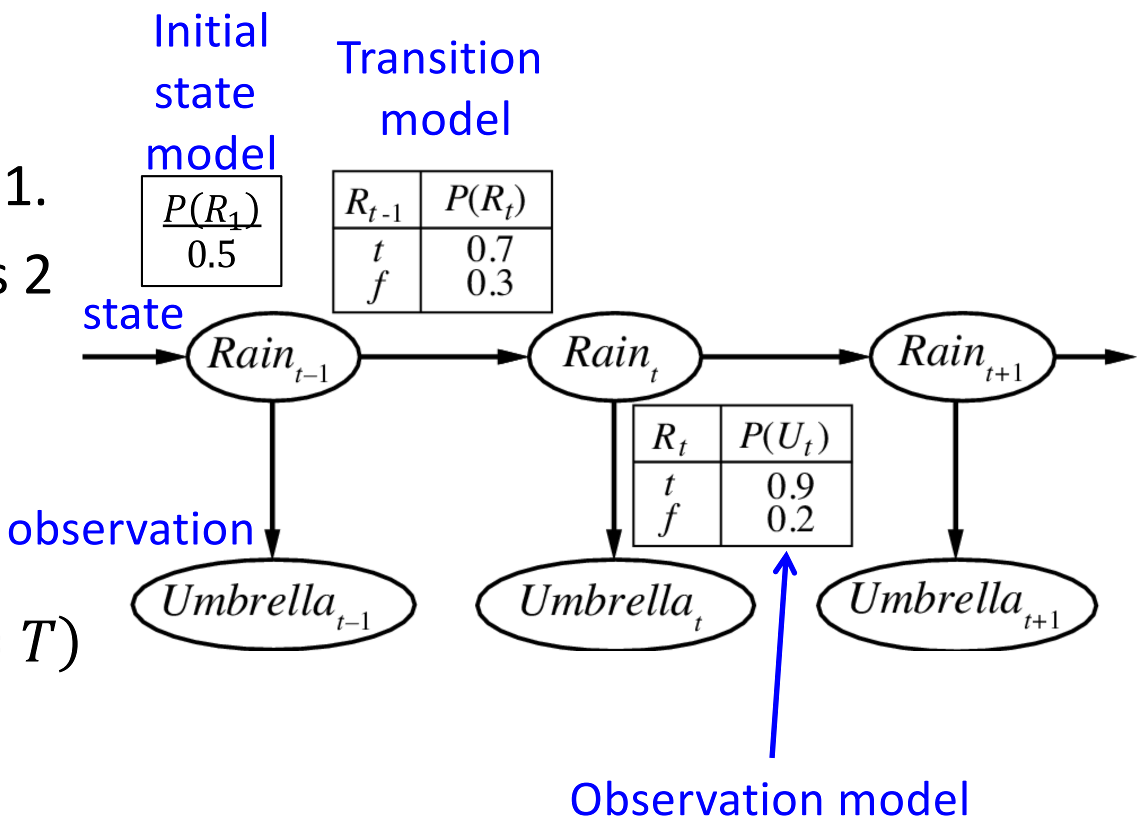
$$P(U_t = T|R_t = T) = 0.9$$
$$P(U_t = T|R_t = F) = 0.2$$



Belief propagation in an HMM: Example

- Elspeth has no umbrella on day 1.
- Elspeth has an umbrella on days 2 and 3.
- What is the probability that it's raining on day 3?

$$P(R_3 = T | U_1 = F, U_2 = T, U_3 = T)$$



$P(R_1)$
0.5

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

R_t	$P(U_t)$
t	0.9
f	0.2

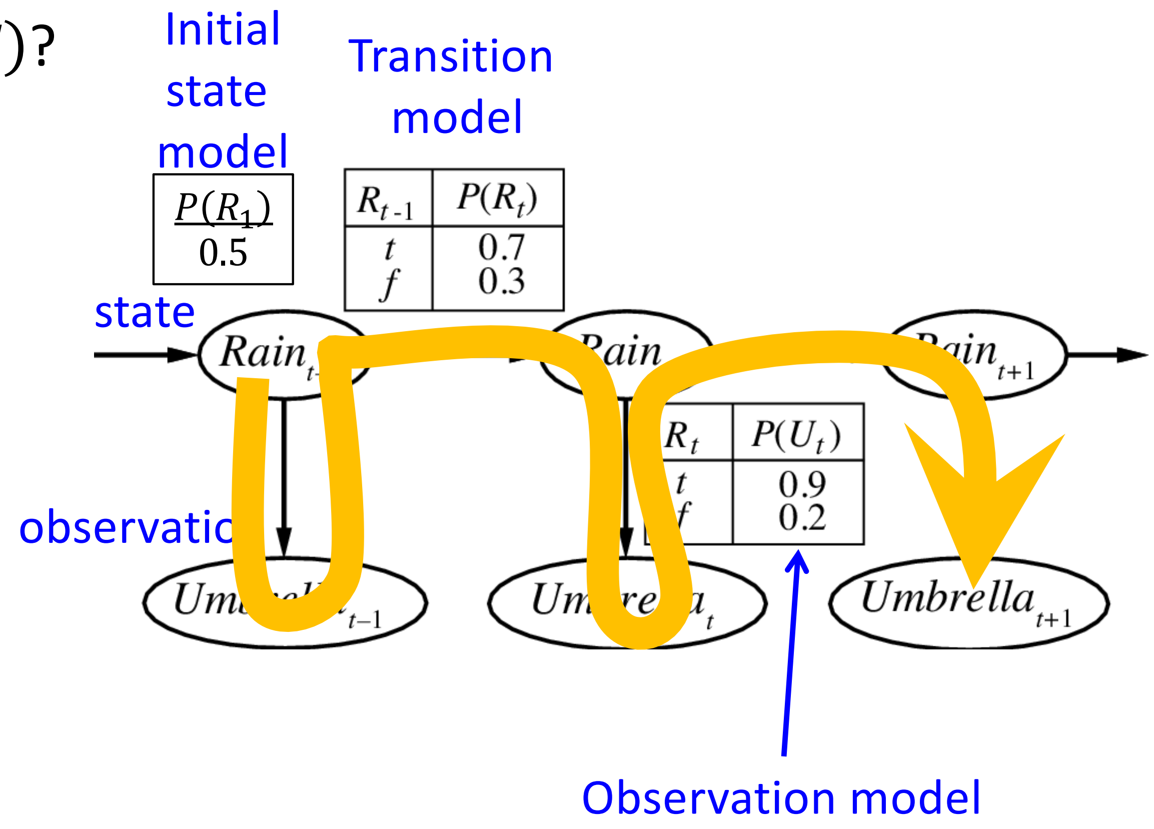
Belief propagation, step by step

1. Identify a path through the Bayesian network that includes all variables, including the query variable and all observed variables, starting at their common ancestor
2. Calculate the joint probability of the query variable and all observed variables, iteratively marginalizing out all intermediate variables step-by-step along the path.
3. Apply Bayes’ rule to get the desired conditional probability

Step 1: Identify a path starting at their common ancestor

$P(R_3 = T | U_1 = F, U_2 = T, U_3 = T)$?

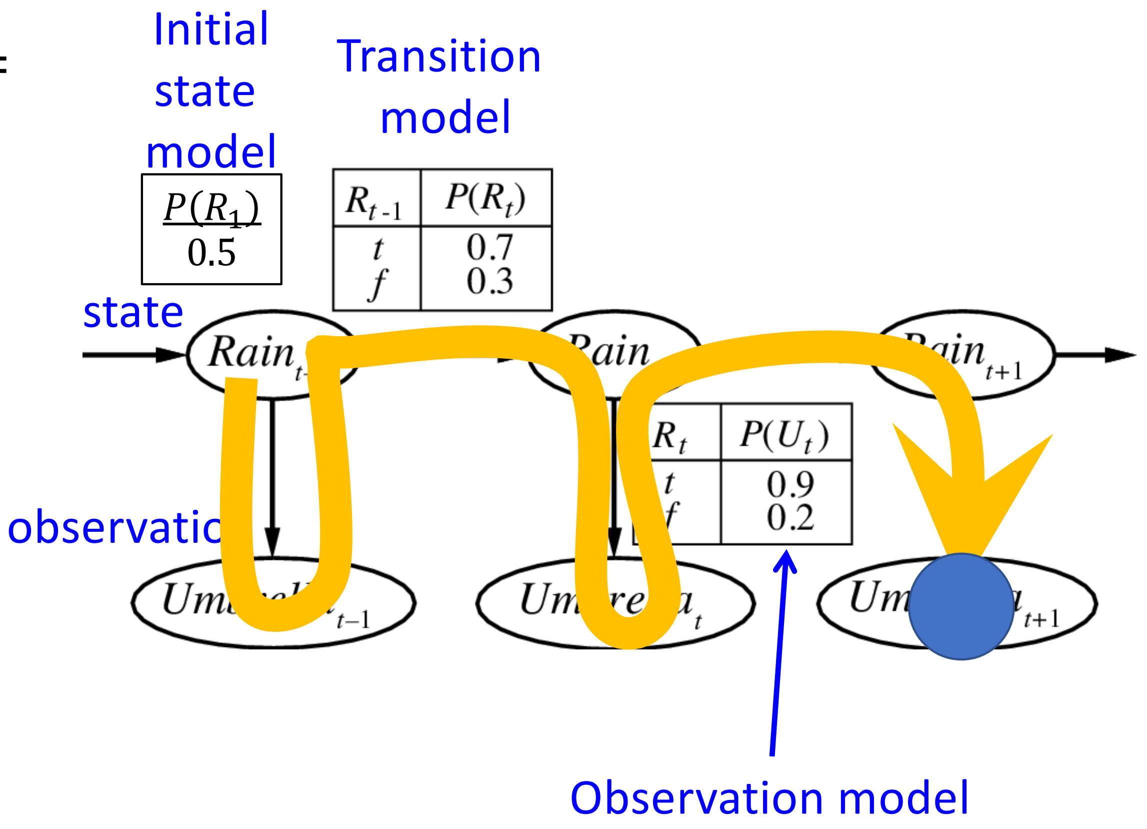
- Query variable: R_3
- Observed variables:
  - $U_1 = F$
  - $U_2 = T$
  - $U_3 = T$



Step 2: Calculate the joint probability, step-by-step...

$$P(U_1 = F, U_2 = T, U_3 = T, R_3) =$$
$$P(U_1 = F, U_2 = T, R_3)\times$$
$$P(U_3 = T | R_3) =$$

$$\begin{cases} (0.11535)(0.9) & R_3 = T \\ (0.08315)(0.2) & R_3 = F \end{cases}$$

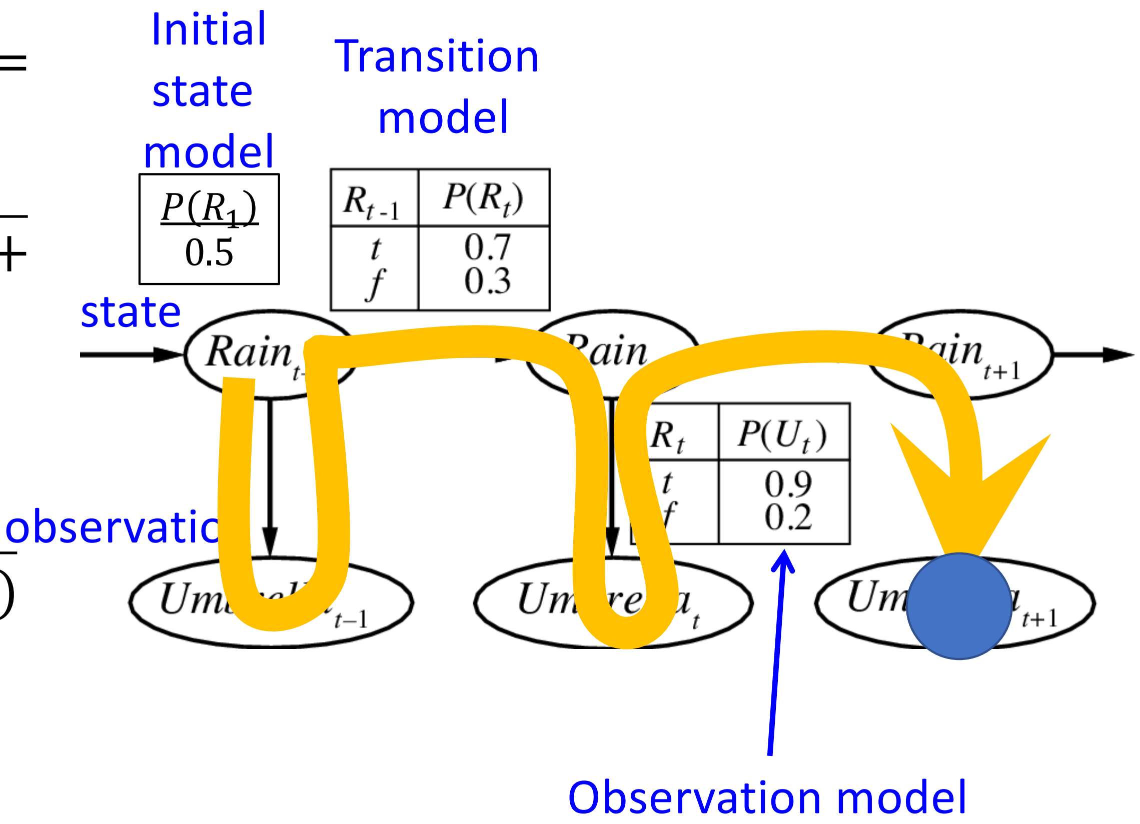


Step 3: Apply Bayes' rule to get conditional probability

$$P(R_3 = T | U_1 = F, U_2 = T, U_3 = T) =$$

$$\frac{P(R_3 = T, U_1 = F, U_2 = T, U_3 = T)}{P(R_3 = T, U_1 = F, U_2 = T, U_3 = T) + P(R_3 = F, U_1 = F, U_2 = T, U_3 = T)}$$

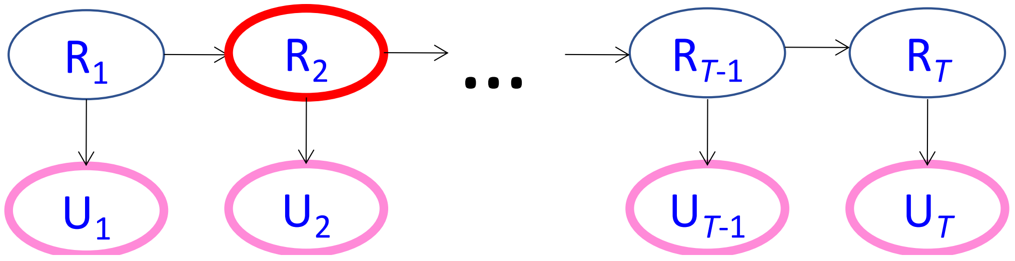
$$= \frac{(0.11535)(0.9)}{(0.11535)(0.9) + (0.08315)(0.2)}$$



Inference by Enumeration

To calculate a probability $P(R_3|U_1,U_2,U_3)$:

1. **Select:** which variables do we need, in order to model the relationship among U_1, U_2, U_3, and R_3?
   - We need also R_1 and R_2.
2. **Multiply** to compute joint probability:
$$P(R_1,R_2,R_3,U_1,U_2,U_3) = P(R_1)P(U_1|R_1)\ldots P(U_3|R_3)$$
3. **Add** to eliminate those we don't care about
$$P(R_2,U_1,U_2,U_3) = \sum_{R_1,R_2} P(R_1,R_2,R_3,U_1,U_2,U_3)$$
4. **Divide:** use Bayes' rule to get the desired conditional
$$P(R_3|U_1,U_2,U_3) = P(R_3,U_1,U_2,U_3)/P(U_1,U_2,U_3)$$



Belief Propagation

- What is $P(Y_t | X_1 = x_1, \dots, X_T = x_T)$?
- The observations are $X_1 = x_1, \dots, X_T = x_T$.
- The state variables are hidden. We need to find $P(Y_1 = y_1, X_1 = x_1, \dots, Y_T = y_T, X_T = x_T)$ for every possible setting of $\{Y_1, \dots, Y_T\}$.
- If there are T state variables, each with N possible values, then there are N^T possible combinations!
- The computational complexity can be $O\{N^2\}$ instead of $O\{N^T\}$ if you alternate the multiply and add steps, one new variable at a time.

Outline

- Belief propagation
  - What is $P(Y_t|X_1 = x_1, \dots, X_T = x_T)$?
- Viterbi Algorithm
  - What is the most probable sequence $\{Y_1, \dots, Y_T\}$ given observations $\{X_1 = x_1, \dots, X_T = x_T\}$?

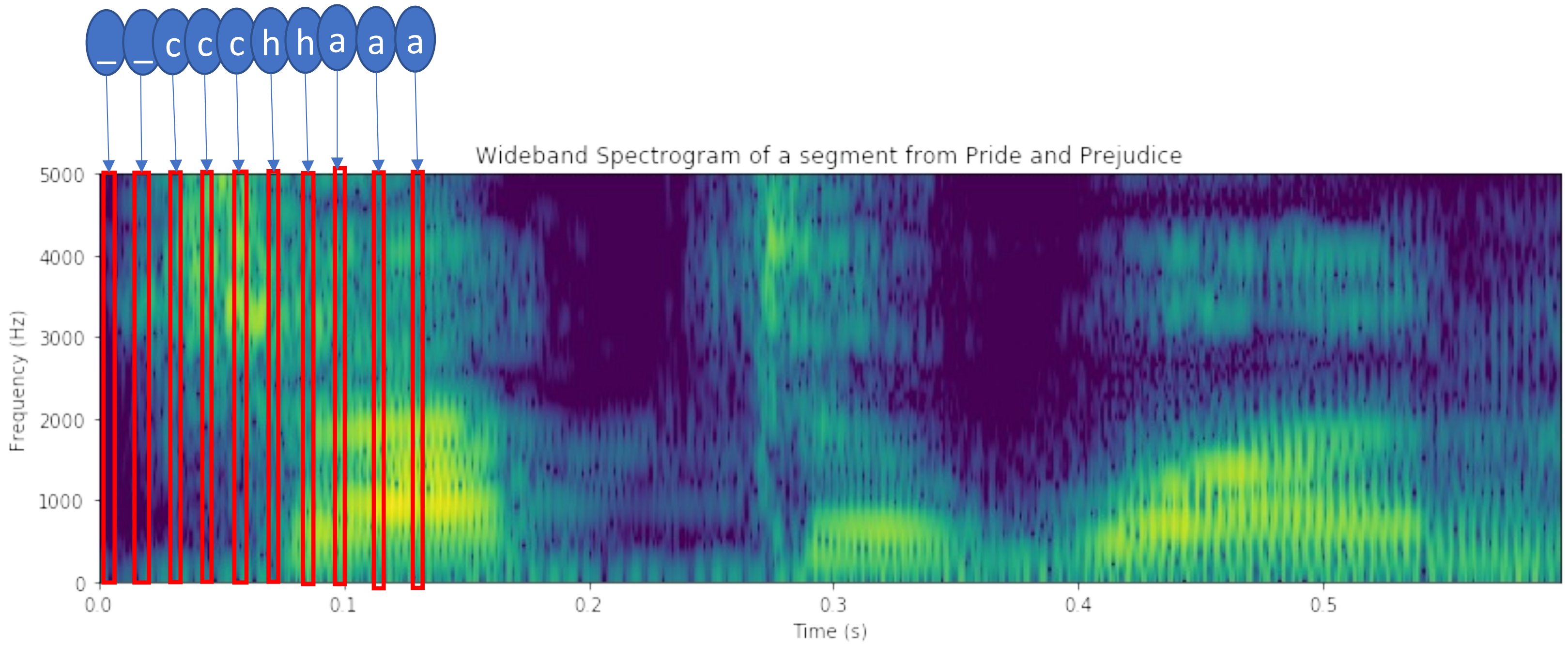
Viterbi algorithm: inferring the entire sequence

- Belief propagation answers questions like "what is $P(Y_t|X_1 = x_1, \ldots, X_T = x_T)$?" In other words, questions about one query variable, Y_t, given an arbitrary number of observed variables, $X_1 = x_1, \ldots, X_T = x_T$.
- Because there is only one query variable, we can keep the computational complexity down to $O\{N^2\}$ by alternating the multiply and add steps, getting rid of every other hidden variable as soon as it's no longer necessary.
- But what if we want to know the most likely values of all of the hidden variables?

Example: Speech Recognition

- Observations: X_t = spectrum of 25ms frame of the speech signal.
- State: Y_t = phoneme or letter being currently produced

The goal of speech recognition: find the most probable sequence $\{Y_1, \ldots, Y_T\}$ given observations $\{X_1 = x_1, \ldots, X_T = x_T\}$.



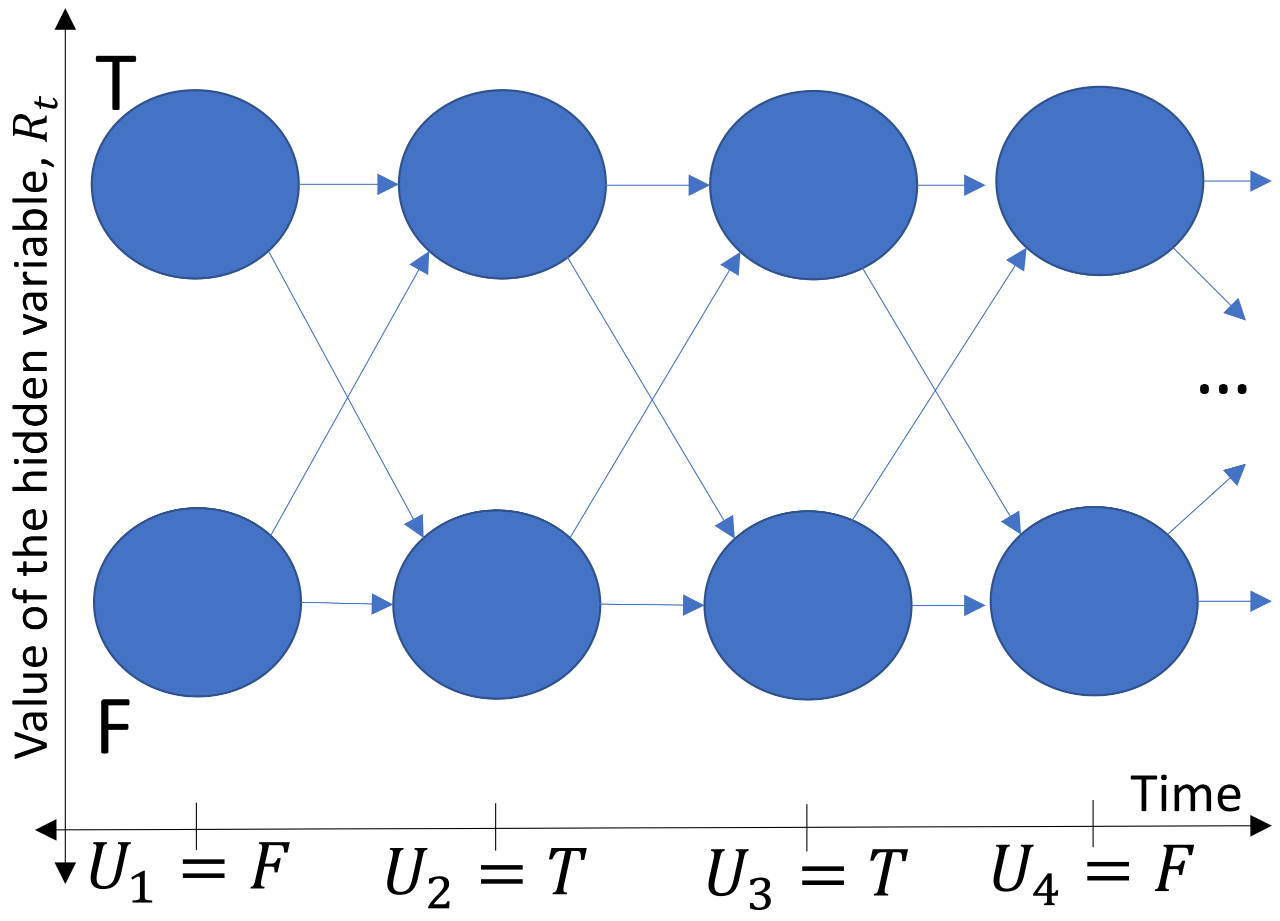
Viterbi Algorithm Example

Given a particular sequence of observations, what is the most likely underlying sequence of states?

- Example: given $U_1 = F, U_2 = T, U_3 = T, U_4 = F$
- what is the most likely sequence of state variables, R_1, R_2, R_3, R_4?

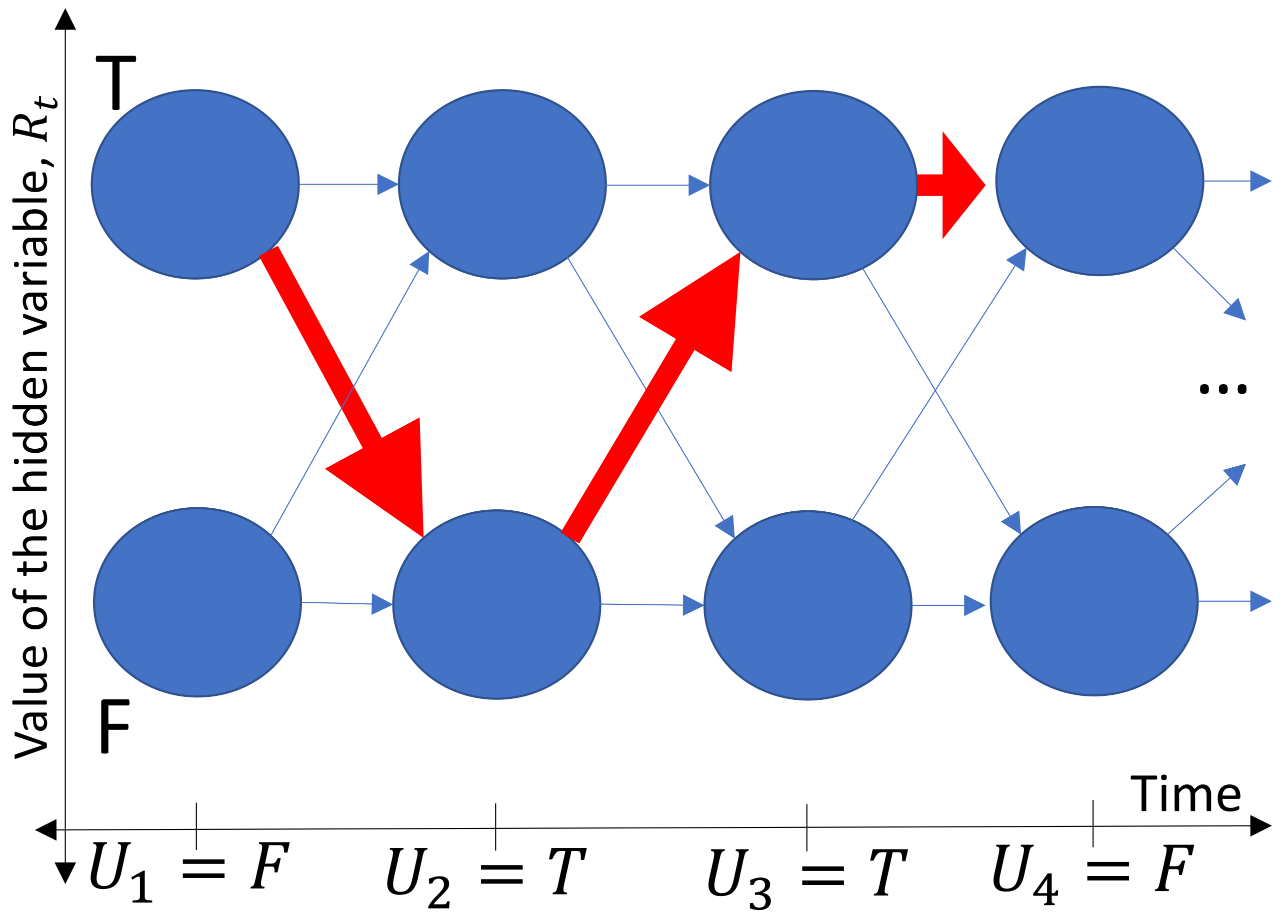
The Trellis

- X-Axis = time
- Y-Axis = state variable (R_t)
- Node = a particular state at a particular time
- Edge = possible transition from R_{t-1} to R_t



A Path Through the Trellis

- A path through the trellis is a sequence of connected states.
- For example, this path is the sequence $R_1 = T, R_2 = F, R_3 = T, R_4 = T$



Viterbi Algorithm Key Concept

Given a particular sequence of observations, what is the most likely underlying sequence of states?

In other words, given a particular sequence of observations, what is the most probable path through the trellis?

Viterbi Algorithm: Key concepts

Nodes and edges have numbers attached to them:

- **<u>Edge Probability</u>**: Probability of taking that transition, and then generating the next observed output

$$e_{ijt} = P(R_t = j, U_t = u_t | R_{t-1} = i)$$

- **<u>Node Probability</u>**: Probability of the best path until node j at time t

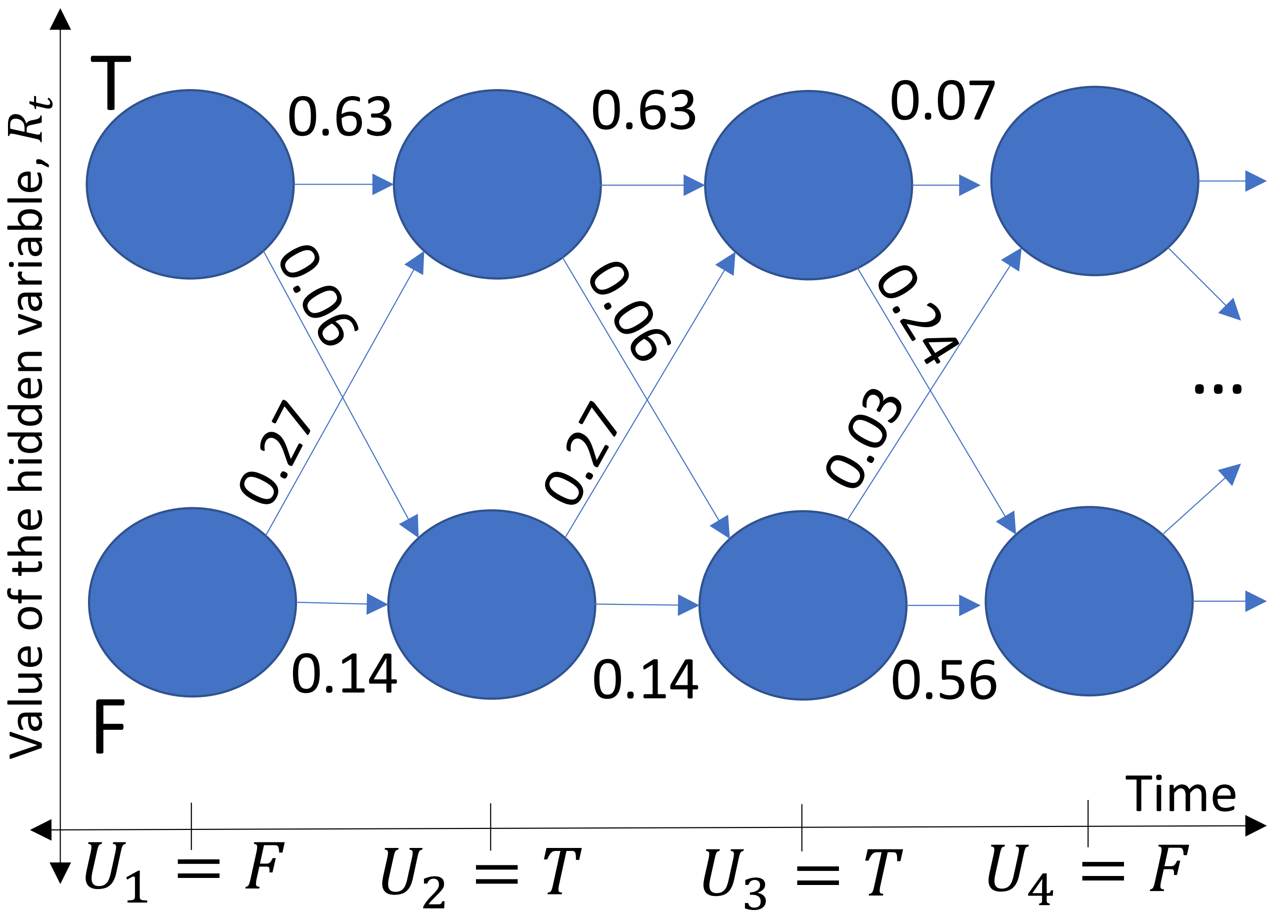
$$v_{jt} = \max_{r_1,\dots,r_{t-1}} P(U_1 = u_1 \dots, U_t = u_t, R_1 = r_1, \dots, R_t = j)$$

Edge Probabilities

$$e_{ijt} = P(R_t = j, U_t = u_t | R_{t-1} = i)$$

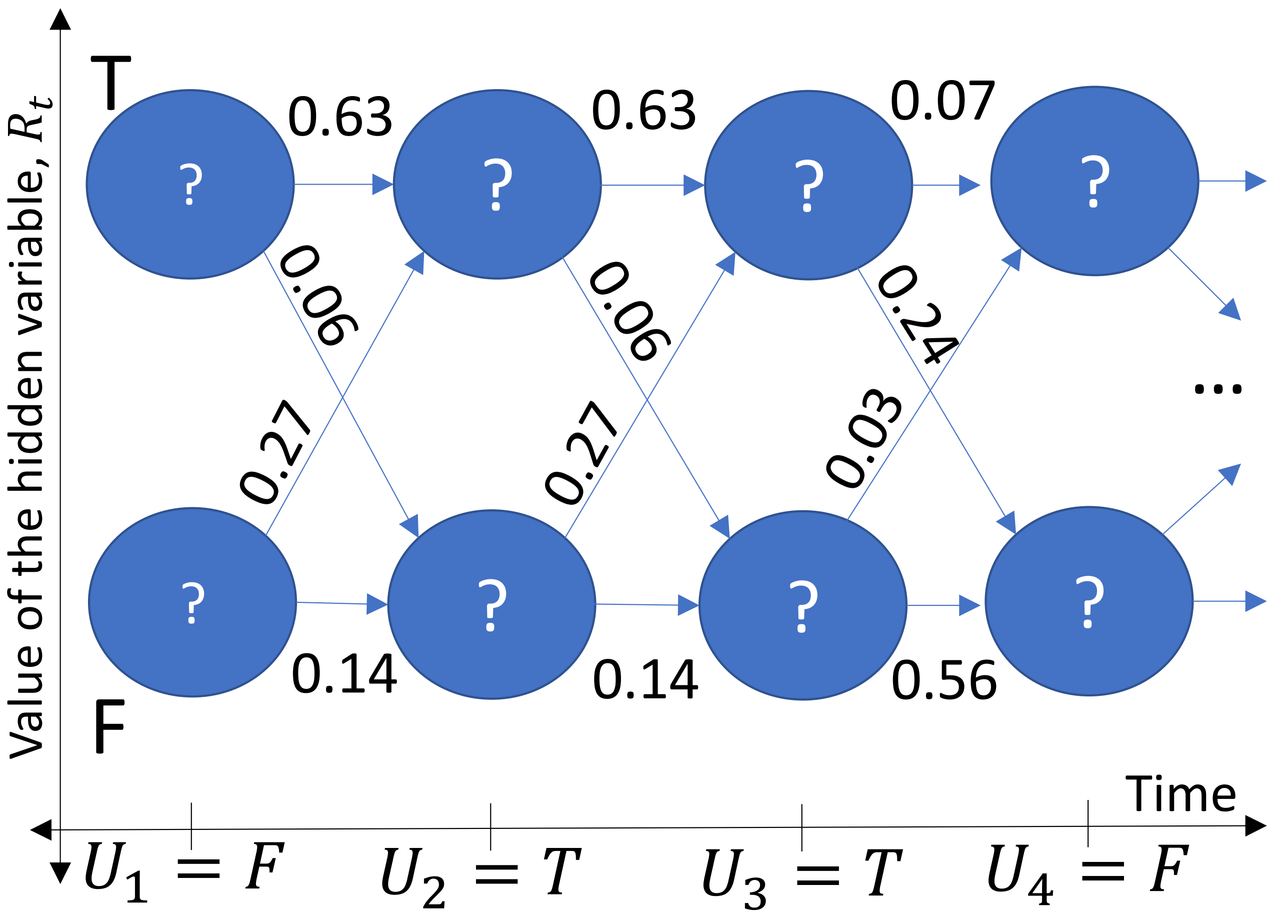
$$= P(R_t = j | R_{t-1} = i) \times P(U_t = u_t | R_t = j)$$

Notice that, since U_2 and U_3 have the same observed values, their inbound edges have the same weights.



Node Probabilities

$$v_{jt} = \max_{r_1,\ldots,r_{t-1}} P(U_1 = u_1 \ldots, U_t = u_t, R_1 = r_1, \ldots, R_t = j)$$

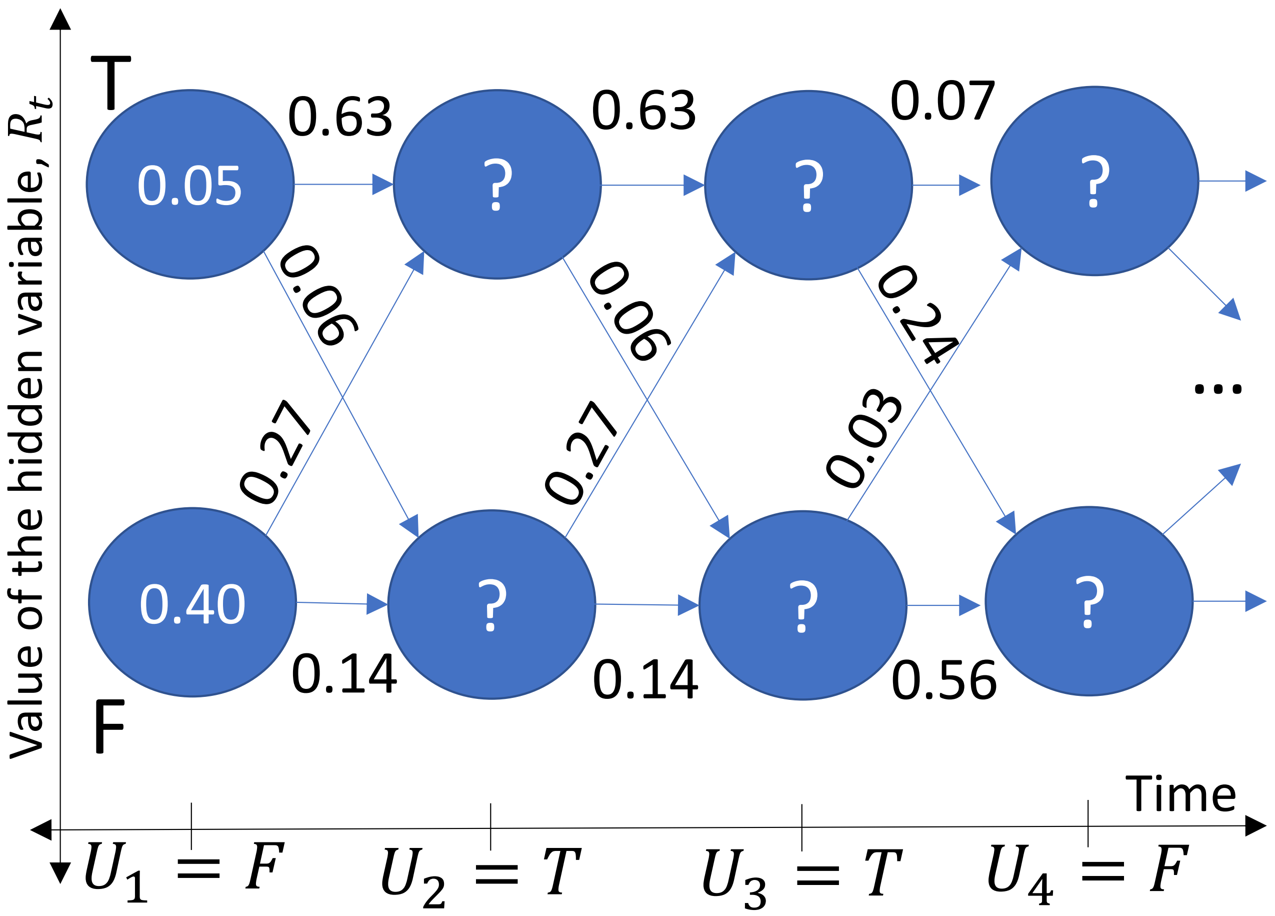


Node Probabilities: Initialization

For example, let's consider how to find v_{jt} for $t = 1$:

$$v_{j1} = P(U_1 = u_1, R_1 = j)$$
$$= P(R_1 = j)$$
$$\times P(U_1 = u_1 | R_1 = j)$$

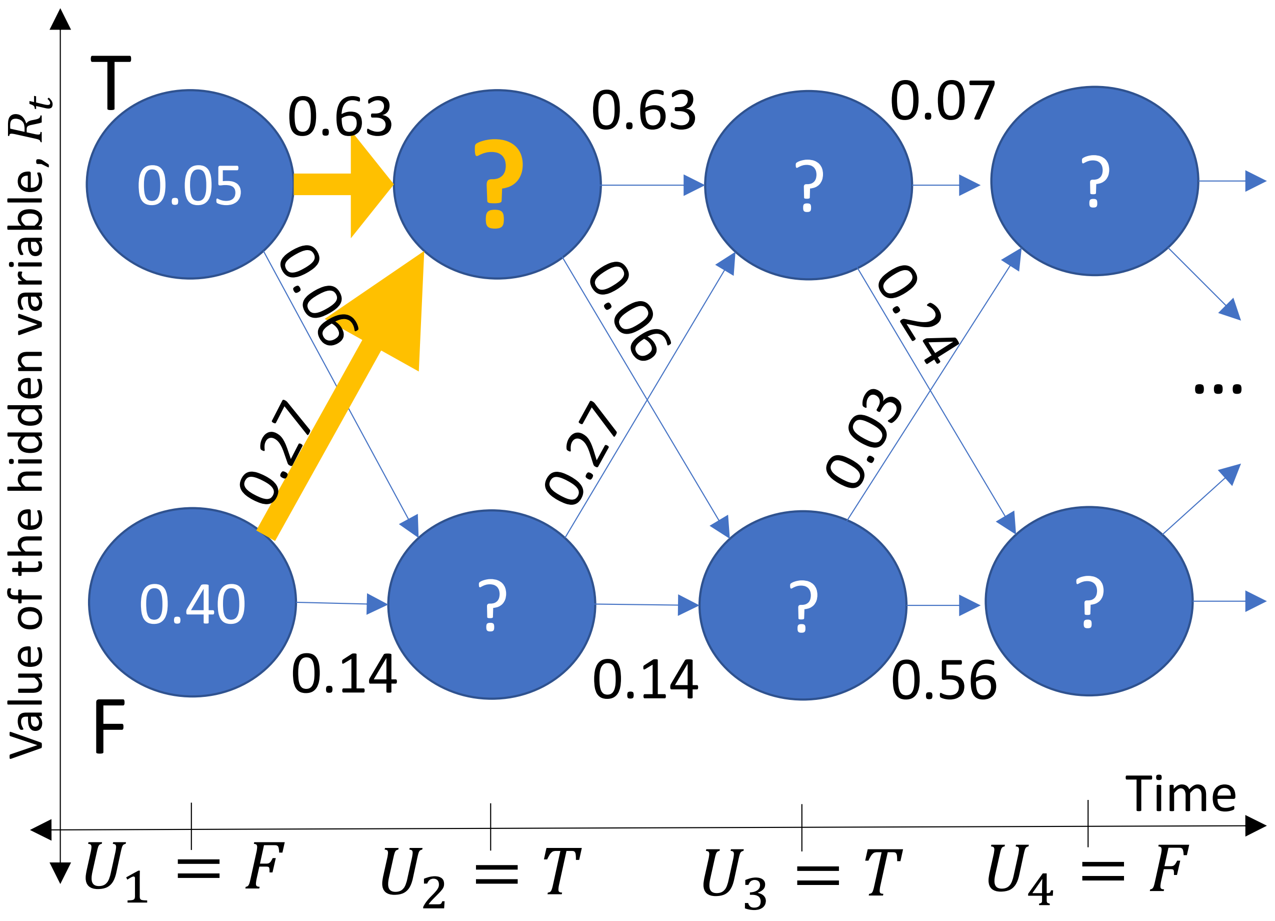
$$= \begin{cases} (0.5)(0.1) & j = T \\ (0.5)(0.8) & j = F \end{cases}$$



… and what about time t=2?

Notice that, at time t=2, there are two ways to get to any particular state:

- The previous state might have been F
- The previous state might have been T



Viterbi Algorithm: the iteration step

Given edge probabilities defined as

$$e_{ijt} = P(R_t = j, U_t = u_t | R_{t-1} = i)$$

and node probabilities defined as

$$v_{jt} = \max_{r_1,\dots,r_{t-2},i} P(U_1 = u_1 \dots, U_t = u_t, R_1 = r_1, \dots, R_{t-1} = i, R_t = j)$$

The node probability can be efficiently computed as

$$v_{jt} = \max_i \left(\max_{r_1,\dots,r_{t-2}} P(U_1 = u_1 \dots, U_{t-1} = u_{t-1}, R_1 = r_1, \dots, R_{t-1} = i) \times P(R_t = j, U_t = u_t | R_{t-1} = i) \right)$$

Viterbi Algorithm: the iteration step

Given edge probabilities defined as

$$e_{ijt} = P(R_t = j, U_t = u_t | R_{t-1} = i)$$

and node probabilities defined as

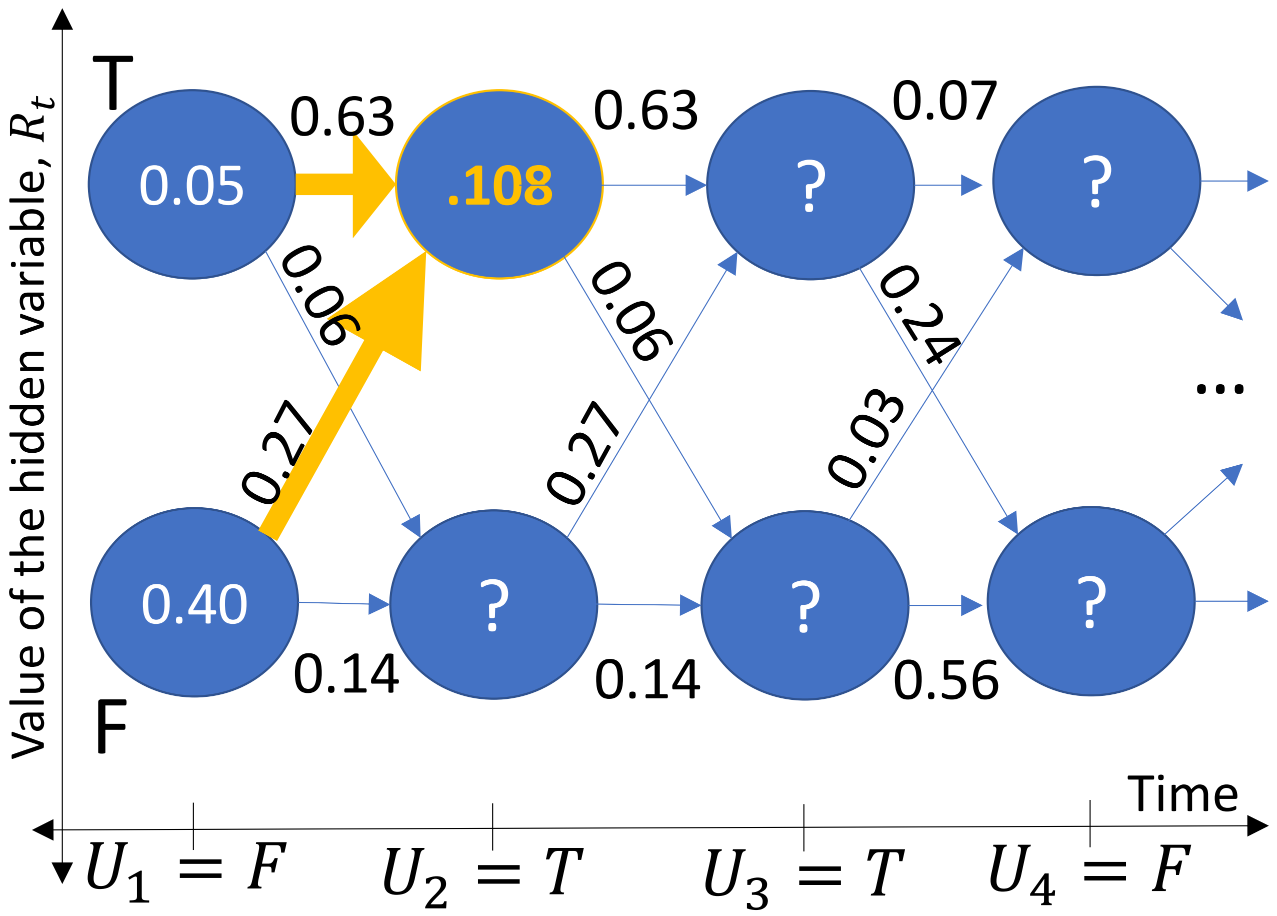
$$v_{jt} = \max_{r_1,\ldots,r_{t-2},i} P(U_1 = u_1 \ldots, U_t = u_t, R_1 = r_1, \ldots, R_{t-1} = i, R_t = j)$$

The node probability can be efficiently computed as

$$v_{jt} = \max_i v_{i,t-1} e_{ijt}$$

… and what about time t=2?

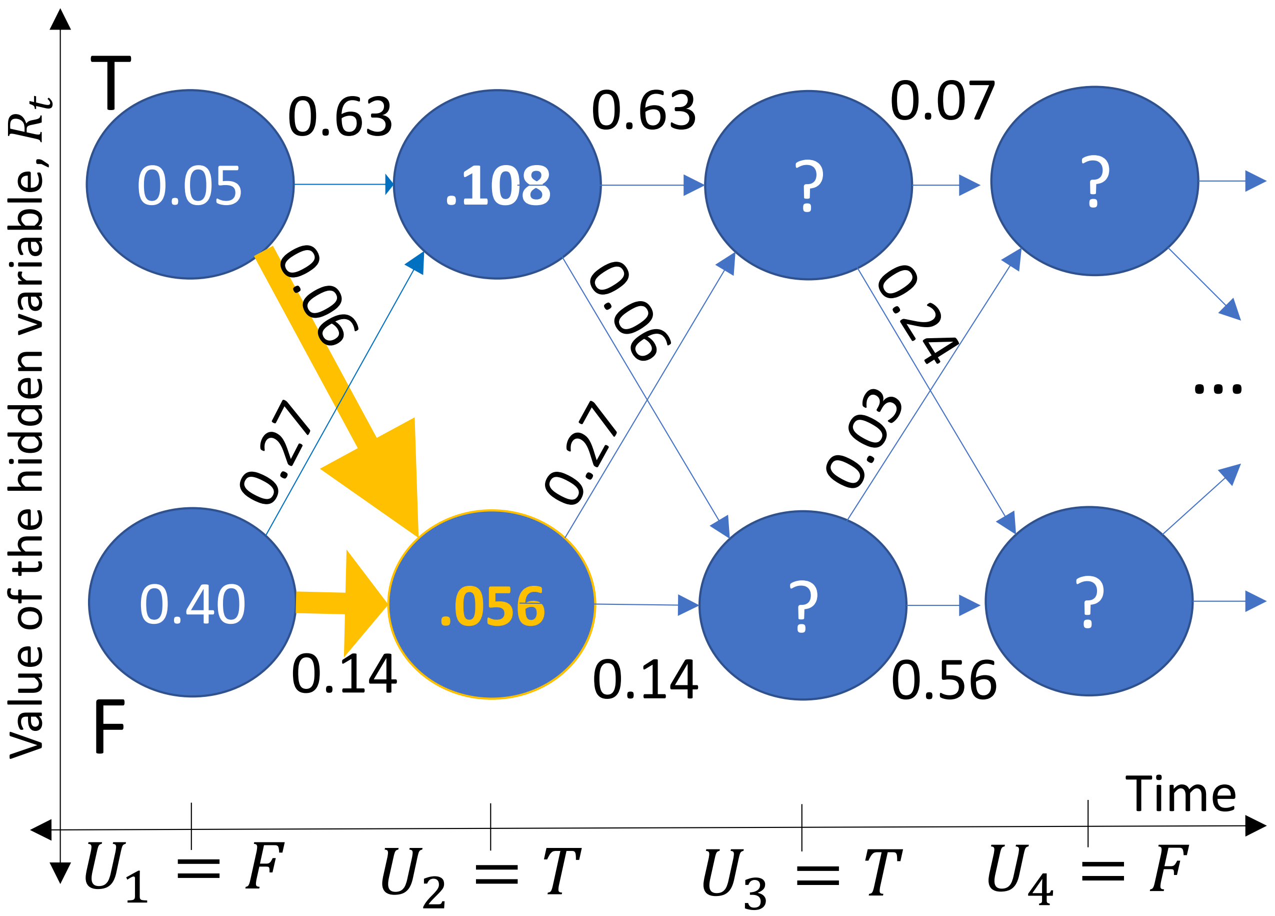
$$v_{T,2} = \max_i \; v_{i1} e_{i,T,2}$$
$$= \max\big((0.05)(0.63), (0.4)(0.27)\big)$$
$$= \; 0.108$$



… and what about time t=2?

$$v_{T,2} = \max_i v_{i1} e_{i,T,2}$$
$$= \max\big((0.05)(0.63), (0.4)(0.27)\big)$$
$$= \ 0.108$$

$$v_{F,2} = \max_i \ v_{i1} e_{i,F,2}$$
$$= \max\big((0.05)(0.06), (0.4)(0.14)\big)$$
$$= \ 0.056$$



Node probabilities and backpointers

- **Node Probability**: Probability of the best path until node j at time t

$$v_{jt} = \max_{r_1,\ldots,r_{t-2},i} P(U_1 = u_1 \ldots, U_t = u_t, R_1 = r_1, \ldots, R_{t-1} = i, R_t = j)$$

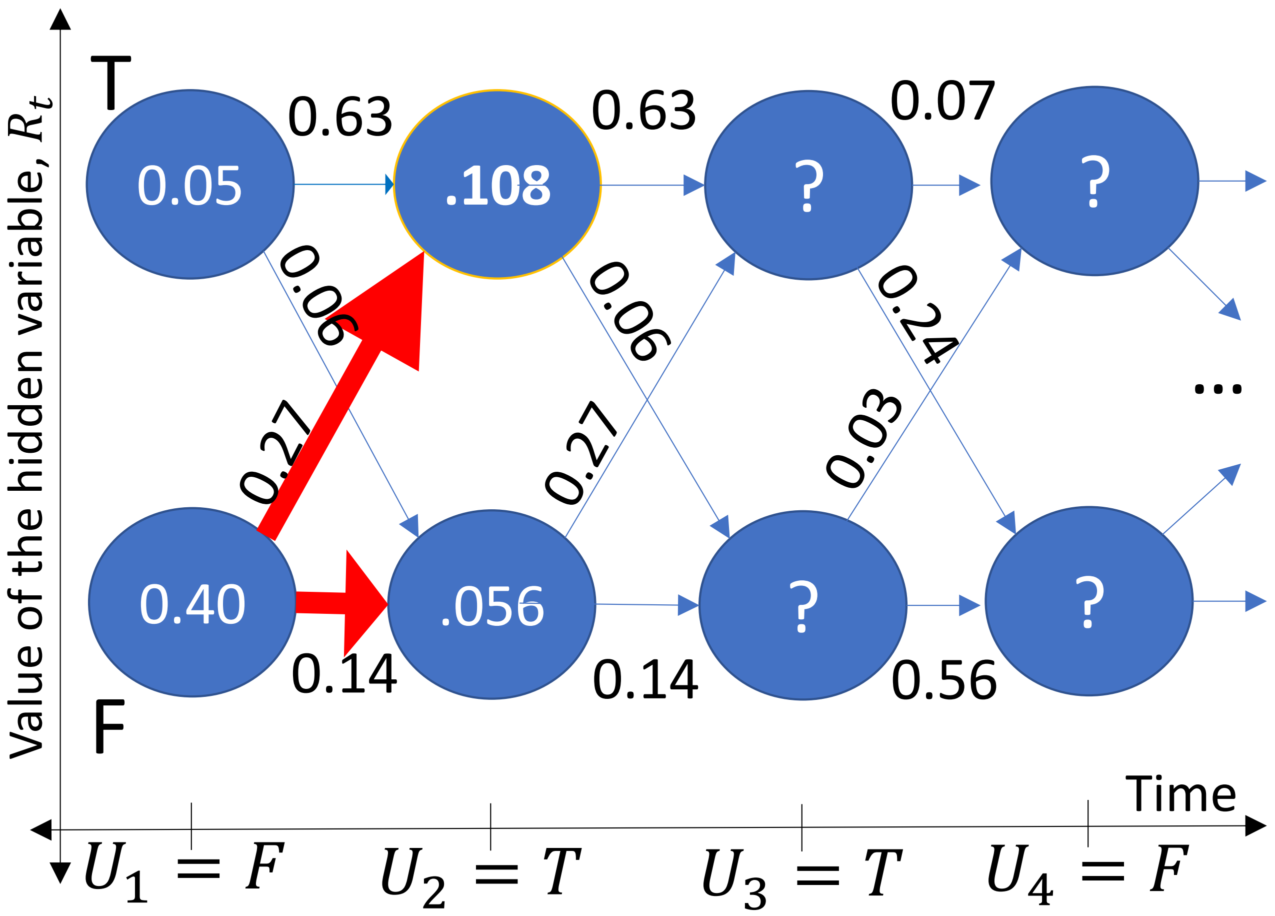
- **Backpointer**: which node, i, precedes node j on the best path?

$$i^*_{jt} = \underset{r_1,\ldots,r_{t-2},i}{\operatorname{argmax}} P(U_1, = u_1 \ldots, U_t = u_t, R_1 = r_1, \ldots, R_{t-1} = i, R_t = j)$$

Backpointers at t=2

$$i^*_{T,2} = \underset{i}{\operatorname{argmax}}\, v_{i1} e_{i,T,2}$$
$$= \operatorname{argmax}\begin{pmatrix}(0.05)(0.63),\\ (0.4)(0.27)\end{pmatrix}$$
$$= F$$

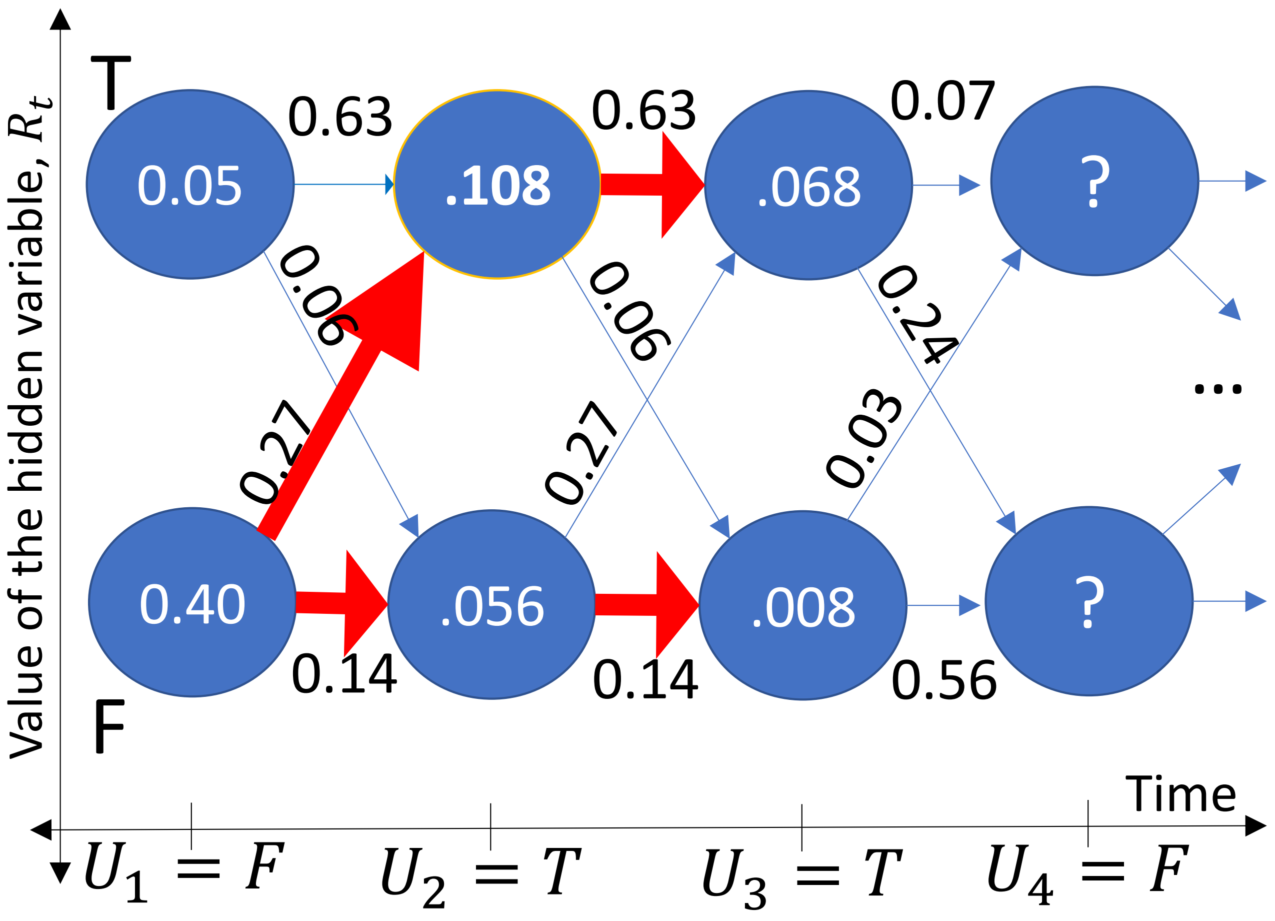
$$i^*_{F,2} = \underset{i}{\operatorname{argmax}}\, v_{i1} e_{i,F,2}$$
$$= \operatorname{argmax}\begin{pmatrix}(0.05)(0.06),\\ (0.4)(0.14)\end{pmatrix}$$
$$= F$$



Backpointers at t=3

$$i^*_{T,3} = \underset{i}{\text{argmax}}\, v_{i2} e_{i,T,3}$$
$$= \text{argmax}\begin{pmatrix}(0.108)(0.63),\\(0.056)(0.27)\end{pmatrix}$$
$$= T$$

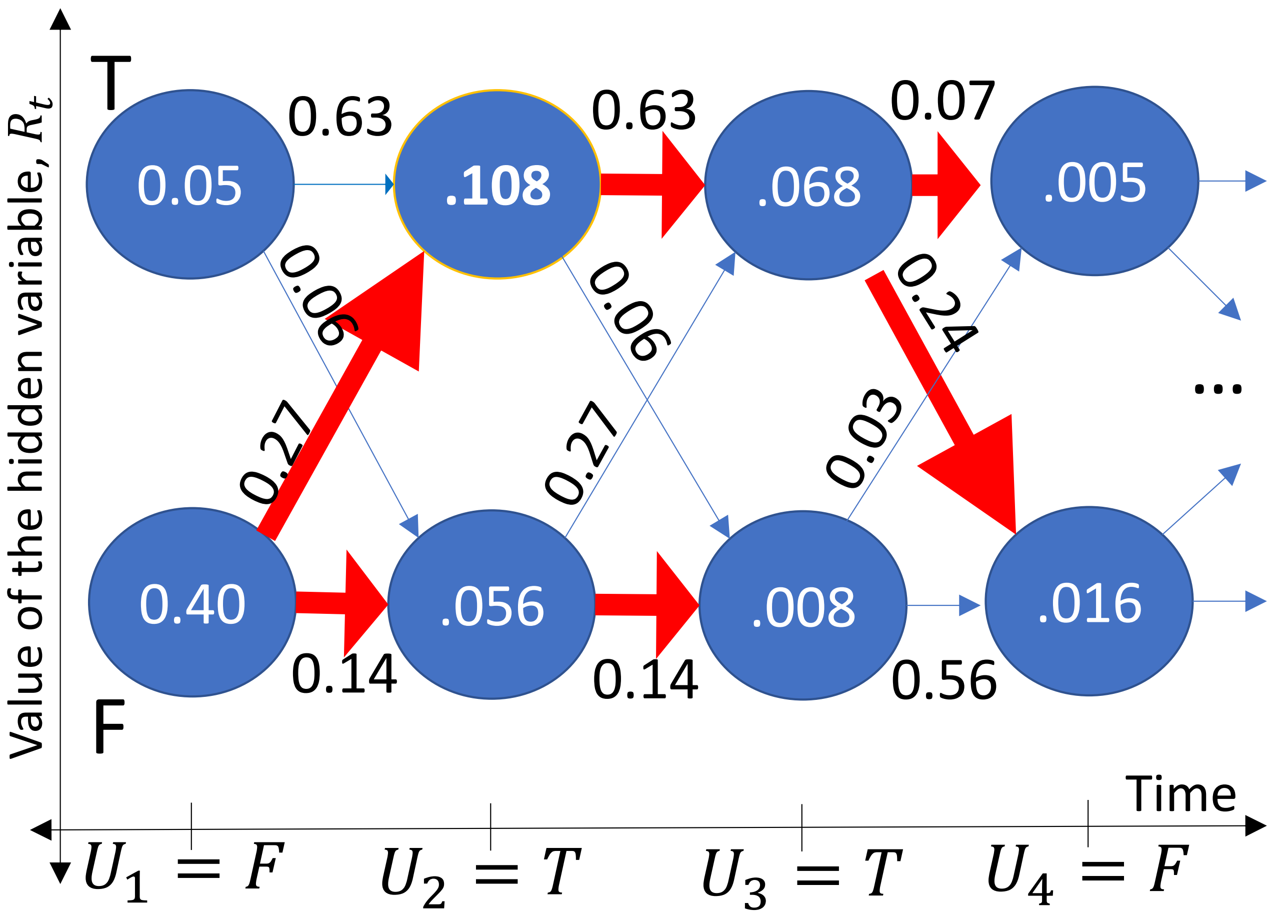
$$i^*_{F,3} = \underset{i}{\text{argmax}}\, v_{i2} e_{i,F,3}$$
$$= \text{argmax}\begin{pmatrix}(0.108)(0.06),\\(0.056)(0.14)\end{pmatrix}$$
$$= F$$



Backpointers at t=4

$$i^*_{T,4} = \operatorname*{argmax}_i v_{i3} e_{i,T,4}$$
$$= \operatorname{argmax}\begin{pmatrix}(0.068)(0.07),\\(0.008)(0.03)\end{pmatrix}$$
$$= T$$

$$i^*_{F,4} = \operatorname*{argmax}_i v_{i3} e_{i,F,4}$$
$$= \operatorname{argmax}\begin{pmatrix}(0.068)(0.24),\\(0.008)(0.56)\end{pmatrix}$$
$$= T$$



So which is the best path?

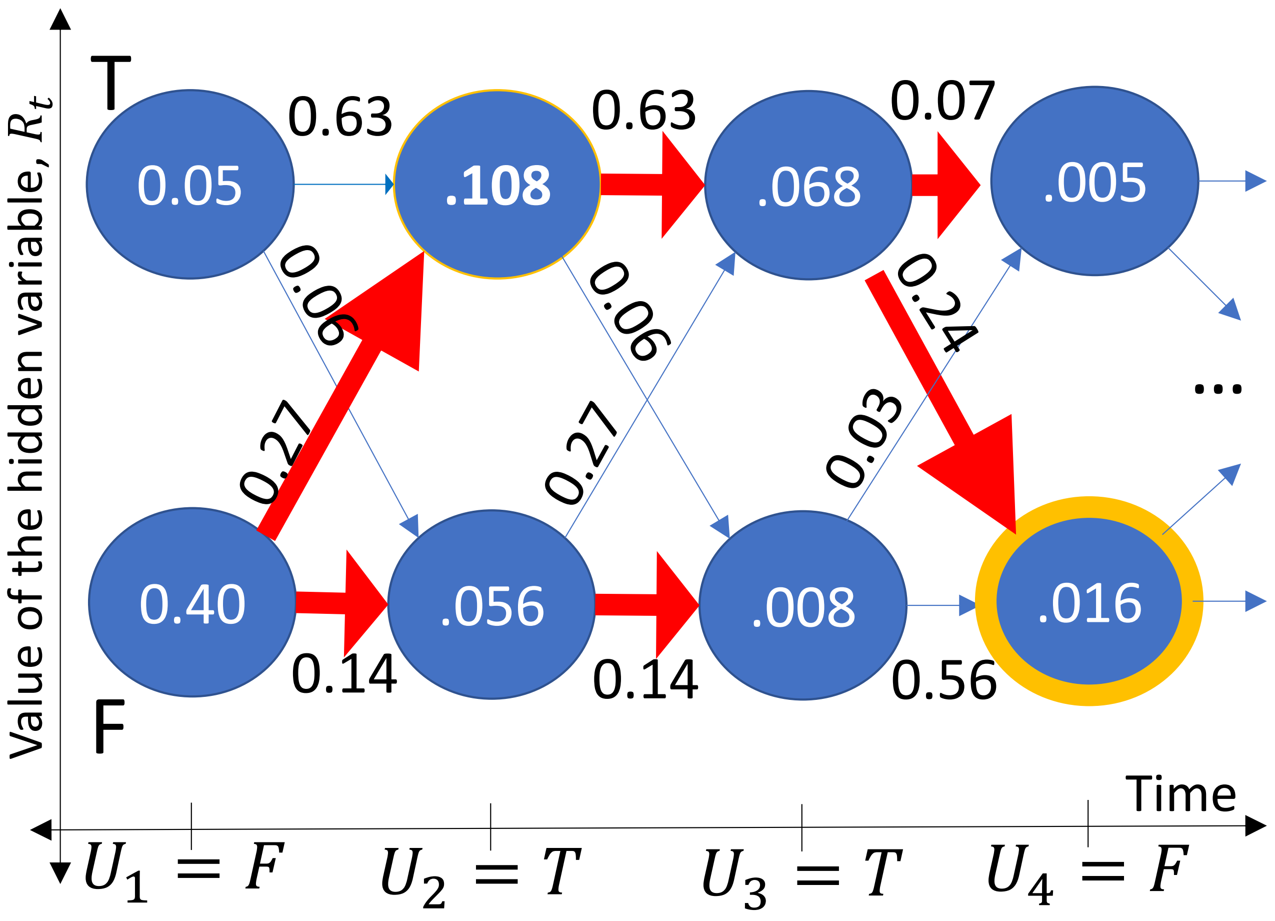
- Answer: whichever one is most probable.

Node probabilities at t=4

$$v_{T,4} = \max_i v_{i3} e_{i,T,4}$$
$$= \max \begin{pmatrix} (0.068)(0.07), \\ (0.008)(0.03) \end{pmatrix}$$
$$= 0.005$$

$$v_{F,4} = \max_i v_{i3} e_{i,F,4}$$
$$= \max \begin{pmatrix} (0.068)(0.24), \\ (0.008)(0.56) \end{pmatrix}$$
$$= 0.016$$

The best path is the one that ends at $R_4 = F$



Termination: which is the best path?

- Best final state is whichever final state has the highest node probability.
- The best path leading to that state is the most probable one
- … but we’ve already found the most probable path…
- …we just need to follow the backpointers!

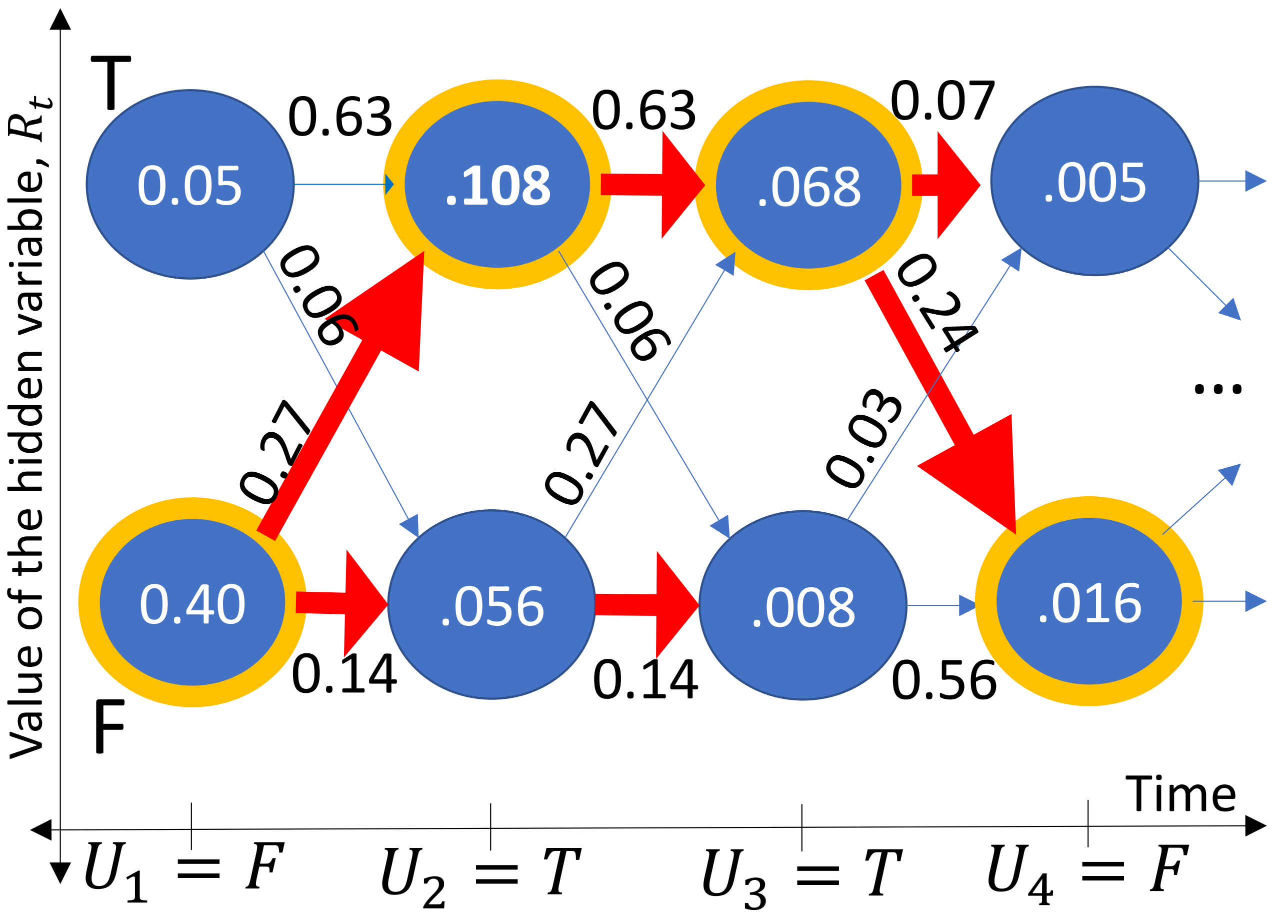
Follow the backpointers!

Given the observations

$$U_1, U_2, U_3, U_4 = F, T, T, F$$

… and given our hidden Markov model, we conclude that the most probable sequence of state variables is

$$R_1, R_2, R_3, R_4 = F, T, T, F$$



Some conclusions

- We have discovered that, if Elspeth brings her umbrella only on days 2 and 3, then the best inference is that it’s raining on only those days.

Some conclusions

Other types of HMMs might be less obvious. For example, consider the following assertion:

A fly flies well. A well does not fly.

In order to decide if these sentences are true or false, you first need to know which words are nouns, which verbs, and which adverbs.

In MP4, you will solve this problem using an HMM.

- State variable = part of speech
- Observation = word
- Transition model: verbs tend to come after nouns.

Final Word: Computational Complexity

- Inference by Enumeration in an HMM: $\mathcal{O}\{N^T\}$

$$P(\text{vars you care about}) = \sum_{\text{don}'\text{t}-\text{care vars}} P(\text{all vars})$$

… the complexity can be reduced to $\mathcal{O}\{TN^2\}$ if you add over each don't-care variable as soon as you no longer need it.

- Decoding using the Viterbi Algorithm: $\mathcal{O}\{TN^2\}$

$$v_{jt} = \max_i v_{i,t-1} e_{ijt}$$

Max over N values of i, performed for N values of j, and for T values of t.