CS440/ECE448 Lecture 19: The Forward Algorithm and the Viterbi Algorithm

Mark Hasegawa-Johnson, 3/2020

CC-BY 3.0: You may remix or redistribute if you cite the source.

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

Outline

- Inference by Enumeration in an HMM
- Filtering using the Forward Algorithm
- Decoding using the Viterbi Algorithm

Inference by Enumeration

To calculate a probability $P(R_2|U_1,U_2)$:

1. **Select:** which variables do we need, in order to model the relationship among U_1, U_2, and R_2?
   - We need also R_0 and R_1.
2. **Multiply** to compute joint probability:

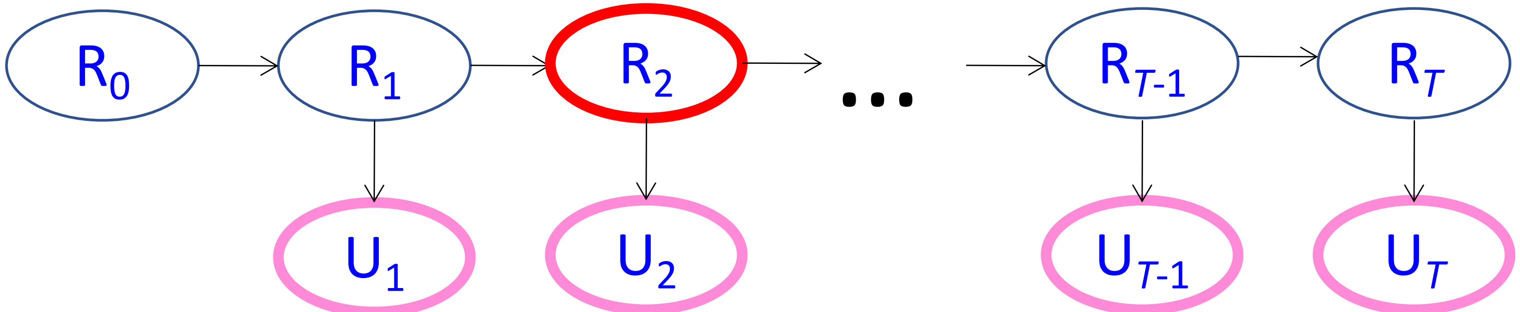
$$P(R_0, R_1, R_2, U_1, U_2) = P(R_0)P(R_1|R_0)P(U_1|R_1) \ldots P(U_2|R_2)$$

3. **Add** to eliminate those we don't care about

$$P(R_2, U_1, U_2) = \sum_{R_0, R_1} P(R_0, R_1, R_2, U_1, U_2)$$

4. **Divide:** use Bayes' rule to get the desired conditional

$$P(R_2|U_1,U_2) = P(R_2, U_1, U_2)/P(U_1, U_2)$$



Computational Complexity of "Inference by Enumeration"

- Russell & Norvig call this "inference by enumeration" because you have to enumerate every possible combination of R_0, R_1, R_2, U_1, U_2, for $R_0 \in \{t, f\}$.
- The complexity comes from this enumeration: if there are 2^5 possible combinations, then the complexity can't be less than 2^5!

First simplification for HMMs: only enumerate the values of the hidden variables

- Notice: we don't really need to calculate $P(R_0, \neg R_1, R_2, \neg U_1, \neg U_2)$ if we have already observed that U_2 is True!
- First computational simplification for HMMs:
  - Only enumerate the possible values of the hidden variables.
  - Set the observed variables to their observed values.

Inference by Enumerating only the Hidden Variables

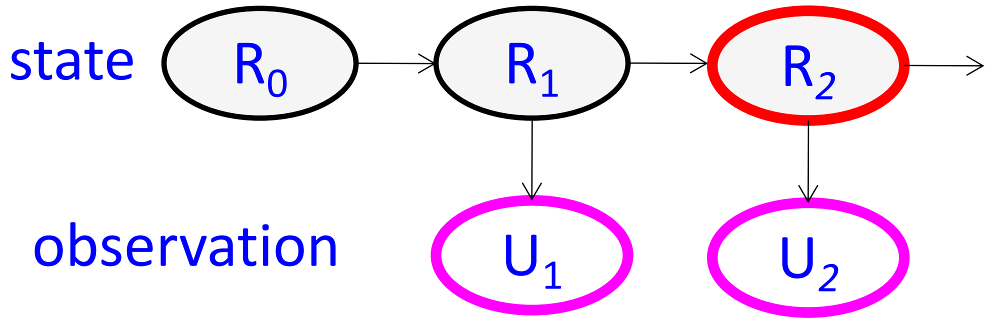
Multiply:

$$P(R_0, R_1, R_2, \neg U_1, U_2) =$$
$$P(R_0)P(R_1|R_0)P(\neg U_1|R_1) \dots P(U_2|R_2)$$

	$\neg R_2 U_2$	$R_2 U_2$
$\neg R_0 \neg R_1 \neg U_1$	0.0392	0.0756
$\neg R_0 R_1 \neg U_1$	0.0009	0.0095
$R_0 \neg R_1 \neg U_1$	0.0168	0.0324
$R_0 R_1 \neg U_1$	0.0021	0.0221

- We only compute joint probabilities that include the observed events, $\neg \boldsymbol{U_1}$ and $\boldsymbol{U_2}$.
- The numbers don't add up to one; they add up to $P(\neg U_1, U_2)$.

Transition model



Transition probabilities

	R_t = T	R_t = F
R_{t-1} = T	0.7	0.3
R_{t-1} = F	0.3	0.7

Observation probabilities

	U_t = T	U_t = F
R_t = T	0.9	0.1
R_t = F	0.2	0.8

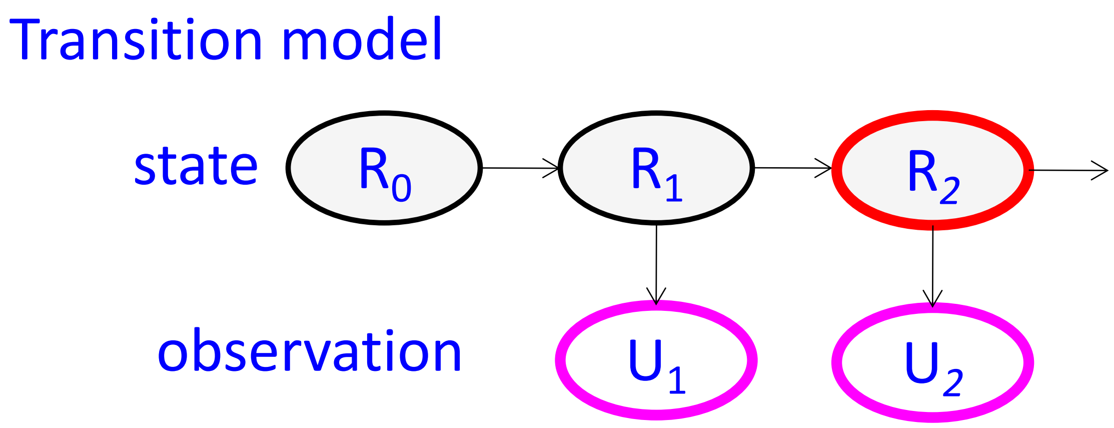
Inference by Enumerating only the Hidden Variables

Add:

$$P(R_2, \neg U_1, U_2) = \sum_{R_0, R_1} P(R_0, R_1, R_2, \neg U_1, U_2)$$

	$\neg U_1 U_2$
$\neg R_2$	0.059
R_2	0.1395

- We only compute joint probabilities that include the observed events, $\neg \boldsymbol{U_1}$ and $\boldsymbol{U_2}$.
- The numbers don't add up to one; they add up to $P(\neg U_1, U_2)$.



Transition probabilities

	R_t = T	R_t = F
R_{t-1} = T	0.7	0.3
R_{t-1} = F	0.3	0.7

Observation probabilities

	U_t = T	U_t = F
R_t = T	0.9	0.1
R_t = F	0.2	0.8

Inference by Enumerating only the Hidden Variables

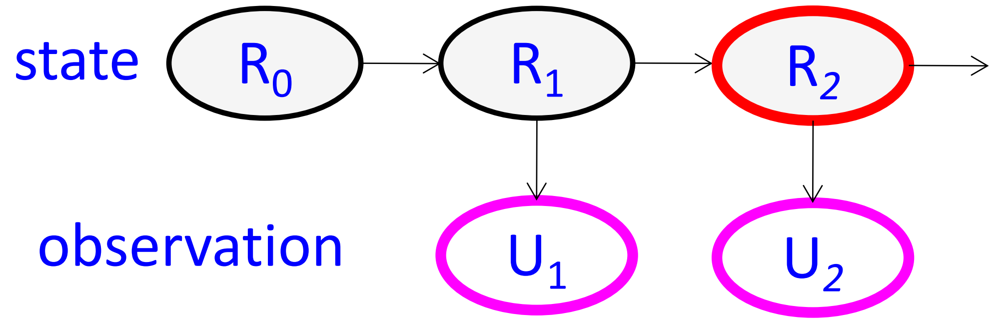
Divide:

$$P(R_2 | \neg U_1, U_2) = \frac{P(R_2, \neg U_1, U_2)}{P(\neg U_1, U_2)}$$

	$\neg U_1 U_2$
$\neg R_2$	0.30
R_2	0.70

- Normalize, so that the column sums to one.

Transition model



Transition probabilities

	R_t = T	R_t = F
R_{t-1} = T	0.7	0.3
R_{t-1} = F	0.3	0.7

Observation probabilities

	U_t = T	U_t = F
R_t = T	0.9	0.1
R_t = F	0.2	0.8

First simplification for HMMs: only enumerate the values of the hidden variables

- Only enumerate the possible values of the hidden variables. Set the observed variables to their observed values.
- **Filtering with binary hidden variables**: enumerate $(R_0, \ldots, R_T)$, complexity is $2^{T+1} = \mathcal{O}\{2^T\}$.
- **Filtering with N-ary hidden variables:** If each of the variables R_t has N possible values, instead of only 2 possible values, then the inference complexity would be $\mathcal{O}\{N^T\}$.

Outline

- Inference by Enumeration in an HMM
- Filtering using the Forward Algorithm
- Decoding using the Viterbi Algorithm

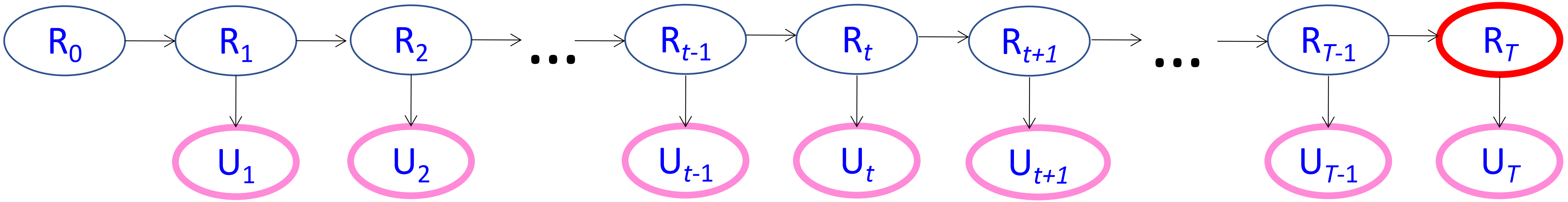
Inference complexity in an HMM

- $\mathcal{O}\{N^T\}$ is still a lot. Can we do better?
- For a general Bayes net, no. Bayes net inference, in an arbitrary Bayes net, is NP-complete.
- For an HMM, yes, we can do better.

The Forward Algorithm

- **Initialize**: look up the value of $\mathrm{P}(R_0)$.
- **Iterate**: for $1 \leq t \leq T$:
  - **Multiply**:
    $\mathrm{P}(R_{t-1}, R_t, U_1, \ldots, U_t) = \mathrm{P}(R_{t-1}, U_1, \ldots, U_{t-1})\mathrm{P}(R_t | R_{t-1})\mathrm{P}(U_t | R_t)$
    for the N^2 combinations of R_{t-1} and R_t.

When t=1, this is just $\mathrm{P}(\boldsymbol{R_0})$

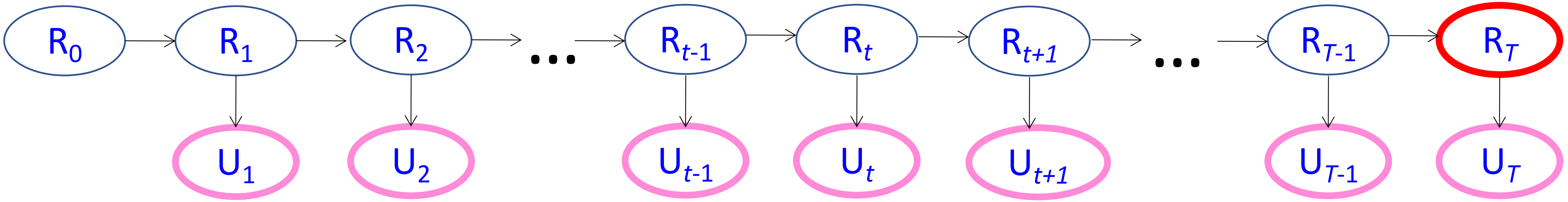


The Forward Algorithm

- **Initialize**: look up the value of $\mathrm{P}(R_0)$.
- **Iterate**: for $1 \le t \le T$:
  - **Multiply**:
    $\mathrm{P}(R_{t-1}, R_t, U_1, \ldots, U_t) = \mathrm{P}(R_{t-1}, U_1, \ldots, U_{t-1})\mathrm{P}(R_t | R_{t-1})\mathrm{P}(U_t | R_t)$
    for the N^2 combinations of R_{t-1} and R_t.
  - **Add:**

$$\mathrm{P}(R_t, U_1, \ldots, U_t) = \sum_{R_{t-1}} \mathrm{P}(R_{t-1}, R_t, U_1, \ldots, U_t)$$

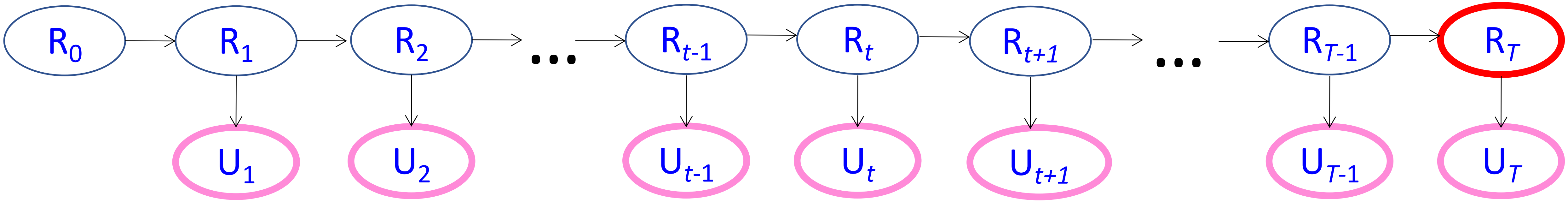
for the N possible values of R_t.



The Forward Algorithm

- **Initialize**: look up the value of $\mathrm{P}(R_0)$.
- **Iterate**: for $1 \le t \le T$:
  - **Multiply**:
    $\mathrm{P}(R_{t-1}, R_t, U_1, \ldots, U_t) = \mathrm{P}(R_{t-1}, U_1, \ldots, U_{t-1})\mathrm{P}(R_t | R_{t-1})\mathrm{P}(U_t | R_t)$
    for the N^2 combinations of R_{t-1} and R_t.
  - **Add:**
    $$\mathrm{P}(R_t, U_1, \ldots, U_t) = \sum_{R_{t-1}} \mathrm{P}(R_{t-1}, R_t, U_1, \ldots, U_t)$$
    for the N possible values of R_t.

When we move to the next value of t…

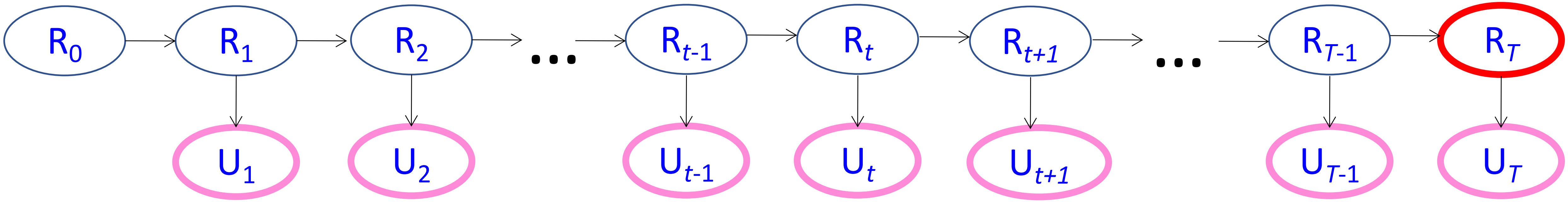


The Forward Algorithm

- **Initialize**: look up the value of $\mathrm{P}(R_0)$.
- **Iterate**: for $1 \le t \le T$: **... and so on, until we reach t=T...**
  - **Multiply**:
    $\mathrm{P}(R_{t-1}, R_t, U_1, \ldots, U_t) = \mathrm{P}(R_{t-1}, U_1, \ldots, U_{t-1})\mathrm{P}(R_t|R_{t-1})\mathrm{P}(U_t|R_t)$
    for the N^2 combinations of R_{t-1} and R_t.
  - **Add:**

$$\mathrm{P}(R_t, U_1, \ldots, U_t) = \sum_{R_{t-1}} \mathrm{P}(R_{t-1}, R_t, U_1, \ldots, U_t)$$

for the N possible values of R_t.

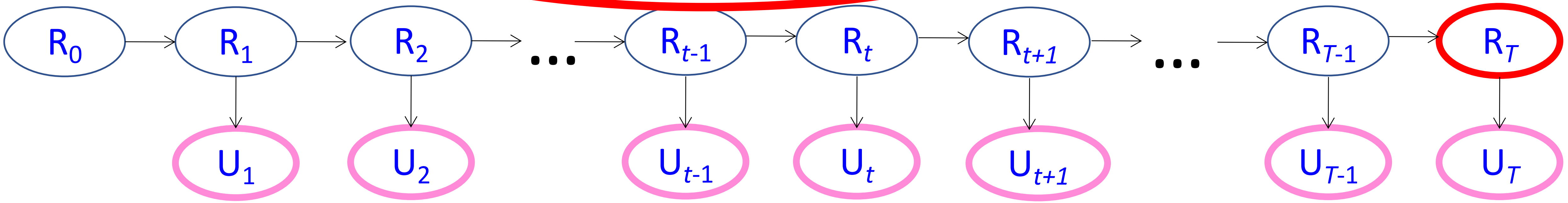


The Forward Algorithm

- **Initialize**: look up the value of $\mathrm{P}(R_0)$.
- **Iterate**: for $1 \le t \le T$:
  - **Multiply**:
    $\mathrm{P}(R_{t-1}, R_t, U_1, \ldots, U_t) = \mathrm{P}(R_{t-1}, U_1, \ldots, U_{t-1})\mathrm{P}(R_t|R_{t-1})\mathrm{P}(U_t|R_t)$
    for the N^2 combinations of R_{t-1} and R_t.
  - **Add:**

$$\mathrm{P}(R_t, U_1, \ldots, U_t) = \sum_{R_{t-1}} \mathrm{P}(R_{t-1}, R_t, U_1, \ldots, U_t)$$

    for the N possible values of R_t.
- **Terminate**: $\mathrm{P}(R_T|U_1, \ldots, U_T) = \frac{\mathrm{P}(R_T, U_1, \ldots, U_T)}{\mathrm{P}(U_1, \ldots, U_t)}$



The Forward Algorithm

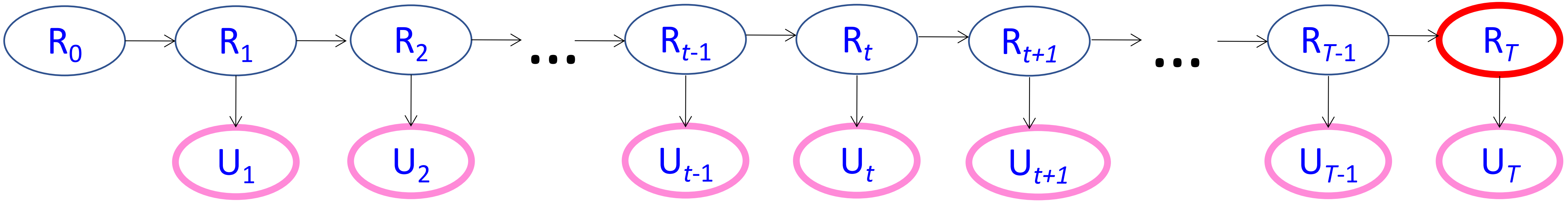
<u>Complexity</u>

- **Initialize**: look up the value of $\mathrm{P}(R_0)$. → $\mathcal{O}\{N\}$
- **Iterate**: for $1 \le t \le T$:
  - **Multiply**: $\mathrm{P}(R_{t-1}, R_t, U_1, \ldots, U_t) = \mathrm{P}(R_{t-1}, U_1, \ldots, U_{t-1})\mathrm{P}(R_t|R_{t-1})\mathrm{P}(U_t|R_t)$

    for the N^2 combinations of R_{t-1} and R_t. → $\mathcal{O}\{N^2T\}$
  - **Add:**

$$\mathrm{P}(R_t, U_1, \ldots, U_t) = \sum_{R_{t-1}} \mathrm{P}(R_{t-1}, R_t, U_1, \ldots, U_t)$$

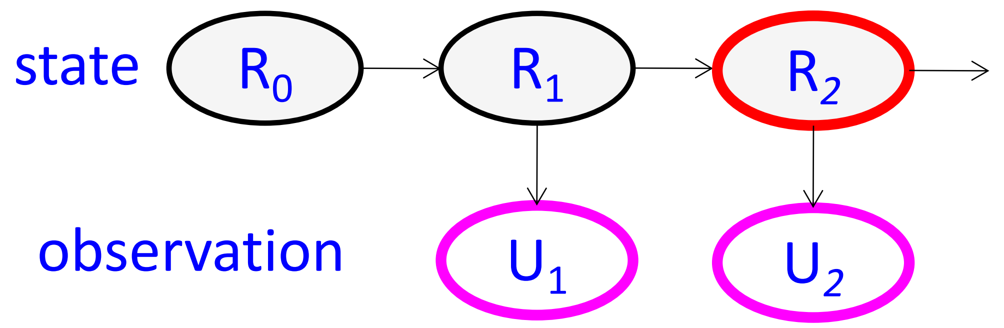
    for the N possible values of R_t. → $\mathcal{O}\{N^2T\}$
- **Terminate**: $\mathrm{P}(R_T|U_1, \ldots, U_T) = \frac{\mathrm{P}(R_T, U_1, \ldots, U_T)}{\mathrm{P}(U_1, \ldots, U_t)}$ → $\mathcal{O}\{N\}$



Example: Filtering in UmbrellaWorld

Richard notices that Ellie brought her umbrella today (U_2) but not yesterday ($\neg U_1$). Is it raining today? What is $P(R_2|\neg U_1, U_2)$?

Transition model



Initialize:

$$P(R_0) = \frac{1}{2}$$

$$P(\neg R_0) = \frac{1}{2}$$

Transition probabilities

	R_t = T	R_t = F
R_{t-1} = T	0.7	0.3
R_{t-1} = F	0.3	0.7

Observation probabilities

	U_t = T	U_t = F
R_t = T	0.9	0.1
R_t = F	0.2	0.8

Example: Filtering in UmbrellaWorld

Iterate $t = 1$:

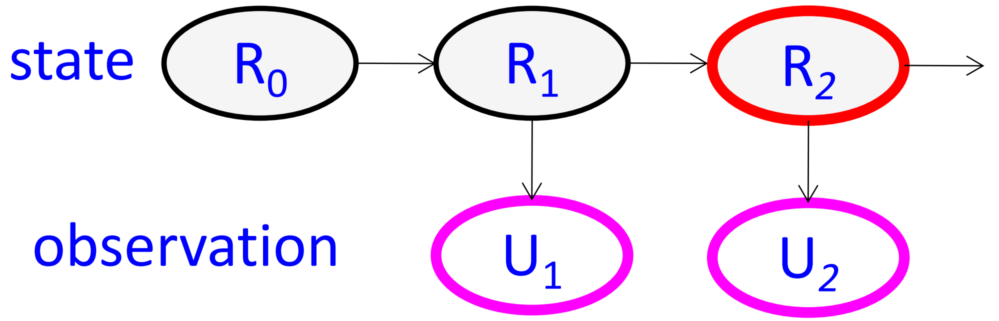
Multiply:

$$P(\neg R_0, \neg R_1, \neg U_1) = (0.5)(0.7)(0.8) = 0.28$$
$$P(\neg R_0, R_1, \neg U_1) = (0.5)(0.3)(0.1) = 0.015$$
$$P(R_0, \neg R_1, \neg U_1) = (0.5)(0.3)(0.8) = 0.12$$
$$P(R_0, R_1, \neg U_1) = (0.5)(0.7)(0.1) = 0.035$$

Add:

$$P(\neg R_1, \neg U_1) = 0.28 + 0.12 = 0.4$$
$$P(R_1, \neg U_1) = 0.015 + 0.035 = 0.05$$

Transition model



Transition probabilities

	R_t = T	R_t = F
R_{t-1} = T	0.7	0.3
R_{t-1} = F	0.3	0.7

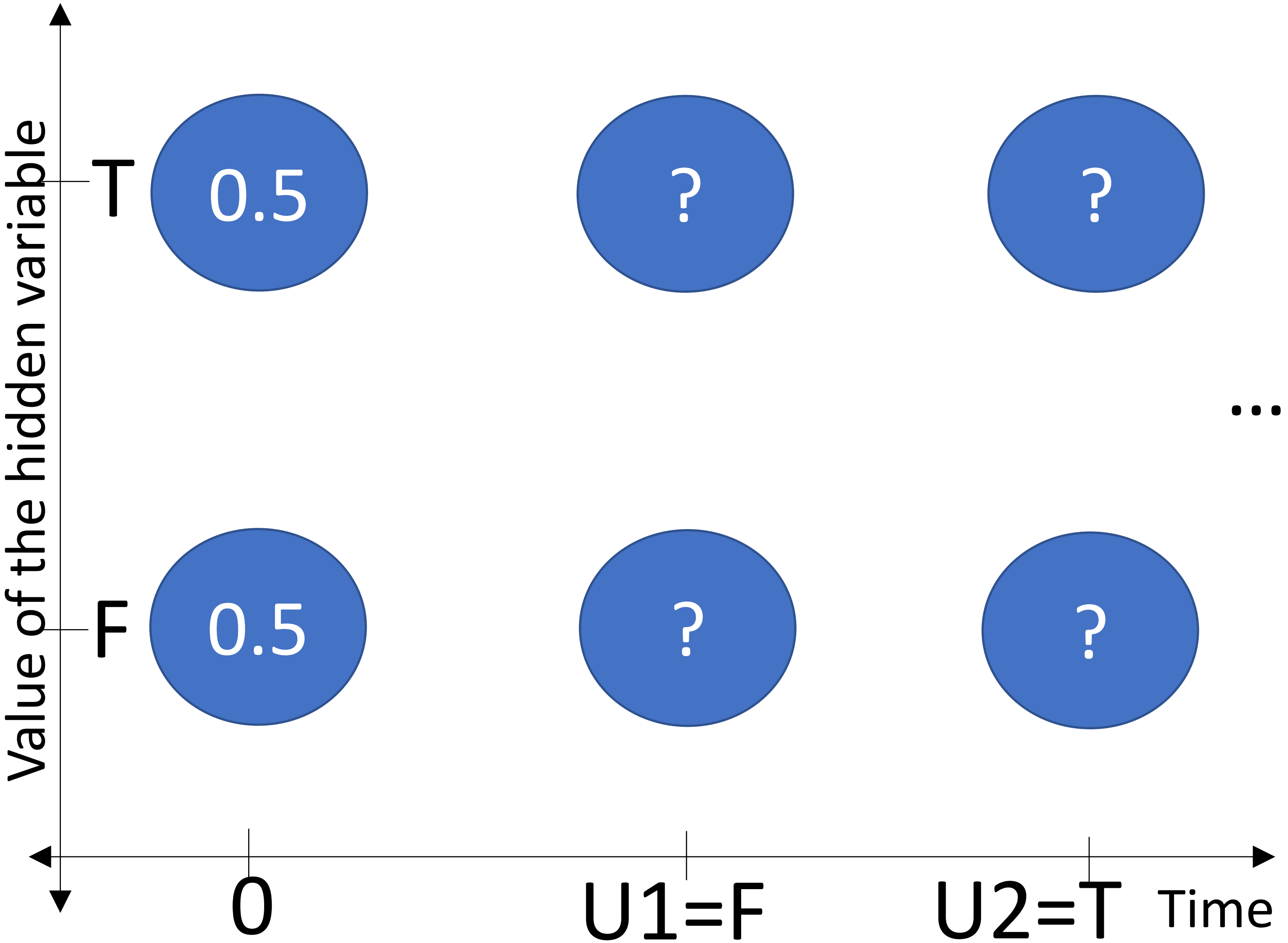
Observation probabilities

	U_t = T	U_t = F
R_t = T	0.9	0.1
R_t = F	0.2	0.8

Forward Algorithm: The Trellis

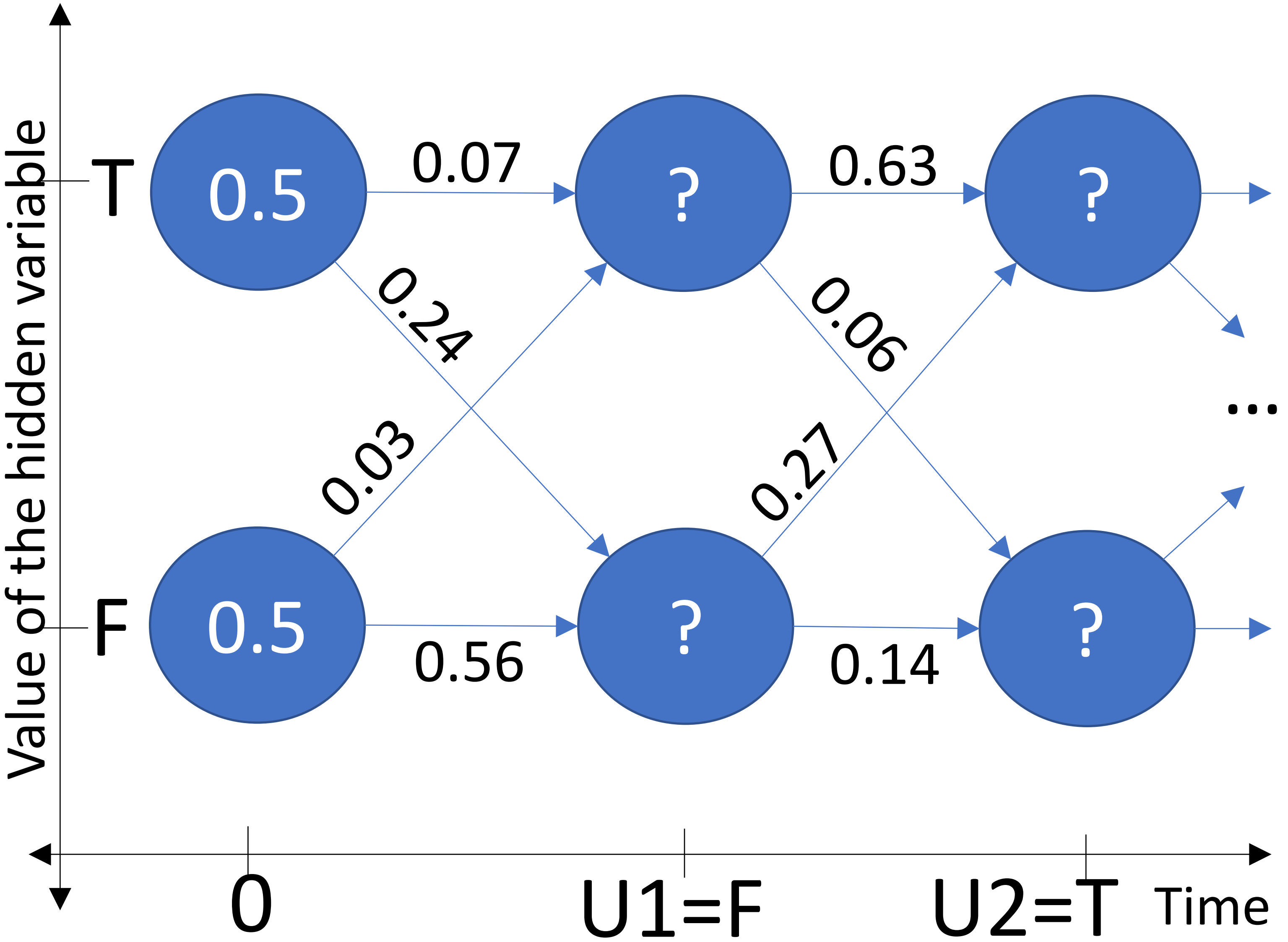
We can visualize the forward algorithm using a TRELLIS:

- Node = a value of the hidden variable at a given time
- Numerical value of the node = probability that the hidden variable takes that value



Forward Algorithm: The Trellis

- Edge = a possible transition from R_{t-1} to R_t
- Numerical value of the edge = $\mathrm{P}(R_t|R_{t-1})\mathrm{P}(U_t|R_t)$

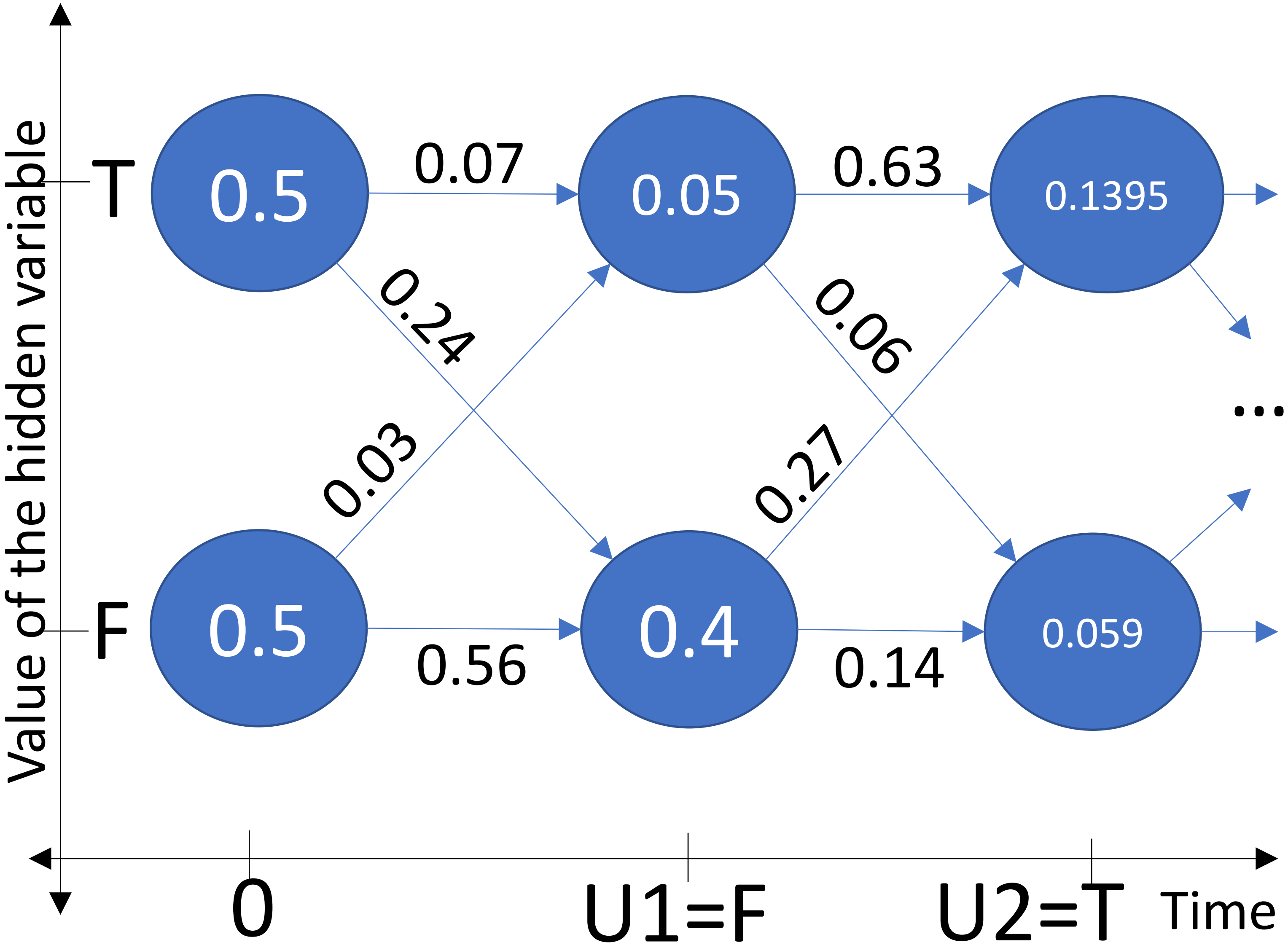


Forward Algorithm: The Trellis

- v_{it} = value of i[th] node at time t
- e_{ijt} = edge connecting node $v_{i,t-1}$ to v_{jt}

Forward algorithm is just:

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



Example: Filtering in UmbrellaWorld

Iterate $t = 2$:

Multiply:

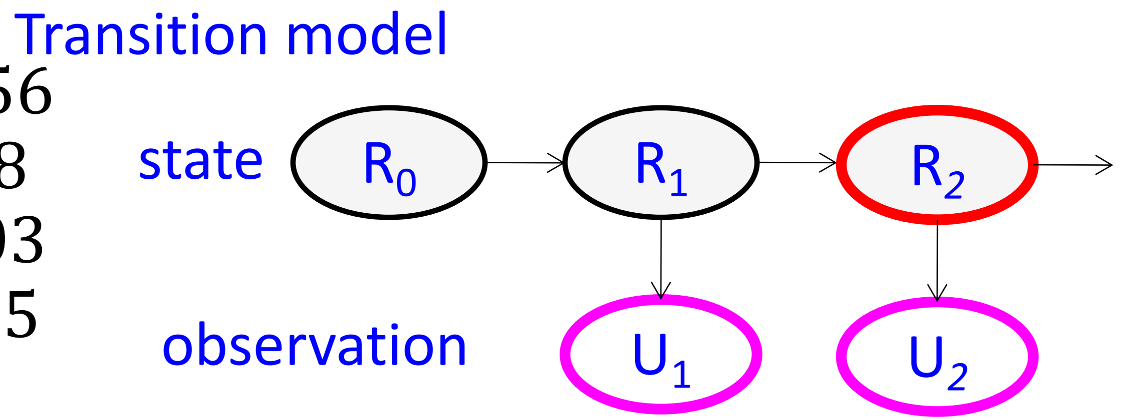
$$P(\neg R_1, \neg R_2, \neg U_1, U_2) = (0.4)(0.7)(0.2) = 0.056$$
$$P(\neg R_1, R_2, \neg U_1, U_2) = (0.4)(0.3)(0.9) = 0.108$$
$$P(R_1, \neg R_2, \neg U_1, U_2) = (0.05)(0.3)(0.2) = 0.003$$
$$P(R_1, R_2, \neg U_1, U_2) = (0.05)(0.7)(0.9) = 0.0315$$

Add:

$$P(\neg R_2, \neg U_1, U_2) = 0.056 + 0.003 = 0.059$$
$$P(R_2, \neg U_1, U_2) = 0.108 + 0.0315 = 0.1395$$

Terminate:

$$P(R_2 | \neg U_1, U_2) = \frac{0.1395}{0.1395 + 0.059} = 0.7$$



Transition probabilities

	R_t = T	R_t = F
R_{t-1} = T	0.7	0.3
R_{t-1} = F	0.3	0.7

Observation probabilities

	U_t = T	U_t = F
R_t = T	0.9	0.1
R_t = F	0.2	0.8

Forward Algorithm: The Trellis

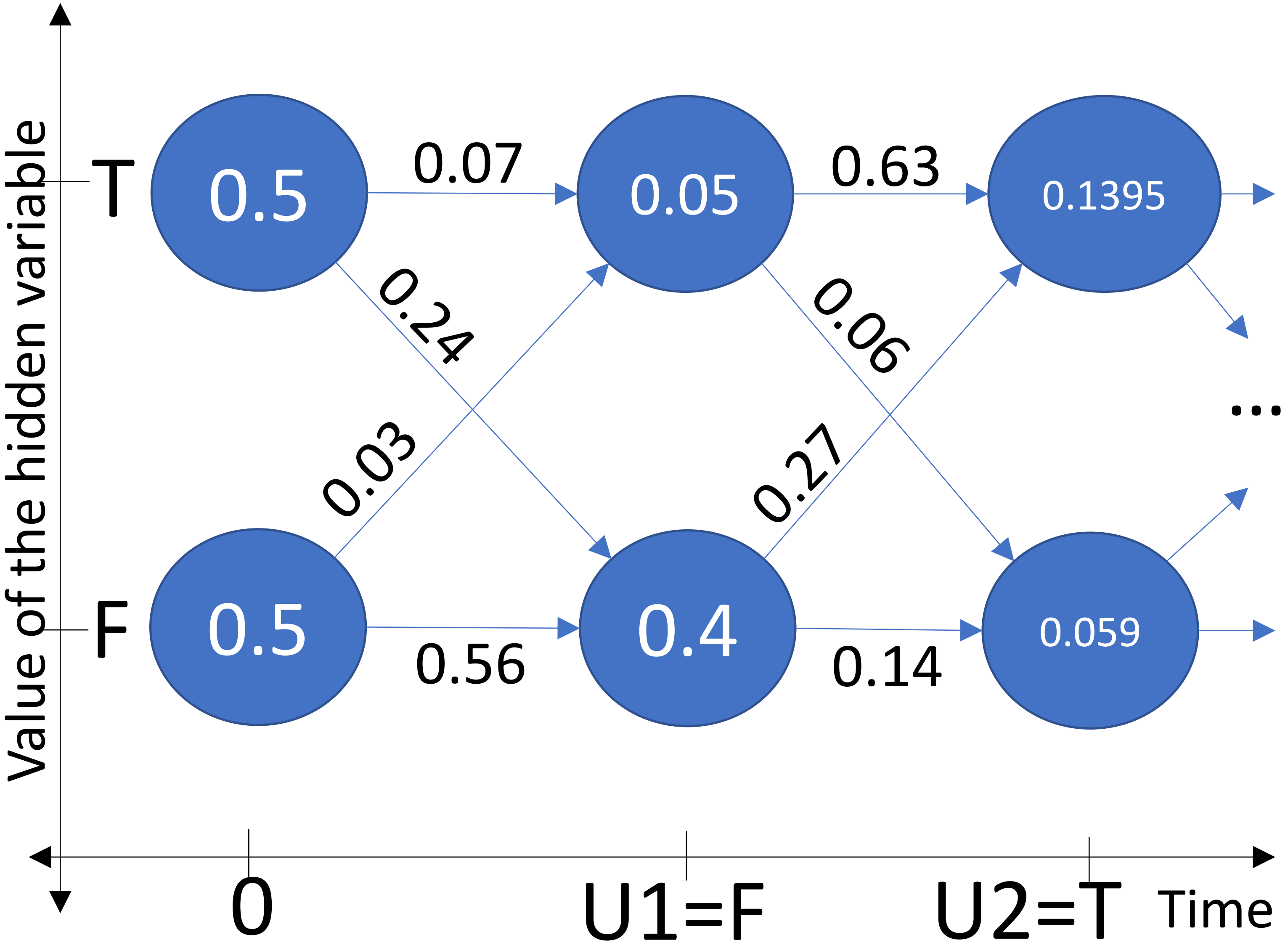
- v_{it} = value of i^th node at time t
- e_{ijt} = edge connecting node $v_{i,t-1}$ to v_{jt}

Forward algorithm is just:

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

<u>**Terminate**</u>:

$$P(R_2 | \neg U_1, U_2) = \frac{0.1395}{0.1395 + 0.059} = 0.7$$

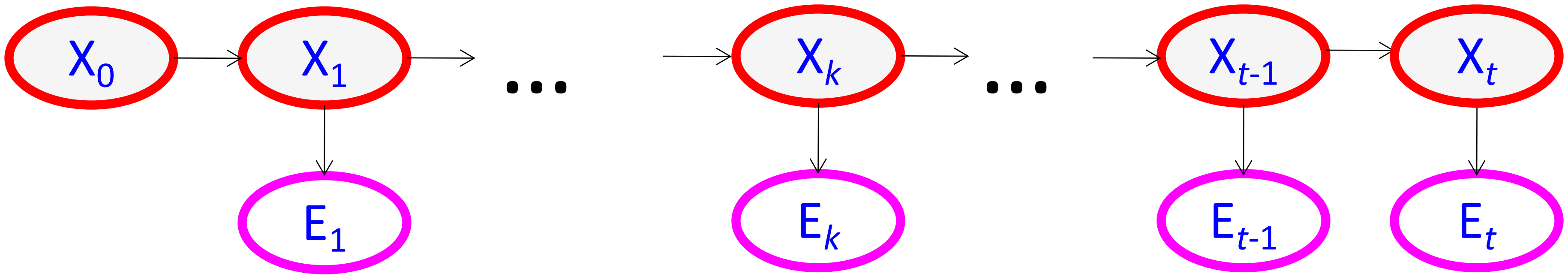


HMM inference tasks

- **Filtering:** what is the distribution over the current state X_t given all the evidence so far, $\mathbf{E}_{1:t}$ **--- use the Forward Algorithm, complexity $\mathcal{O}\{N^2T\}$**
- **Smoothing:** what is the distribution of some state X_k (k<t) given the entire observation sequence $\mathbf{E}_{1:t}$? **--- Forward-Backward Algorithm**
- **Evaluation:** compute the probability of a given observation sequence $\mathbf{E}_{1:t}$ **--- Forward Algorithm computes this!**

SOS

- **Decoding:** what is the most likely state sequence $\mathbf{X}_{0:t}$ given the observation sequence $\mathbf{E}_{1:t}$? (example: what's the weather every day?) **--- let's solve this problem next.**



Outline

- Inference by Enumeration in an HMM
- Filtering using the Forward Algorithm
- Decoding using the Viterbi Algorithm

Forward Algorithm vs. Viterbi Algorithm

- Forward Algorithm
  - Goal: efficiently compute $\mathrm{P}(R_T|U_1,\ldots,U_T)$
  - Complexity $\mathcal{O}\{N^2T\}$
  - Key equation: $v_{jt} = \sum_i v_{i,t-1}e_{ijt}$
- Viterbi Algorithm
  - Goal: efficiently find the values of
    $$R_0^*,\ldots,R_T^* = \operatorname{argmax} \mathrm{P}(R_0,\ldots,R_T|U_1,\ldots,U_T)$$
  - Complexity $\mathcal{O}\{N^2T\}$
  - Key equation: $v_{jt} = \max_i v_{i,t-1}e_{ijt}$
  - Back-pointer: $i^*(j,t) = \operatorname*{argmax}_i v_{i,t-1}e_{ijt}$

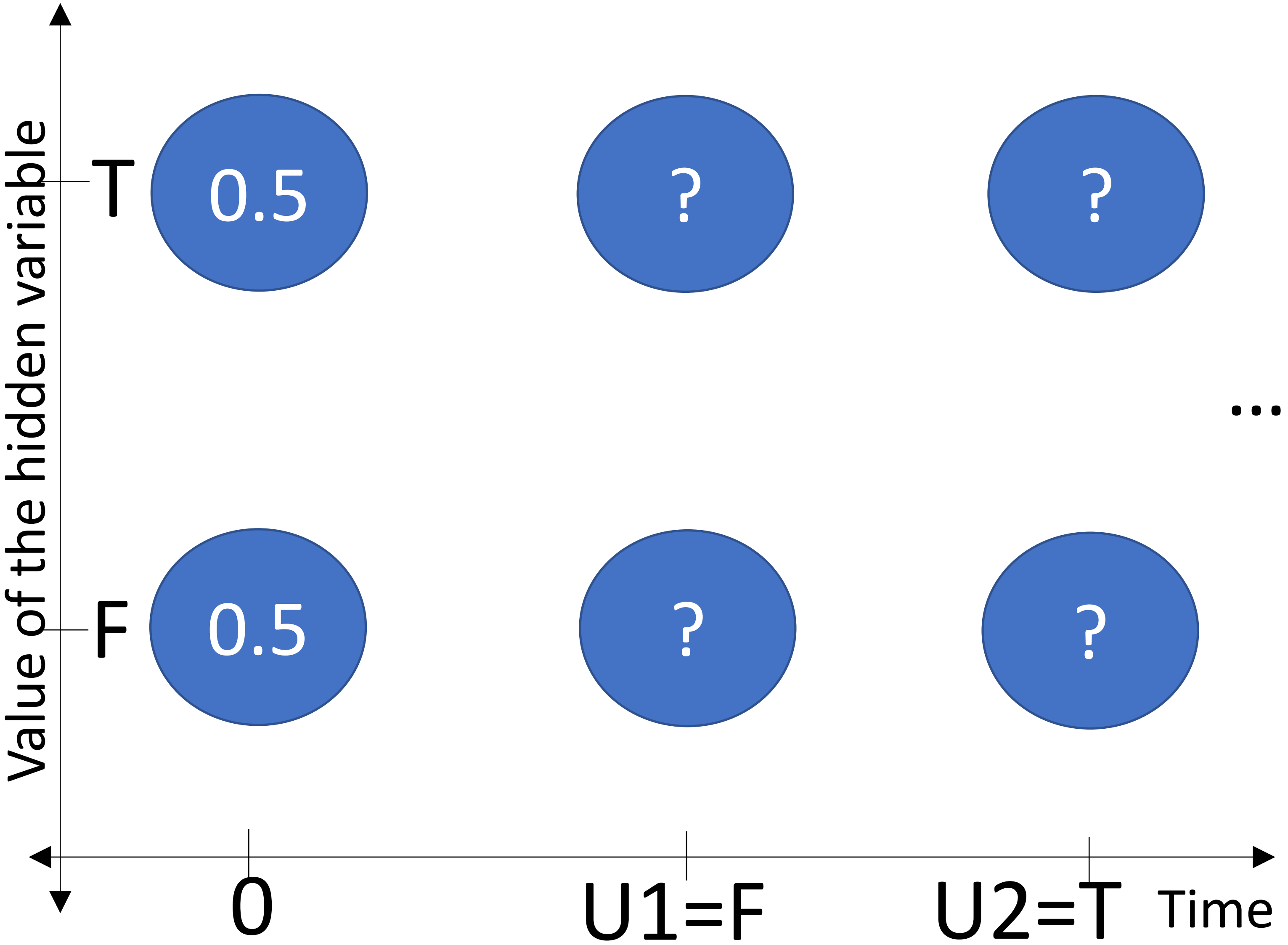
Viterbi Algorithm: Key concepts

- Goal: efficiently find the values of
$$R_0^*, \dots, R_T^* = \mathrm{argmax}\ \mathrm{P}(R_0, \dots, R_T | U_1, \dots, U_T)$$
- To do that efficiently, we need to keep track of TWO pieces of information at each node v_{jt}:
  - **Path Cost**: Probability of the best path until node j at time t
$$v_{jt} = \max_i \max_{R_0, \dots, R_{t-2}} P(R_0, U_1, R_1, \dots, U_{t-1}, R_{t-1} = i, U_t)$$
  - **Backpointer**: which node, i, precedes node j on the best path?
$$i^*(j,t) = \mathrm{argmax}_i \max_{R_0, \dots, R_{t-2}} P(R_0, U_1, R_1, \dots, U_{t-1}, R_{t-1} = i, U_t)$$

Viterbi Algorithm: The Trellis

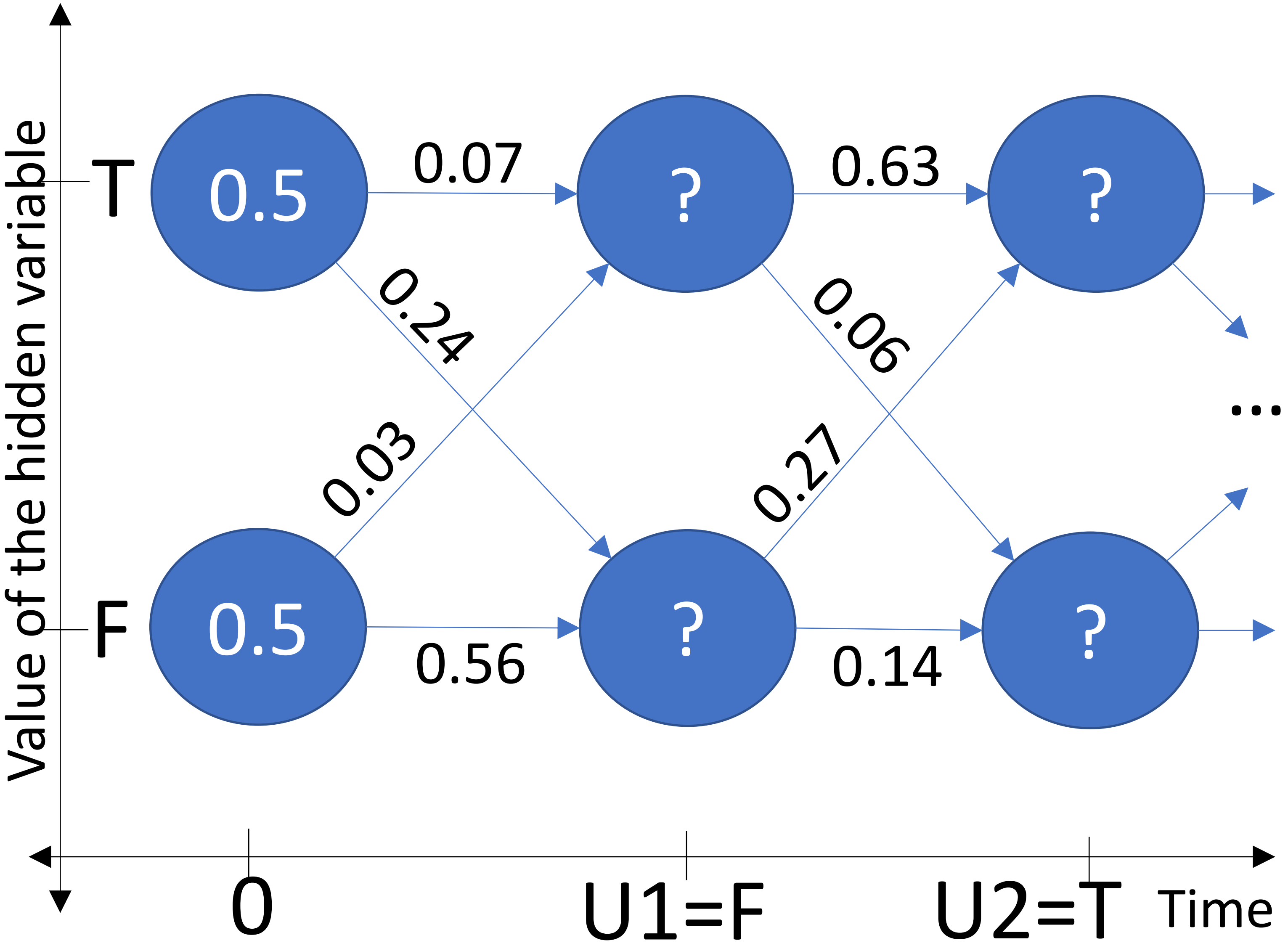
We can visualize the Viterbi algorithm using a TRELLIS:

- Node = a value of the hidden variable at a given time
- Numerical value of the node = probability that the hidden variable takes that value



Viterbi Algorithm: The Trellis

- Edge = a possible transition from R_{t-1} to R_t
- Numerical value of the edge = $\mathrm{P}(R_t|R_{t-1})\mathrm{P}(U_t|R_t)$



Viterbi Algorithm: The Trellis

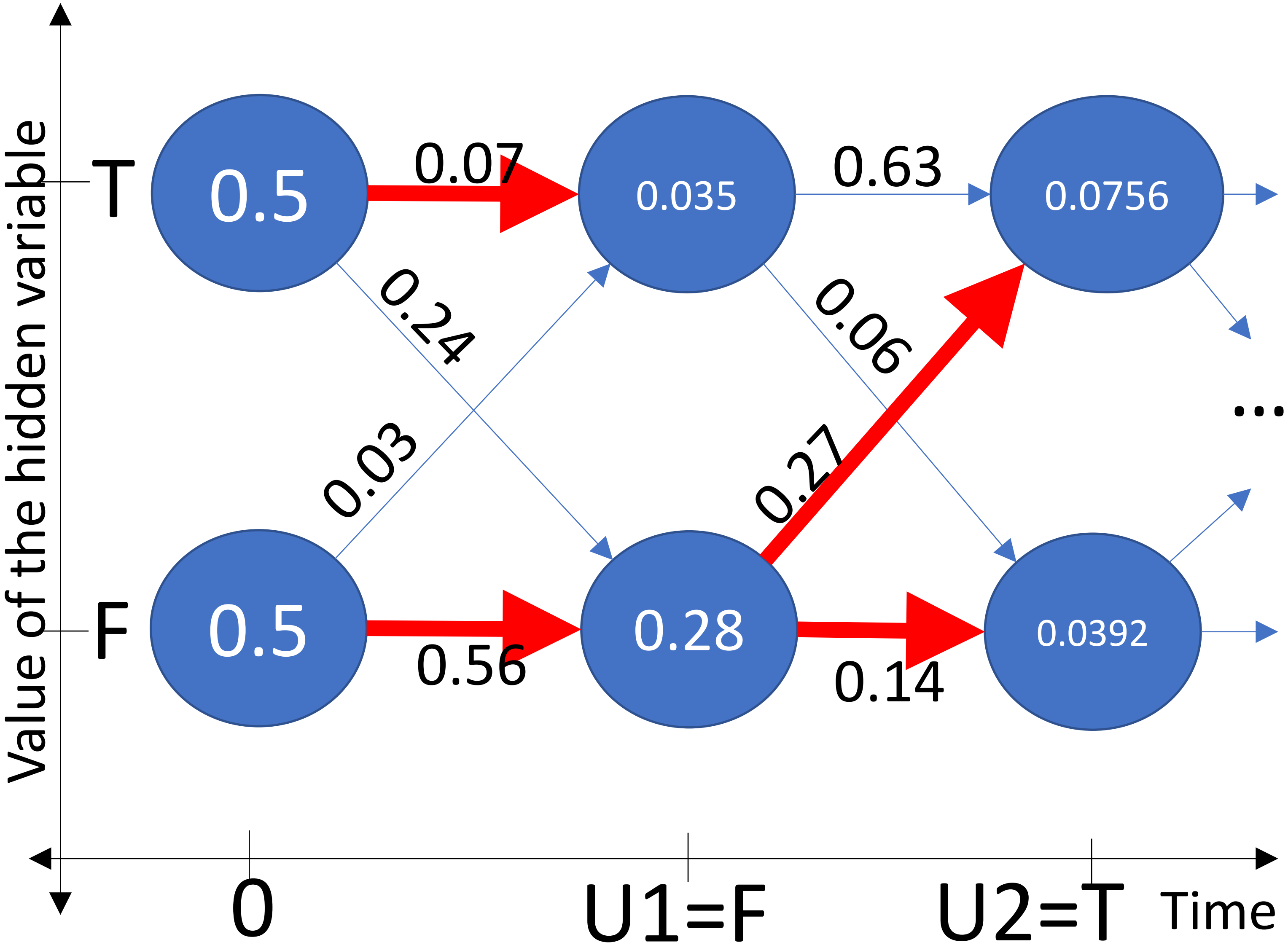
- v_{it} = value of i[th] node at time t
- e_{ijt} = edge connecting node $v_{i,t-1}$ to v_{jt}

Viterbi algorithm is:

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

Backpointer is:

$$i^*(j,t) = \underset{i}{\operatorname{argmax}}\, v_{i,t-1} e_{ijt}$$



Viterbi Algorithm: Termination

- Choose the node with the largest final value

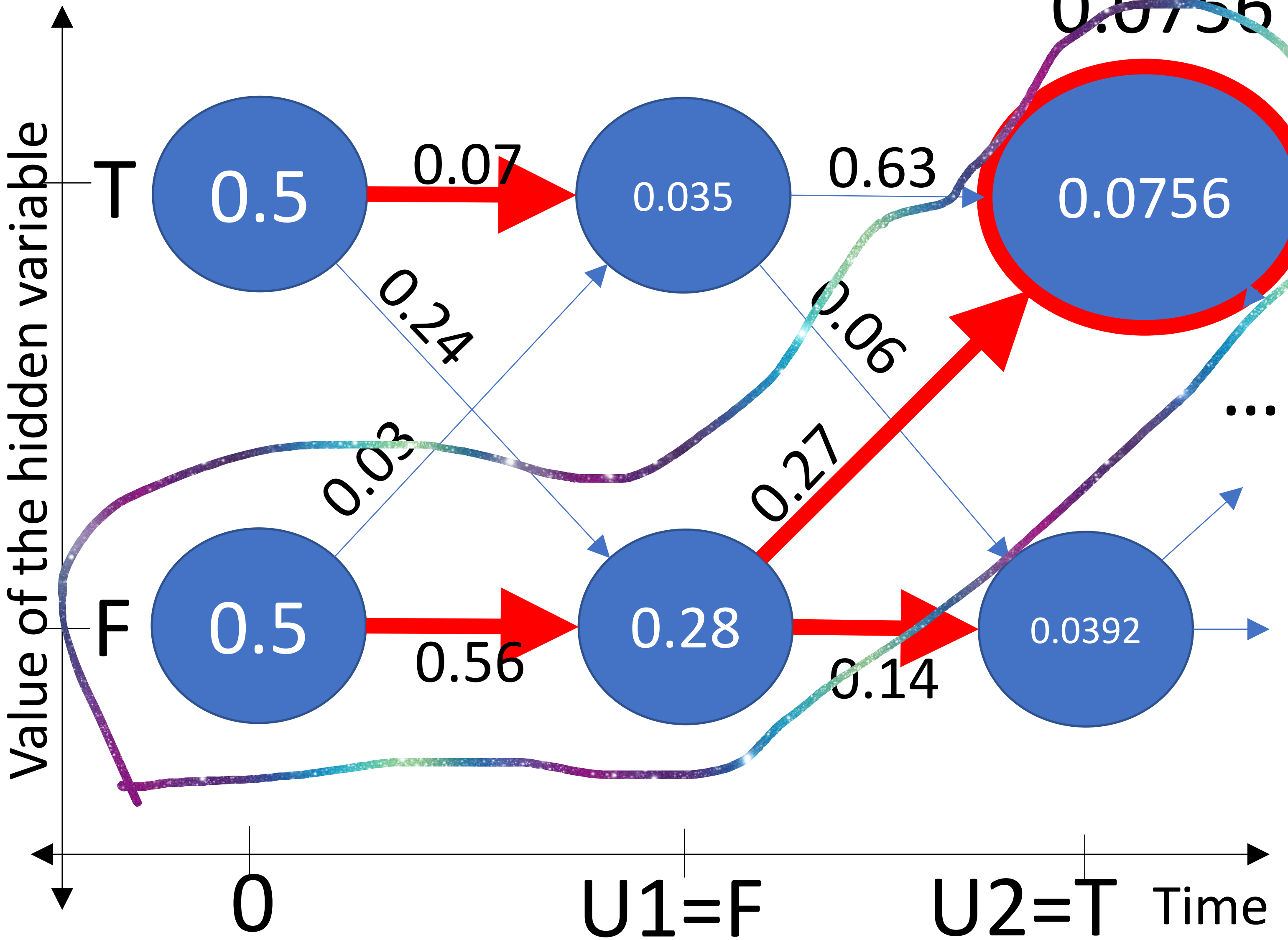
$$\max_{j} v_{jT} = \max_{R_0,\ldots,R_T} P(R_0, U_1, \ldots, U_T, R_T)$$

- Trace its backpointers to find

$$R_0^*, \ldots, R_T^*$$

Viterbi Algorithm: Termination

Best path probability = 0.0756



Best path: $\neg R_0 \neg R_1 R_2$

HMM inference tasks

- **Filtering:** what is the distribution over the current state X_t given all the evidence so far, $\mathbf{E}_{1:t}$ **--- use the Forward Algorithm, complexity $\mathcal{O}\{N^2T\}$**
- **Smoothing:** what is the distribution of some state X_k (k<t) given the entire observation sequence $\mathbf{E}_{1:t}$? **--- Forward-Backward Algorithm**
- **Evaluation:** compute the probability of a given observation sequence $\mathbf{E}_{1:t}$ **--- Forward Algorithm computes this!**
- **Decoding:** what is the most likely state sequence $\mathbf{X}_{0:t}$ given the observation sequence $\mathbf{E}_{1:t}$? (example: what's the weather every day?) **--- use the Viterbi Algorithm, complexity $\mathcal{O}\{N^2T\}$**