TTIC 31110
Speech Technologies

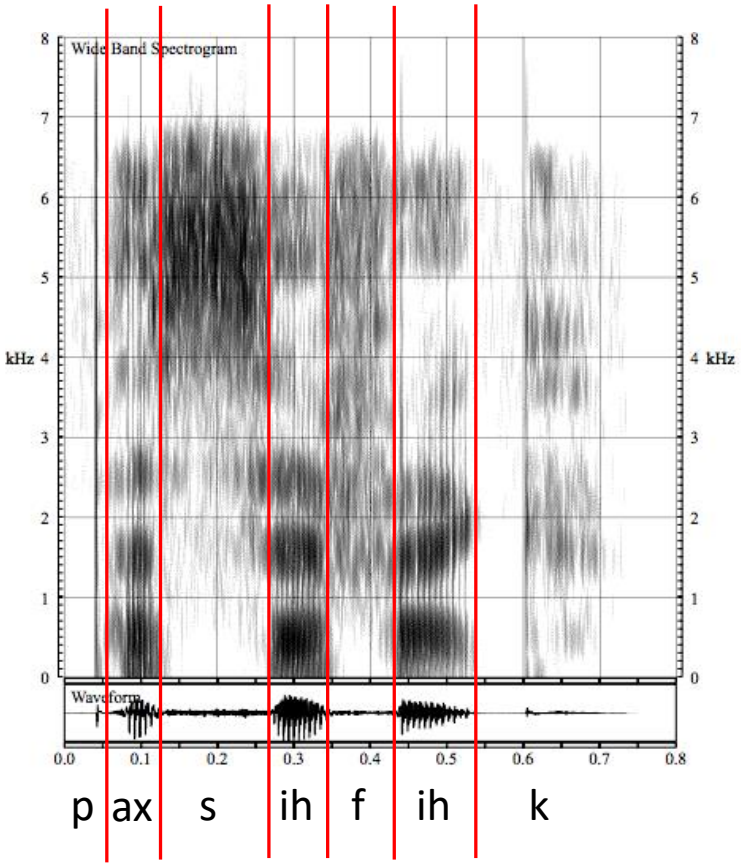
April 23, 2020

Announcements

- Gaussian mixture + HMM readings posted; see optional reading for EM proofs
- HW2 due April 27 7pm; submission via PDF + iPython notebook
- Please let us know ASAP of any issues with compute resources!

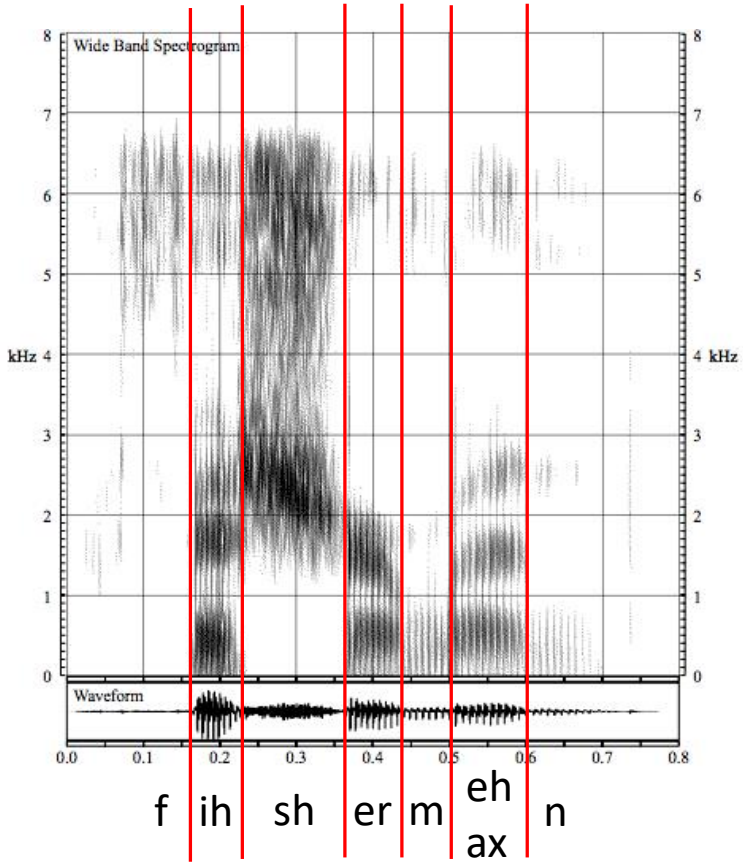
HW1: Solution to Q2

(i) “pacific”



HW1: Solution to Q2

(ii) “fisherman/fishermen”



Questions from last time

Connection between DTW with Euclidean distance and Gaussian distributions. Recall: log density of a Gaussian is
$\ln p(\mathbf{x}|\mu, \mathbf{\Sigma}) = \text{const.} - \frac{1}{2}(\mathbf{x} - \mu)^T \mathbf{\Sigma^{-1}} (\mathbf{x} - \mu)$

- If $\Sigma = \mathbf{I}$, then this equals

$$\text{const.} - \frac{1}{2}(\mathbf{x} - \mu)^T (\mathbf{x} - \mu) = \sum_{d=1}^{D} (x_d - \mu_d)^2$$

- $=$ almost the squared Euclidean distance between $\mathbf{x}$ and μ
- Now in DTW, define the distance function between a test frame and a template frame to be
  $d(\mathbf{x}_{\text{test}}, \mathbf{x}_{\text{template}}) = \ln p(\mathbf{x}_{\text{test}}|\mathbf{x}_{\text{template}}, \mathbf{I})$
- Extension to Gaussian mixtures: Each Gaussian corresponds to a different template of the same word (just a thought experiment! In practice Gaussian mixtures are used with HMMs, not DTW)

Recap: The expectation-maximization (EM) algorithm

- Initialization: Guess θ, π
- Iterate:

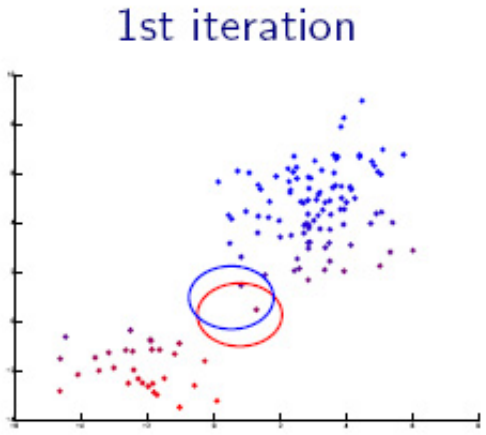
  E-step: Compute γ_{ik} using current estimates of θ, π

  M-step: Estimate new parameters, maximizing the expected likelihood, given the current γ_{ik}

- Until log likelihood converges

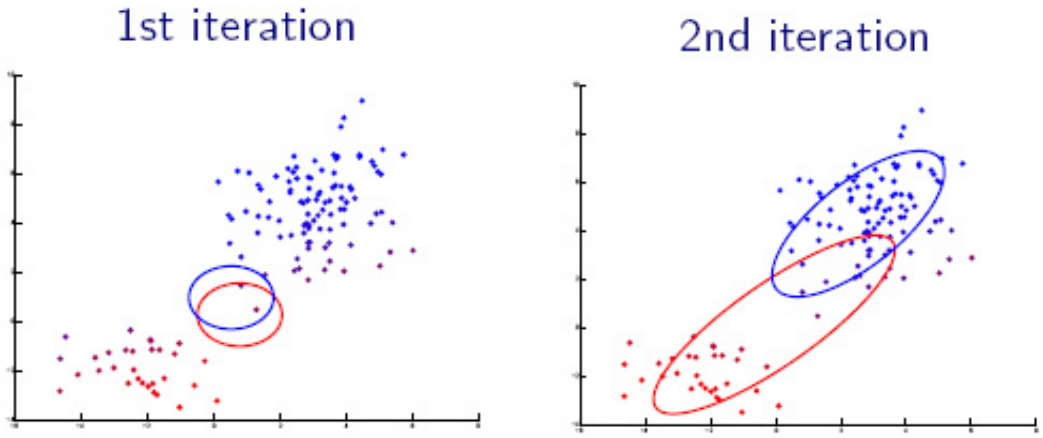
Recap: EM for Gaussian mixtures

Colors represent γ_{ik} after the E-step



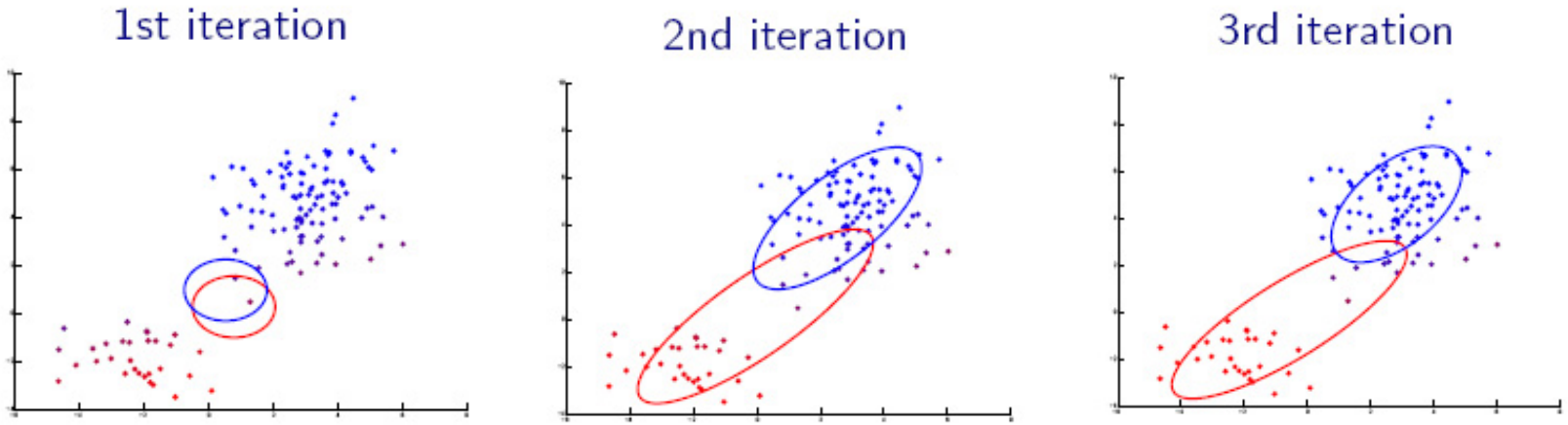
Recap: EM for Gaussian mixtures

Colors represent γ_{ik} after the E-step



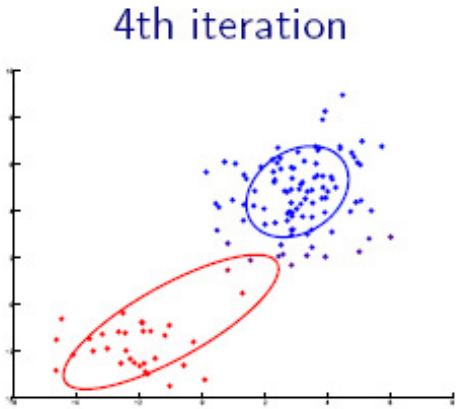
Recap: EM for Gaussian mixtures

Colors represent γ_{ik} after the E-step



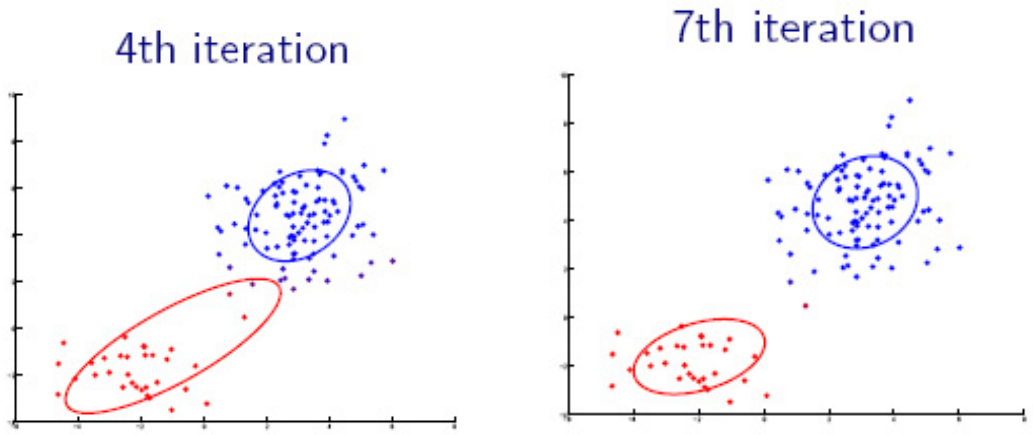
Recap: EM for Gaussian mixtures

Colors represent γ_{ik} after the E-step



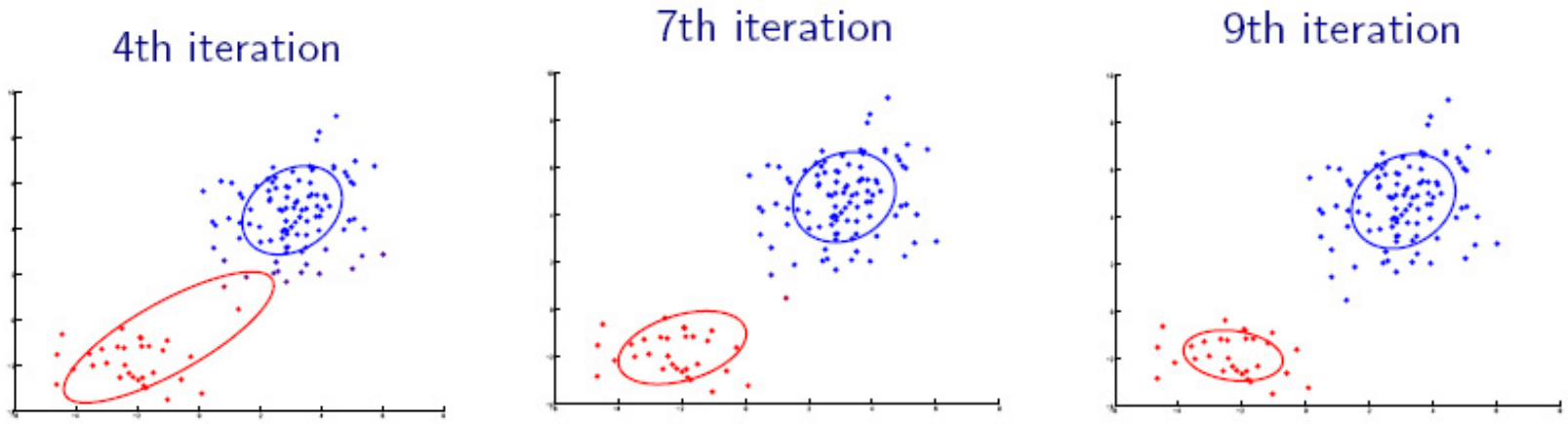
Recap: EM for Gaussian mixtures

Colors represent γ_{ik} after the E-step



Recap: EM for Gaussian mixtures

Colors represent γ_{ik} after the E-step



Recap: Convergence of EM

- Let $\ell^{(t)}$ be $\ln p(\mathbf{X}|\hat{\mu}, \hat{\mathbf{\Sigma}}, \hat{\pi})$ after t iterations
- Can show:

  $$\ell^{(0)} \leq \ell^{(1)} \leq \ldots \leq \ell^{(t)} \ldots$$

- EM converges, but possibly to a local maximum of the likelihood
- One solution (as in k-means): Use multiple initializations and pick the result with highest likelihood
- Note: EM converges if there exists a finite ML solution, *but there may not be one (e.g. for Gaussian mixtures)*

Aside: Connection between EM and gradient descent

Xu and Jordan, "On Convergence Properties of the EM Algorithm for Gaussian Mixtures," *Neural Computation* **8**(1)129-151, 1996.

- EM update can be obtained from the gradient via a projection matrix
- Convergence rates of EM vs. gradient methods can depend on shapre of the likelihood function
- Advantages of EM: No learning rate to tune, no need to add constraints

Outline

Hidden Markov models (HMMs)

- Can serve as alternative to "templates" and "path weights" in DTW
- Ubiquitous model for speech recognition
- Today and next time:
  - Introduction to HMMs
  - Solving the 3 Problems: Scoring, decoding, training
  - Implementation issues, extensions

Hidden Markov models (HMMs): An example

- Your friend is in a distant city. You talk with him once per day and assess his mood. His mood depends on the weather at his location.
- 3 weather conditions (*states*): Fair (sunny), Cloudy, Rainy.
- 2 moods (i.e. observations): Happy, Unhappy
- Example observation sequence for 4 days: $\mathbf{O} = HHUH$

Hidden Markov models (HMMs): An example

- Your friend is in a distant city. You talk with him once per day and assess his mood. His mood depends on the weather at his location.
- 3 weather conditions (*states*): Fair (sunny), Cloudy, Rainy.
- 2 moods (i.e. observations): Happy, Unhappy
- Example observation sequence for 4 days: $\mathbf{O} = HHUH$
- The 3 problems:
  1. What is the probability of this sequence? (the *scoring* problem)
  2. What is your best guess of the sequence of weather states, $q_1, \ldots, q_4$? (the *decoding* problem)
  3. Given a large number of observations, how could you learn a model of the weather-mood system? (the *training* problem)

HMMs: An example (2)

Modeling the weather-mood system as a hidden Markov model:

HMMs: An example (2)

Modeling the weather-mood system as a hidden Markov model:

- On the first day, the *a priori* probabilities of the three weather states F, C, R are $\{0.4, 0.3, 0.3\}$, respectively

HMMs: An example (2)

Modeling the weather-mood system as a hidden Markov model:

- On the first day, the *a priori* probabilities of the three weather states F, C, R are $\{0.4, 0.3, 0.3\}$, respectively
- On any other day t, the weather state q_t depends on the previous day's weather (and nothing else), according to $P(q_{t+1}|q_t)$
  - I.e., the state sequence is a *Markov chain*
  - It is *hidden* since you don't observe the weather.

HMMs: An example (2)

Modeling the weather-mood system as a hidden Markov model:

- On the first day, the *a priori* probabilities of the three weather states F, C, R are $\{0.4, 0.3, 0.3\}$, respectively
- On any other day t, the weather state q_t depends on the previous day's weather (and nothing else), according to $P(q_{t+1}|q_t)$
  - I.e., the state sequence is a *Markov chain*
  - It is *hidden* since you don't observe the weather.
- Each day, your friend's mood o_t depends probabilistically on that day's weather (and on nothing else), according to $P(o_t|q_t)$

HMMs: An example (2)

Modeling the weather-mood system as a hidden Markov model:

- On the first day, the *a priori* probabilities of the three weather states F, C, R are $\{0.4, 0.3, 0.3\}$, respectively
- On any other day t, the weather state q_t depends on the previous day's weather (and nothing else), according to $P(q_{t+1}|q_t)$
  - I.e., the state sequence is a *Markov chain*
  - It is *hidden* since you don't observe the weather.
- Each day, your friend's mood o_t depends probabilistically on that day's weather (and on nothing else), according to $P(o_t|q_t)$

$P(o_t \mid q_t)$	$o_t = H$	$o_t = U$
$q_t = F$	0.9	0.1
$q_t = C$	0.5	0.5
$q_t = R$	0.2	0.8

$P(q_{t+1} \mid q_t)$	$q_{t+1} = F$	$q_{t+1} = C$	$q_{t+1} = R$
$q_t = F$	0.8	0.2	0
$q_t = C$	0.3	0.4	0.3
$q_t = R$	0	0.3	0.7

Elements of a (discrete) HMM

- N: Number of states; state at time t: $q_t \in \{1, \ldots, N\}$
- $V = \{v_1, \ldots, v_M\}$: Set of M possible observation labels (or vectors, in general); observation at time t: $o_t \in V$
- $\pi = \{\pi_i\}$: Initial state distribution,
  $\pi_i = P(q_1 = i), \;\; 1 \leq i \leq N$
- $\mathbf{A} = \{a_{ij}\}$: $N \times N$ state transition probability matrix,
  $a_{ij} = P(q_{t+1} = j | q_t = i), \;\; 1 \leq i, j, \leq N$
- $\mathbf{B} = \{b_i(k)\}$: Observation (or *emission*) distribution in state i,
  $b_i(k) = P(o_t = v_k | q_t = i), \;\; 1 \leq i \leq N, 1 \leq k \leq M$

The entire model can be denoted $\lambda = \{\mathbf{A}, \mathbf{B}, \pi\}$

Elements of the weather-mood HMM

- States: $N = 3$; let state 1 be F, state 2 be C, state 3 be R
- Obervation labels: $M = 2$; let $v_1 = H, v_2 = U$
- Model probabilities $\lambda = \{\mathbf{A}, \mathbf{B}, \pi\}$:

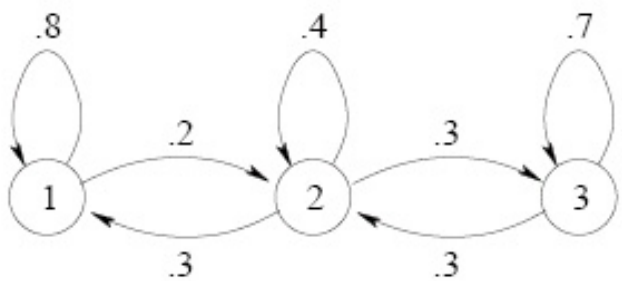
$$\pi = [0.4\ 0.3\ 0.3], \quad A = \begin{bmatrix} 0.8 & 0.2 & 0 \\ 0.3 & 0.4 & 0.3 \\ 0 & 0.3 & 0.7 \end{bmatrix}, \quad B = \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \\ 0.2 & 0.8 \end{bmatrix}$$

Elements of the weather-mood HMM

- States: $N = 3$; let state 1 be F, state 2 be C, state 3 be R
- Obervation labels: $M = 2$; let $v_1 = H, v_2 = U$
- Model probabilities $\lambda = \{\mathbf{A}, \mathbf{B}, \pi\}$:

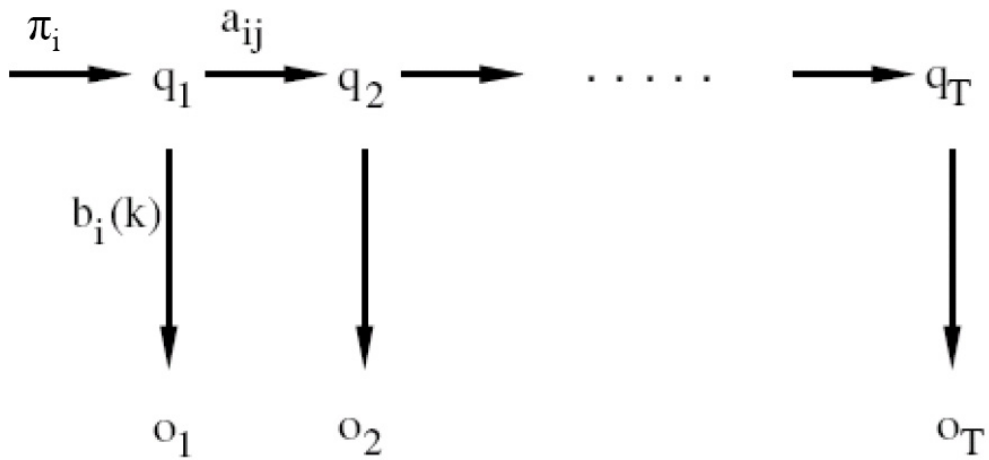
$$\pi = [0.4\ 0.3\ 0.3], \quad A = \begin{bmatrix} 0.8 & 0.2 & 0 \\ 0.3 & 0.4 & 0.3 \\ 0 & 0.3 & 0.7 \end{bmatrix}, \quad B = \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \\ 0.2 & 0.8 \end{bmatrix}$$

- $\mathbf{A}$ can also be represented via a *state transition diagram*:



Generation of observations from an HMM

1. Draw an initial state $q_1 = i$ from the initial distribution π
2. For $t = 1$ to T:
   - Choose $o_t = v_k$ according to state i's observation distribution $b_i(k)$
   - Choose a new state $q_{t+1} = j$ according to state i's transition probabilities a_{ij}. Let $i \leftarrow j$



Notation notes

- Some formulations also allow for states that emit no (null) observations
- In some formulations (e.g. much of the original HMM work), observation are generated at *transitions* rather than in states
  - Note: There is a one-to-one mapping between HMMs with state-generated and transition-generated observations

The three basic HMM problems

The three basic HMM problems

1. *Scoring*: Given an observation sequence $\mathbf{O} = \{\mathbf{o}_1, \ldots, \mathbf{o}_T\}$ and a model $\lambda = \{\mathbf{A}, \mathbf{B}, \pi\}$, how do we compute $P(\mathbf{O}|\lambda)$, the probability of the observation sequence?
   $\longrightarrow$ The forward & backward algorithms

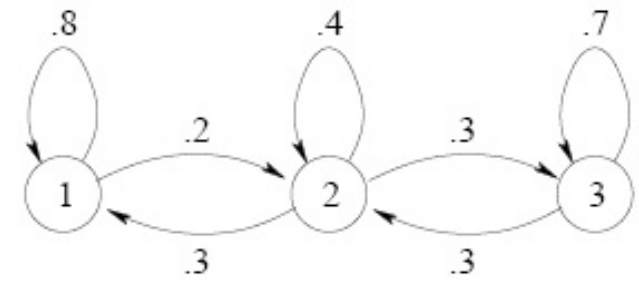
The three basic HMM problems

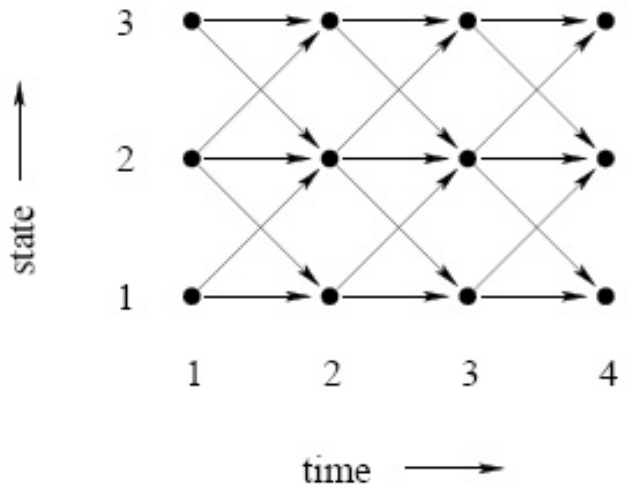
1. *Scoring*: Given an observation sequence $\mathbf{O} = \{\mathbf{o}_1, \ldots, \mathbf{o}_T\}$ and a model $\lambda = \{\mathbf{A}, \mathbf{B}, \pi\}$, how do we compute $P(\mathbf{O}|\lambda)$, the probability of the observation sequence?
   $\longrightarrow$ The forward & backward algorithms
2. *Decoding*: Given an observation sequence $\mathbf{O} = \{o_1, \ldots, o_T\}$, how do we choose the state sequence $\mathbf{q} = \{q_1, \ldots, q_T\}$ most likely to have generated the observations?
   $\longrightarrow$ The Viterbi algorithm

The three basic HMM problems

1. *Scoring*: Given an observation sequence $\mathbf{O} = \{\mathbf{o}_1, \ldots, \mathbf{o}_T\}$ and a model $\lambda = \{\mathbf{A}, \mathbf{B}, \pi\}$, how do we compute $P(\mathbf{O}|\lambda)$, the probability of the observation sequence?
   $\longrightarrow$ The forward & backward algorithms
2. *Decoding*: Given an observation sequence $\mathbf{O} = \{o_1, \ldots, o_T\}$, how do we choose the state sequence $\mathbf{q} = \{q_1, \ldots, q_T\}$ most likely to have generated the observations?
   $\longrightarrow$ The Viterbi algorithm
3. *Training*: Given a training set of observations $\mathbf{O}$, how do we set the model parameters $\lambda = \{\mathbf{A}, \mathbf{B}, \pi\}$ to maximize $P(\mathbf{O}|\lambda)$ (maximum-likelihood estimation)
   $\longrightarrow$ The Baum-Welch algorithm (EM applied to HMMs)

Representing transition diagram with a trellis





Useful representation for visualizing HMM algorithms

Outline

Problem 1: Computing $p(\mathbf{O}|\lambda)$

$$
\begin{aligned}
p(\mathbf{O}|\lambda) &= \sum_{\text{all } \mathbf{q}} p(\mathbf{O}, \mathbf{q}|\lambda) \\
p(\mathbf{O}, \mathbf{q}|\lambda) &= p(\mathbf{O}|\mathbf{q}, \lambda) p(\mathbf{q}|\lambda)
\end{aligned}
$$

Problem 1: Computing $p(\mathbf{O}|\lambda)$

$$
\begin{aligned}
p(\mathbf{O}|\lambda) &= \sum_{\text{all } \mathbf{q}} p(\mathbf{O}, \mathbf{q}|\lambda) \\
p(\mathbf{O}, \mathbf{q}|\lambda) &= p(\mathbf{O}|\mathbf{q}, \lambda) p(\mathbf{q}|\lambda)
\end{aligned}
$$

- Consider some *fixed* state sequence $\mathbf{q} = q_1 q_2 \ldots q_T$. Then:

Problem 1: Computing $p(\mathbf{O}|\lambda)$

$$
\begin{aligned}
p(\mathbf{O}|\lambda) &= \sum_{\text{all } \mathbf{q}} p(\mathbf{O}, \mathbf{q}|\lambda) \\
p(\mathbf{O}, \mathbf{q}|\lambda) &= p(\mathbf{O}|\mathbf{q}, \lambda) p(\mathbf{q}|\lambda)
\end{aligned}
$$

- Consider some *fixed* state sequence $\mathbf{q} = q_1 q_2 \ldots q_T$. Then:

$$
\begin{aligned}
p(\mathbf{O}|\mathbf{q}, \lambda) &= b_{q_1}(\mathbf{o}_1) b_{q_2}(\mathbf{o}_2) \ldots b_{q_T}(\mathbf{o}_T) \\
p(\mathbf{q}|\lambda) &= \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \ldots a_{q_{T-1} q_T}
\end{aligned}
$$

Problem 1: Computing $p(\mathbf{O}|\lambda)$

$$\begin{aligned} p(\mathbf{O}|\lambda) &= \sum_{\text{all } \mathbf{q}} p(\mathbf{O}, \mathbf{q}|\lambda) \\ p(\mathbf{O}, \mathbf{q}|\lambda) &= p(\mathbf{O}|\mathbf{q}, \lambda) p(\mathbf{q}|\lambda) \end{aligned}$$

- Consider some *fixed* state sequence $\mathbf{q} = q_1 q_2 \dots q_T$. Then:

$$\begin{aligned} p(\mathbf{O}|\mathbf{q}, \lambda) &= b_{q_1}(\mathbf{o}_1) b_{q_2}(\mathbf{o}_2) \dots b_{q_T}(\mathbf{o}_T) \\ p(\mathbf{q}|\lambda) &= \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \dots a_{q_{T-1} q_T} \end{aligned}$$

- Therefore,

$$p(\mathbf{O}|\lambda) = \sum_{q_1, q_2, \dots, q_T} \pi_{q_1} b_{q_1}(\mathbf{o}_1) a_{q_1 q_2} b_{q_2}(\mathbf{o}_2) \dots a_{q_{T-1} q_T} b_{q_T}(\mathbf{o}_T)$$

Problem 1: Computing $p(\mathbf{O}|\lambda)$

$$\begin{aligned} p(\mathbf{O}|\lambda) &= \sum_{\text{all } \mathbf{q}} p(\mathbf{O}, \mathbf{q}|\lambda) \\ p(\mathbf{O}, \mathbf{q}|\lambda) &= p(\mathbf{O}|\mathbf{q}, \lambda) p(\mathbf{q}|\lambda) \end{aligned}$$

- Consider some *fixed* state sequence $\mathbf{q} = q_1 q_2 \dots q_T$. Then:

$$\begin{aligned} p(\mathbf{O}|\mathbf{q}, \lambda) &= b_{q_1}(\mathbf{o}_1) b_{q_2}(\mathbf{o}_2) \dots b_{q_T}(\mathbf{o}_T) \\ p(\mathbf{q}|\lambda) &= \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \dots a_{q_{T-1} q_T} \end{aligned}$$

- Therefore,
$$p(\mathbf{O}|\lambda) = \sum_{q_1, q_2, \dots, q_T} \pi_{q_1} b_{q_1}(\mathbf{o}_1) a_{q_1 q_2} b_{q_2}(\mathbf{o}_2) \dots a_{q_{T-1} q_T} b_{q_T}(\mathbf{o}_T)$$

- N^T possible sequences $\mathbf{q} \Rightarrow$ computation $\approx 2T \cdot N^T$. For $N = 5, T = 100$, this is $2 \cdot 100 \cdot 5^{100} \approx 10^{72}$ computations!

Computing $p(O|\lambda)$: The forward algorithm

- A dynamic programming algorithm
- Define the "forward" variable $\alpha_t(i)$:
  $\alpha_t(i) = P(\mathbf{o}_1\mathbf{o}_2 \ldots \mathbf{o}_t, q_t = i|\lambda), \;\; 1 \leq t \leq T, 1 \leq i \leq N$

Computing $p(O|\lambda)$: The forward algorithm

- A dynamic programming algorithm
- Define the "forward" variable $\alpha_t(i)$:
  $\alpha_t(i) = P(\mathbf{o}_1\mathbf{o}_2 \ldots \mathbf{o}_t, q_t = i|\lambda), \quad 1 \leq t \leq T, 1 \leq i \leq N$
- For $t = 1$: $\alpha_1(i) = \pi_i b_i(\mathbf{o}_1), 1 \leq i \leq N$

Computing $p(O|\lambda)$: The forward algorithm

- A dynamic programming algorithm
- Define the "forward" variable $\alpha_t(i)$:
  $\alpha_t(i) = P(\mathbf{o}_1\mathbf{o}_2 \ldots \mathbf{o}_t, q_t = i|\lambda), \quad 1 \leq t \leq T, 1 \leq i \leq N$
- For $t = 1$: $\alpha_1(i) = \pi_i b_i(\mathbf{o}_1), 1 \leq i \leq N$
- For $t > 1$: Sum over all ways of getting to current state at time t:

$$\alpha_t(j) = \left[\sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij}\right] b_j(\mathbf{o}_t)$$

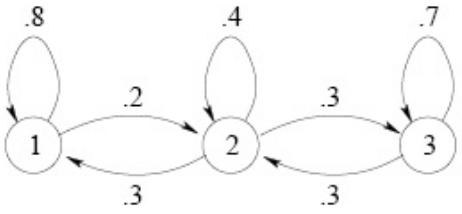
Computing $p(O|\lambda)$: The forward algorithm

- A dynamic programming algorithm
- Define the "forward" variable $\alpha_t(i)$:
  $\alpha_t(i) = P(\mathbf{o}_1\mathbf{o}_2 \ldots \mathbf{o}_t, q_t = i|\lambda), \quad 1 \leq t \leq T, 1 \leq i \leq N$
- For $t = 1$: $\alpha_1(i) = \pi_i b_i(\mathbf{o}_1), 1 \leq i \leq N$
- For $t > 1$: Sum over all ways of getting to current state at time t:
  $$\alpha_t(j) = \left[\sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij}\right] b_j(\mathbf{o}_t)$$
- Finally: $p(\mathbf{O}|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$

Computing $p(O|\lambda)$: The forward algorithm

- A dynamic programming algorithm
- Define the "forward" variable $\alpha_t(i)$:
  $\alpha_t(i) = P(\mathbf{o}_1\mathbf{o}_2 \ldots \mathbf{o}_t, q_t = i|\lambda), \quad 1 \leq t \leq T, 1 \leq i \leq N$
- For $t = 1$: $\alpha_1(i) = \pi_i b_i(\mathbf{o}_1), 1 \leq i \leq N$
- For $t > 1$: Sum over all ways of getting to current state at time t:
  $$\alpha_t(j) = \left[\sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij}\right] b_j(\mathbf{o}_t)$$
- Finally: $p(\mathbf{O}|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$
- Calculation is on the order of N^2T.
  $N = 5, T = 100 \Rightarrow \sim 2500$ computations, instead of 10^{72}

The forward algorithm: Example



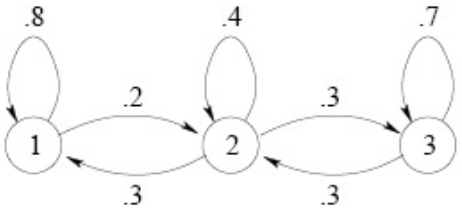
$$\pi = [0.4\ 0.3\ 0.3]$$

$$B = \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \\ 0.2 & 0.8 \end{bmatrix}$$

$$O = HHUH$$

i	$\alpha_1(i)$	$\alpha_2(i)$	$\alpha_3(i)$	$\alpha_4(i)$
3 (R)				
2 (C)				
1 (F)				
	H	H	U	H

The forward algorithm: Example



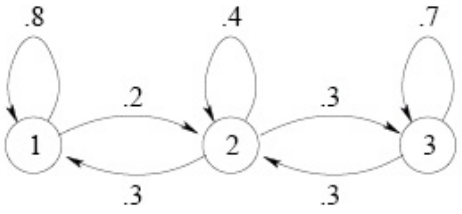
$$\pi = [0.4\ 0.3\ 0.3]$$

$$B = \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \\ 0.2 & 0.8 \end{bmatrix}$$

$$O = HHUH$$

i	$\alpha_1(i)$	$\alpha_2(i)$	$\alpha_3(i)$	$\alpha_4(i)$
3 (R)				
2 (C)				
1 (F)	.4			
	H	H	U	H

The forward algorithm: Example



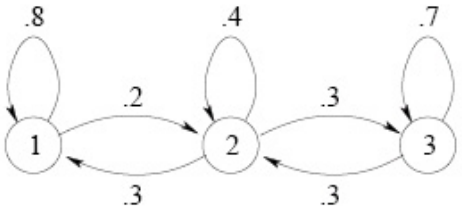
$$\pi = [0.4\ 0.3\ 0.3]$$

$$B = \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \\ 0.2 & 0.8 \end{bmatrix}$$

$$O = HHUH$$

i	$\alpha_1(i)$	$\alpha_2(i)$	$\alpha_3(i)$	$\alpha_4(i)$
3 (R)				
2 (C)				
1 (F)	$.4\times.9$			
	H	H	U	H

The forward algorithm: Example



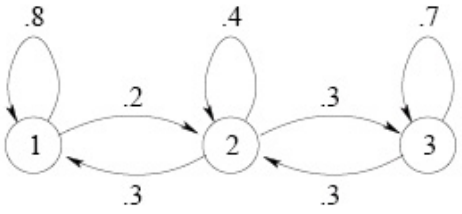
$$\pi = [0.4\ 0.3\ 0.3]$$

$$B = \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \\ 0.2 & 0.8 \end{bmatrix}$$

$$O = HHUH$$

i	$\alpha_1(i)$	$\alpha_2(i)$	$\alpha_3(i)$	$\alpha_4(i)$
3 (R)				
2 (C)				
1 (F)	$.4 \times .9 = .36$			
	H	H	U	H

The forward algorithm: Example



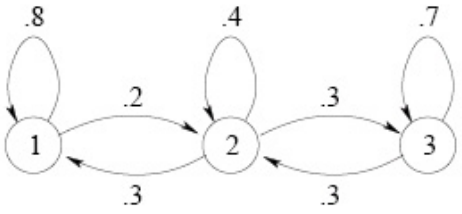
$\pi = [0.4\ 0.3\ 0.3]$

$$B = \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \\ 0.2 & 0.8 \end{bmatrix}$$

$O = HHUH$

i	$\alpha_1(i)$	$\alpha_2(i)$	$\alpha_3(i)$	$\alpha_4(i)$
3 (R)				
2 (C)	$.3 \times .5 = .15$			
1 (F)	$.4 \times .9 = .36$			
	H	H	U	H

The forward algorithm: Example



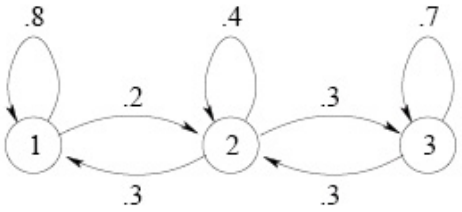
$$\pi = [0.4\ 0.3\ 0.3]$$

$$B = \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \\ 0.2 & 0.8 \end{bmatrix}$$

$$O = HHUH$$

i	$\alpha_1(i)$	$\alpha_2(i)$	$\alpha_3(i)$	$\alpha_4(i)$
3 (R)	$.3 \times .2 = .06$			
2 (C)	$.3 \times .5 = .15$			
1 (F)	$.4 \times .9 = .36$			
	H	H	U	H

The forward algorithm: Example



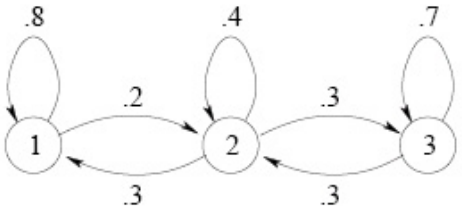
$$\pi = [0.4\ 0.3\ 0.3]$$

$$B = \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \\ 0.2 & 0.8 \end{bmatrix}$$

$$O = HHUH$$

i	$\alpha_1(i)$	$\alpha_2(i)$	$\alpha_3(i)$	$\alpha_4(i)$
3 (R)	$.3 \times .2 = .06$			
2 (C)	$.3 \times .5 = .15$			
1 (F)	$.4 \times .9 = .36$	$(.36 \times .8$		
	H	H	U	H

The forward algorithm: Example



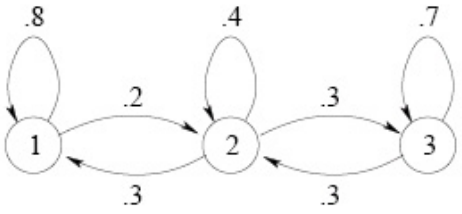
$$\pi = [0.4\ 0.3\ 0.3]$$

$$B = \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \\ 0.2 & 0.8 \end{bmatrix}$$

$$O = HHUH$$

i	$\alpha_1(i)$	$\alpha_2(i)$	$\alpha_3(i)$	$\alpha_4(i)$
3 (R)	$.3 \times .2 = .06$			
2 (C)	$.3 \times .5 = .15$			
1 (F)	$.4 \times .9 = .36$	$(.36 \times .8 + .15 \times .3)$		
	H	H	U	H

The forward algorithm: Example



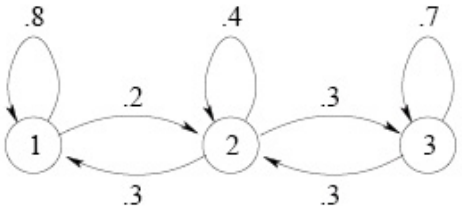
$$\pi = [0.4\ 0.3\ 0.3]$$

$$B = \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \\ 0.2 & 0.8 \end{bmatrix}$$

$$O = HHUH$$

i	$\alpha_1(i)$	$\alpha_2(i)$	$\alpha_3(i)$	$\alpha_4(i)$
3 (R)	$.3 \times .2 = .06$			
2 (C)	$.3 \times .5 = .15$			
1 (F)	$.4 \times .9 = .36$	$(.36 \times .8 + .15 \times .3) \times .9$		
	H	H	U	H

The forward algorithm: Example



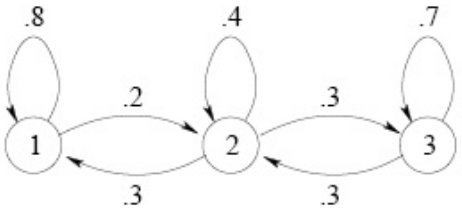
$$\pi = [0.4\ 0.3\ 0.3]$$

$$B = \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \\ 0.2 & 0.8 \end{bmatrix}$$

$$O = HHUH$$

i	$\alpha_1(i)$	$\alpha_2(i)$	$\alpha_3(i)$	$\alpha_4(i)$
3 (R)	$.3 \times .2 = .06$			
2 (C)	$.3 \times .5 = .15$			
1 (F)	$.4 \times .9 = .36$	$(.36 \times .8 + .15 \times .3) \times .9 = .2997$		
	H	H	U	H

The forward algorithm: Example



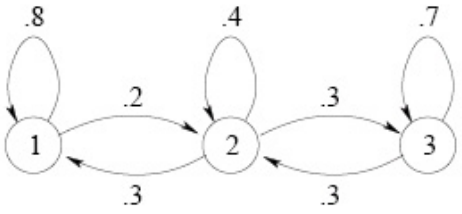
$$\pi = [0.4\ 0.3\ 0.3]$$

$$B = \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \\ 0.2 & 0.8 \end{bmatrix}$$

$$O = HHUH$$

i	$\alpha_1(i)$	$\alpha_2(i)$	$\alpha_3(i)$	$\alpha_4(i)$
3 (R)	$.3 \times .2 = .06$			
2 (C)	$.3 \times .5 = .15$	$(.36 \times .2$		
1 (F)	$.4 \times .9 = .36$	$(.36 \times .8 + .15 \times .3) \times .9 = .2997$		
	H	H	U	H

The forward algorithm: Example



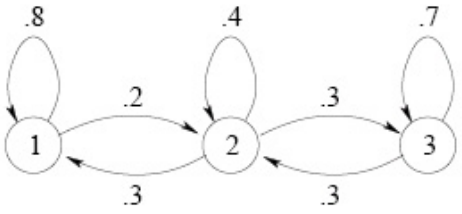
$$\pi = [0.4\ 0.3\ 0.3]$$

$$B = \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \\ 0.2 & 0.8 \end{bmatrix}$$

$$O = HHUH$$

i	$\alpha_1(i)$	$\alpha_2(i)$	$\alpha_3(i)$	$\alpha_4(i)$
3 (R)	$.3 \times .2 = .06$			
2 (C)	$.3 \times .5 = .15$	$(.36 \times .2 + .15 \times .4$		
1 (F)	$.4 \times .9 = .36$	$(.36 \times .8 + .15 \times .3) \times .9 = .2997$		
	H	H	U	H

The forward algorithm: Example



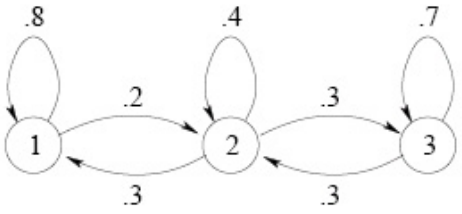
$$\pi = [0.4\ 0.3\ 0.3]$$

$$B = \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \\ 0.2 & 0.8 \end{bmatrix}$$

$$O = HHUH$$

i	$\alpha_1(i)$	$\alpha_2(i)$	$\alpha_3(i)$	$\alpha_4(i)$
3 (R)	$.3 \times .2 = .06$			
2 (C)	$.3 \times .5 = .15$	$(.36 \times .2 + .15 \times .4 + .06 \times .3)$		
1 (F)	$.4 \times .9 = .36$	$(.36 \times .8 + .15 \times .3) \times .9 = .2997$		
	H	H	U	H

The forward algorithm: Example



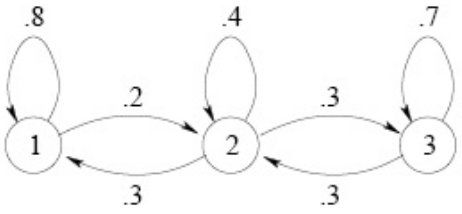
$$\pi = [0.4\ 0.3\ 0.3]$$

$$B = \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \\ 0.2 & 0.8 \end{bmatrix}$$

$$O = HHUH$$

i	$\alpha_1(i)$	$\alpha_2(i)$	$\alpha_3(i)$	$\alpha_4(i)$
3 (R)	$.3 \times .2 = .06$			
2 (C)	$.3 \times .5 = .15$	$(.36 \times .2 + .15 \times .4 + .06 \times .3) \times .5$		
1 (F)	$.4 \times .9 = .36$	$(.36 \times .8 + .15 \times .3) \times .9 = .2997$		
	H	H	U	H

The forward algorithm: Example



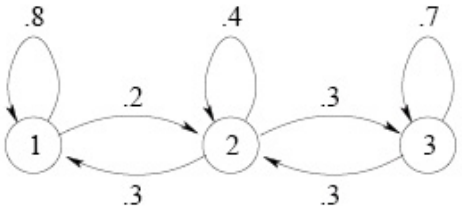
$$\pi = [0.4\ 0.3\ 0.3]$$

$$B = \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \\ 0.2 & 0.8 \end{bmatrix}$$

$$O = HHUH$$

i	$\alpha_1(i)$	$\alpha_2(i)$	$\alpha_3(i)$	$\alpha_4(i)$
3 (R)	$.3 \times .2 = .06$			
2 (C)	$.3 \times .5 = .15$	$(.36 \times .2 + .15 \times .4 + .06 \times .3) \times .5 = .075$		
1 (F)	$.4 \times .9 = .36$	$(.36 \times .8 + .15 \times .3) \times .9 = .2997$		
	H	H	U	H

The forward algorithm: Example



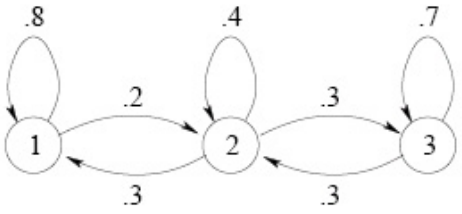
$$\pi = [0.4\ 0.3\ 0.3]$$

$$B = \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \\ 0.2 & 0.8 \end{bmatrix}$$

$$O = HHUH$$

i	$\alpha_1(i)$	$\alpha_2(i)$	$\alpha_3(i)$	$\alpha_4(i)$
3 (R)	$.3 \times .2 = .06$	$(.15 \times .3 + .06 \times .7)$		
2 (C)	$.3 \times .5 = .15$	$(.36 \times .2 + .15 \times .4 + .06 \times .3) \times .5 = .075$		
1 (F)	$.4 \times .9 = .36$	$(.36 \times .8 + .15 \times .3) \times .9 = .2997$		
	H	H	U	H

The forward algorithm: Example



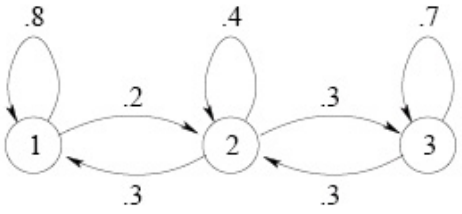
$$\pi = [0.4\ 0.3\ 0.3]$$

$$B = \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \\ 0.2 & 0.8 \end{bmatrix}$$

$$O = HHUH$$

i	$\alpha_1(i)$	$\alpha_2(i)$	$\alpha_3(i)$	$\alpha_4(i)$
3 (R)	$.3 \times .2 = .06$	$(.15 \times .3 + .06 \times .7) \times .2$		
2 (C)	$.3 \times .5 = .15$	$(.36 \times .2 + .15 \times .4 + .06 \times .3) \times .5 = .075$		
1 (F)	$.4 \times .9 = .36$	$(.36 \times .8 + .15 \times .3) \times .9 = .2997$		
	H	H	U	H

The forward algorithm: Example



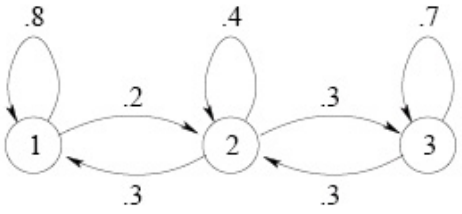
$$\pi = [0.4\ 0.3\ 0.3]$$

$$B = \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \\ 0.2 & 0.8 \end{bmatrix}$$

$$O = HHUH$$

i	$\alpha_1(i)$	$\alpha_2(i)$	$\alpha_3(i)$	$\alpha_4(i)$
3 (R)	$.3 \times .2 = .06$	$(.15 \times .3 + .06 \times .7) \times .2 = .0174$		
2 (C)	$.3 \times .5 = .15$	$(.36 \times .2 + .15 \times .4 + .06 \times .3) \times .5 = .075$		
1 (F)	$.4 \times .9 = .36$	$(.36 \times .8 + .15 \times .3) \times .9 = .2997$		
	H	H	U	H

The forward algorithm: Example



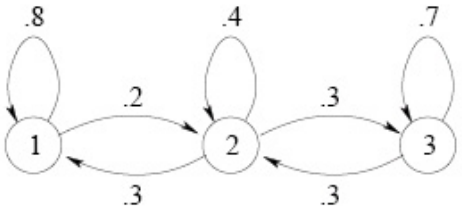
$$\pi = [0.4\ 0.3\ 0.3]$$

$$B = \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \\ 0.2 & 0.8 \end{bmatrix}$$

$$O = HHUH$$

i	$\alpha_1(i)$	$\alpha_2(i)$	$\alpha_3(i)$	$\alpha_4(i)$
3 (R)	$.3 \times .2 = .06$	$(.15 \times .3 + .06 \times .7) \times .2 = .0174$		
2 (C)	$.3 \times .5 = .15$	$(.36 \times .2 + .15 \times .4 + .06 \times .3) \times .5 = .075$		
1 (F)	$.4 \times .9 = .36$	$(.36 \times .8 + .15 \times .3) \times .9 = .2997$	$(.2997 \times .8 + .075 \times .3) \times .1 = .026226$	
	H	H	U	H

The forward algorithm: Example



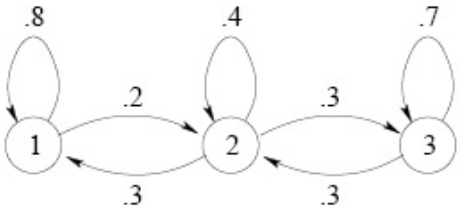
$$\pi = [0.4\ 0.3\ 0.3]$$

$$B = \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \\ 0.2 & 0.8 \end{bmatrix}$$

$$O = HHUH$$

i	$\alpha_1(i)$	$\alpha_2(i)$	$\alpha_3(i)$	$\alpha_4(i)$
3 (R)	$.3 \times .2 = .06$	$(.15 \times .3 + .06 \times .7) \times .2 = .0174$		
2 (C)	$.3 \times .5 = .15$	$(.36 \times .2 + .15 \times .4 + .06 \times .3) \times .5 = .075$	$(.2997 \times .2 + .075 \times .4 + .0174 \times .3) \times .5 = .04758$	
1 (F)	$.4 \times .9 = .36$	$(.36 \times .8 + .15 \times .3) \times .9 = .2997$	$(.2997 \times .8 + .075 \times .3) \times .1 = .026226$	
	H	H	U	H

The forward algorithm: Example



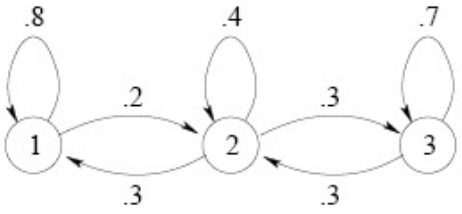
$$\pi = [0.4\ 0.3\ 0.3]$$

$$B = \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \\ 0.2 & 0.8 \end{bmatrix}$$

$$O = HHUH$$

i	$\alpha_1(i)$	$\alpha_2(i)$	$\alpha_3(i)$	$\alpha_4(i)$
3 (R)	$.3 \times .2$ $= .06$	$(.15 \times .3$ $+.06 \times .7) \times .2$ $= .0174$	$(.075 \times .3$ $+.0174 \times .7) \times .8$ $= .027744$	
2 (C)	$.3 \times .5$ $= .15$	$(.36 \times .2$ $+.15 \times .4$ $+.06 \times .3) \times .5$ $= .075$	$(.2997 \times .2$ $+.075 \times .4$ $+.0174 \times .3) \times .5$ $= .04758$	
1 (F)	$.4 \times .9$ $= .36$	$(.36 \times .8$ $+.15 \times .3) \times .9$ $= .2997$	$(.2997 \times .8$ $+.075 \times .3) \times .1$ $= .026226$	
	H	H	U	H

The forward algorithm: Example



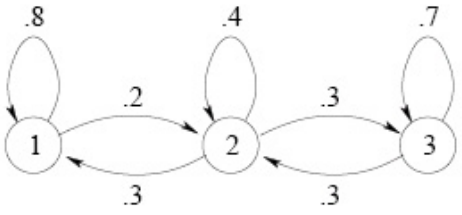
$$\pi = [0.4\ 0.3\ 0.3]$$

$$B = \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \\ 0.2 & 0.8 \end{bmatrix}$$

$$O = HHUH$$

i	$\alpha_1(i)$	$\alpha_2(i)$	$\alpha_3(i)$	$\alpha_4(i)$
3 (R)	$.3 \times .2 = .06$	$(.15 \times .3 + .06 \times .7) \times .2 = .0174$	$(.075 \times .3 + .0174 \times .7) \times .8 = .027744$	
2 (C)	$.3 \times .5 = .15$	$(.36 \times .2 + .15 \times .4 + .06 \times .3) \times .5 = .075$	$(.2997 \times .2 + .075 \times .4 + .0174 \times .3) \times .5 = .04758$	
1 (F)	$.4 \times .9 = .36$	$(.36 \times .8 + .15 \times .3) \times .9 = .2997$	$(.2997 \times .8 + .075 \times .3) \times .1 = .026226$	$\ldots$ $\ldots$ $= .03172932$
	H	H	U	H

The forward algorithm: Example



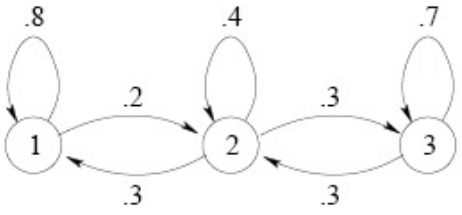
$$\pi = [0.4\ 0.3\ 0.3]$$

$$B = \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \\ 0.2 & 0.8 \end{bmatrix}$$

$$O = HHUH$$

i	$\alpha_1(i)$	$\alpha_2(i)$	$\alpha_3(i)$	$\alpha_4(i)$
3 (R)	$.3 \times .2$ $= .06$	$(.15 \times .3$ $+.06 \times .7) \times .2$ $= .0174$	$(.075 \times .3$ $+.0174 \times .7) \times .8$ $= .027744$	
2 (C)	$.3 \times .5$ $= .15$	$(.36 \times .2$ $+.15 \times .4$ $+.06 \times .3) \times .5$ $= .075$	$(.2997 \times .2$ $+.075 \times .4$ $+.0174 \times .3) \times .5$ $= .04758$	$\ldots$ $\ldots$ $\ldots$ $= .0163002$
1 (F)	$.4 \times .9$ $= .36$	$(.36 \times .8$ $+.15 \times .3) \times .9$ $= .2997$	$(.2997 \times .8$ $+.075 \times .3) \times .1$ $= .026226$	$\ldots$ $\ldots$ $= .03172932$
	H	H	U	H

The forward algorithm: Example



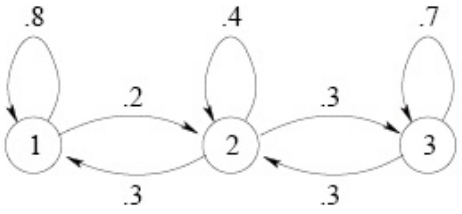
$$\pi = [0.4\ 0.3\ 0.3]$$

$$B = \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \\ 0.2 & 0.8 \end{bmatrix}$$

$$O = HHUH$$

i	$\alpha_1(i)$	$\alpha_2(i)$	$\alpha_3(i)$	$\alpha_4(i)$
3 (R)	$.3 \times .2 = .06$	$(.15 \times .3 + .06 \times .7) \times .2 = .0174$	$(.075 \times .3 + .0174 \times .7) \times .8 = .027744$	$\ldots \ldots = .00673896$
2 (C)	$.3 \times .5 = .15$	$(.36 \times .2 + .15 \times .4 + .06 \times .3) \times .5 = .075$	$(.2997 \times .2 + .075 \times .4 + .0174 \times .3) \times .5 = .04758$	$\ldots \ldots = .0163002$
1 (F)	$.4 \times .9 = .36$	$(.36 \times .8 + .15 \times .3) \times .9 = .2997$	$(.2997 \times .8 + .075 \times .3) \times .1 = .026226$	$\ldots \ldots = .03172932$
	H	H	U	H

The forward algorithm: Example



$\pi = [0.4\ 0.3\ 0.3]$

$$B = \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \\ 0.2 & 0.8 \end{bmatrix}$$

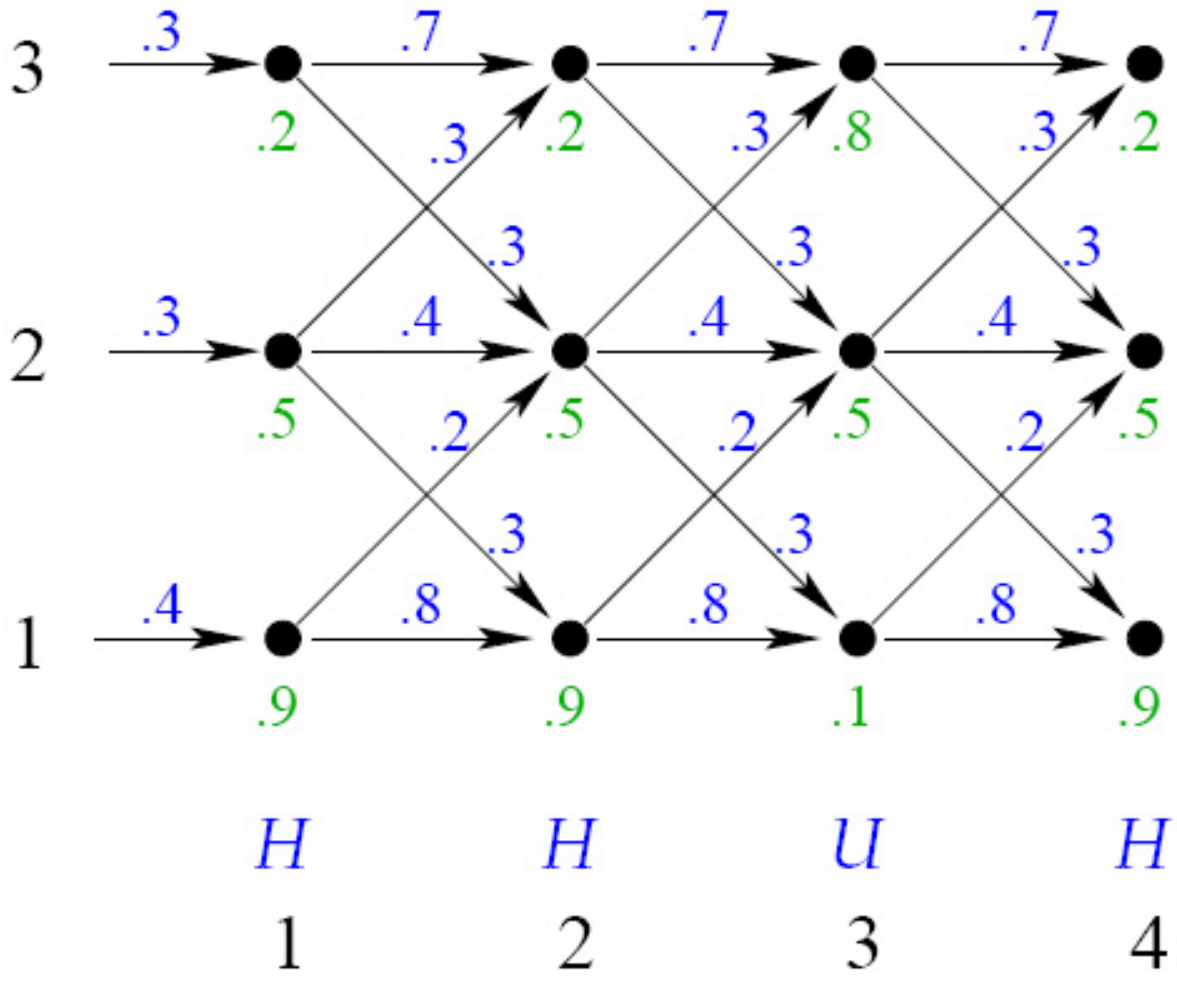
$O = HHUH$

i	$\alpha_1(i)$	$\alpha_2(i)$	$\alpha_3(i)$	$\alpha_4(i)$
3 (R)	$.3 \times .2$ $= .06$	$(.15 \times .3$ $+.06 \times .7) \times .2$ $= .0174$	$(.075 \times .3$ $+.0174 \times .7) \times .8$ $= .027744$	$\ldots$ $\ldots$ $= .00673896$
2 (C)	$.3 \times .5$ $= .15$	$(.36 \times .2$ $+.15 \times .4$ $+.06 \times .3) \times .5$ $= .075$	$(.2997 \times .2$ $+.075 \times .4$ $+.0174 \times .3) \times .5$ $= .04758$	$\ldots$ $\ldots$ $\ldots$ $= .0163002$
1 (F)	$.4 \times .9$ $= .36$	$(.36 \times .8$ $+.15 \times .3) \times .9$ $= .2997$	$(.2997 \times .8$ $+.075 \times .3) \times .1$ $= .026226$	$\ldots$ $\ldots$ $= .03172932$
	H	H	U	H

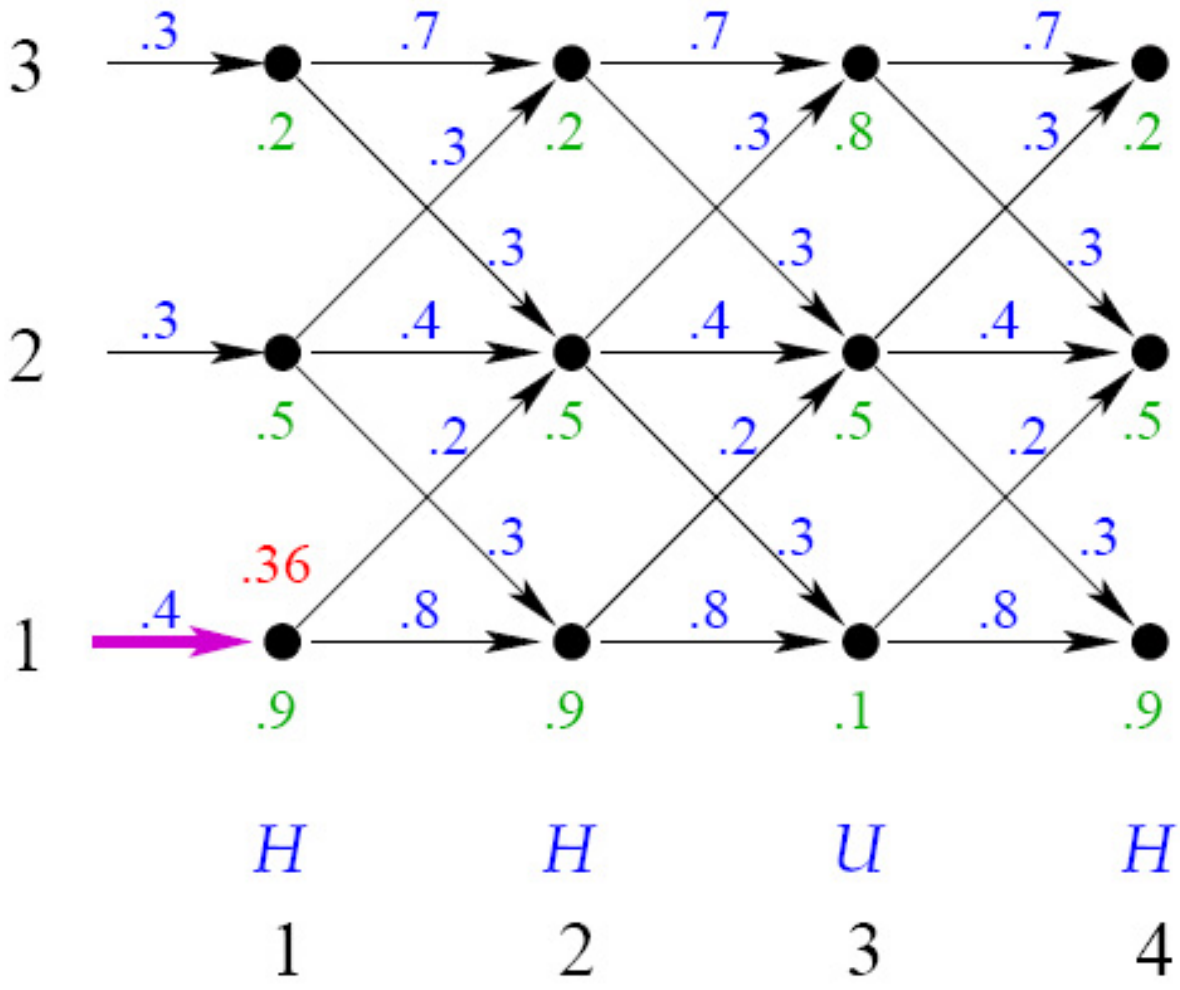
Finally:

$$p(HHUH|\lambda) = .00673896 + .0163002 + .03172932 = .05476848$$

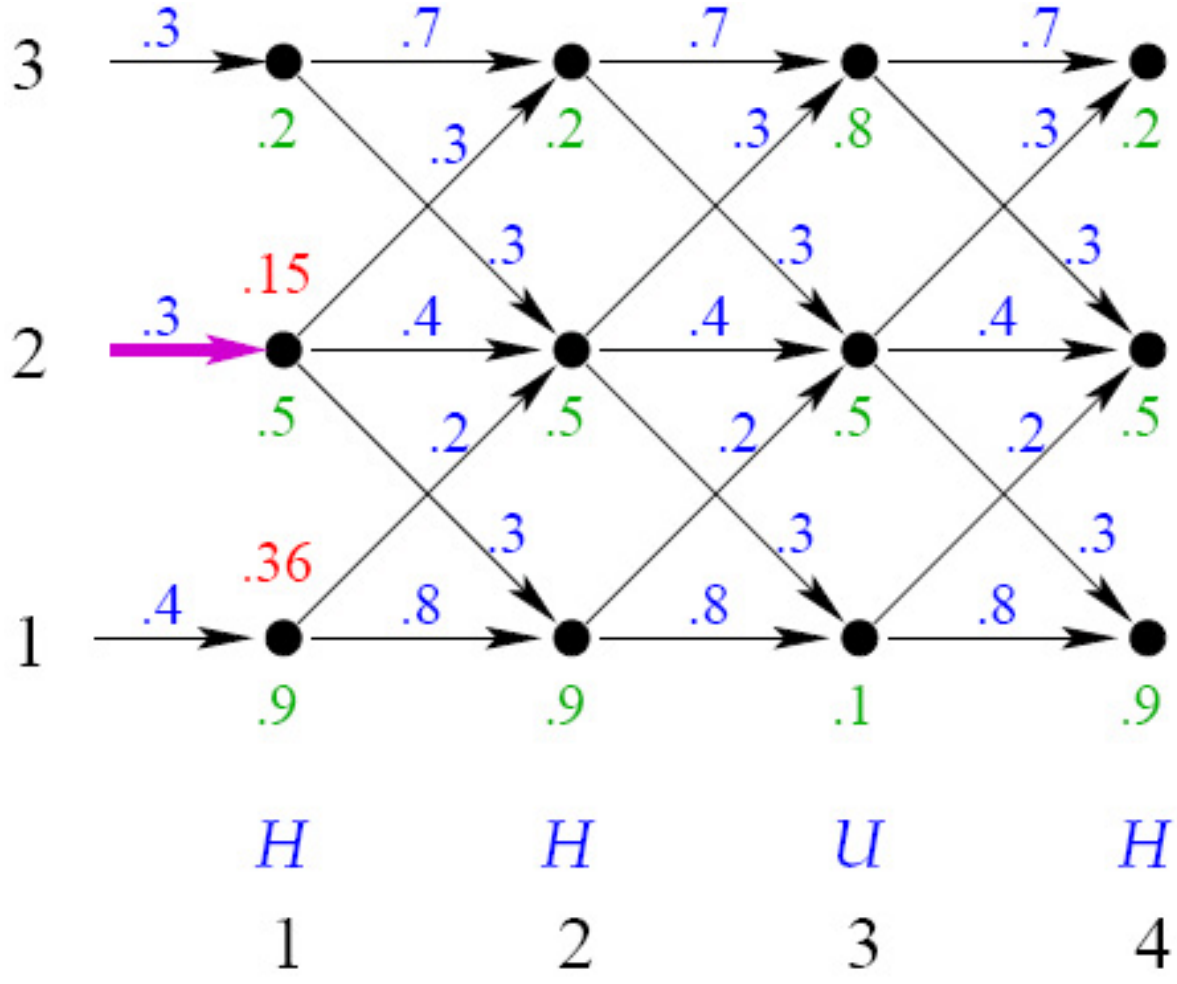
The forward algorithm: Example



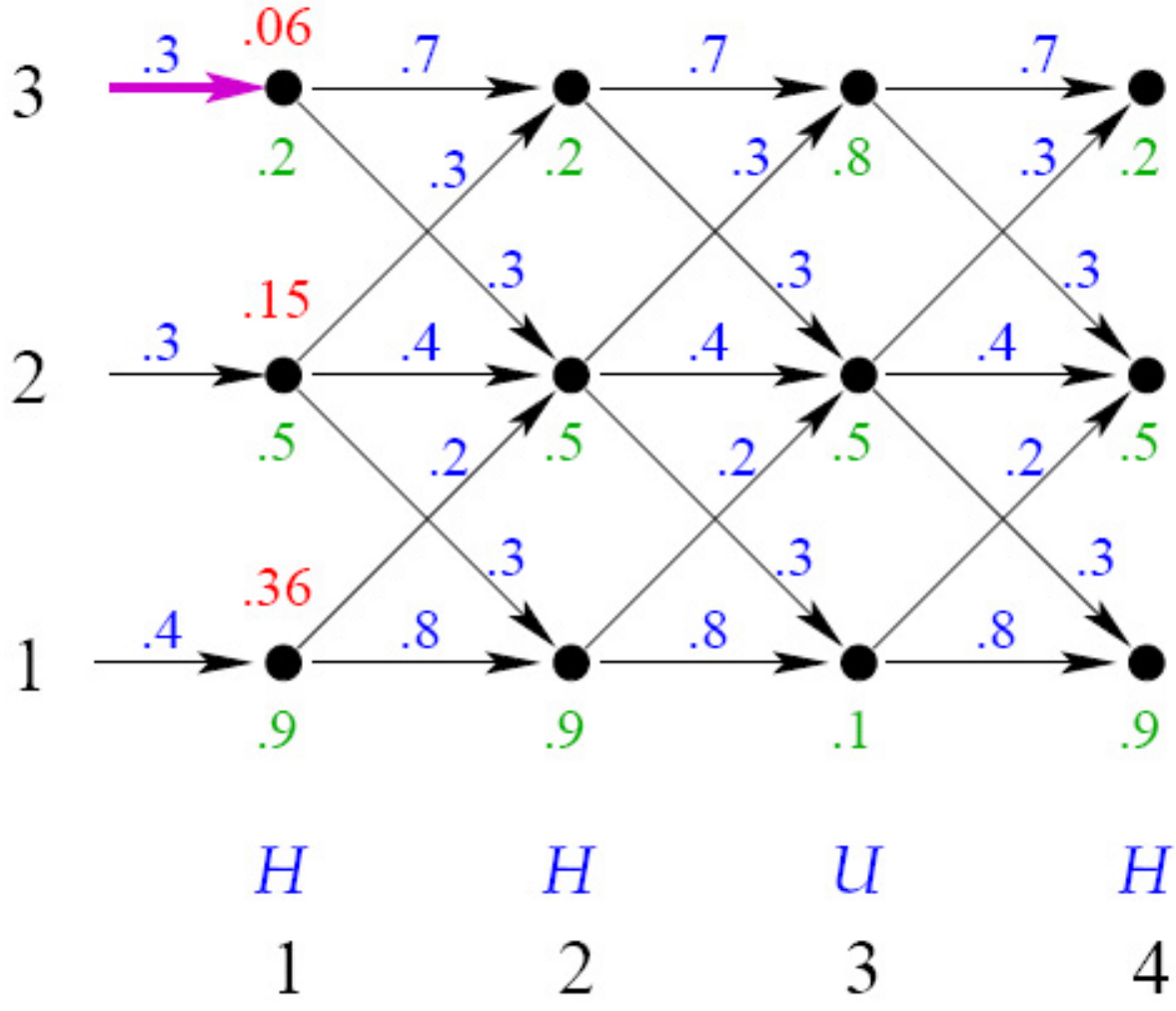
The forward algorithm: Example



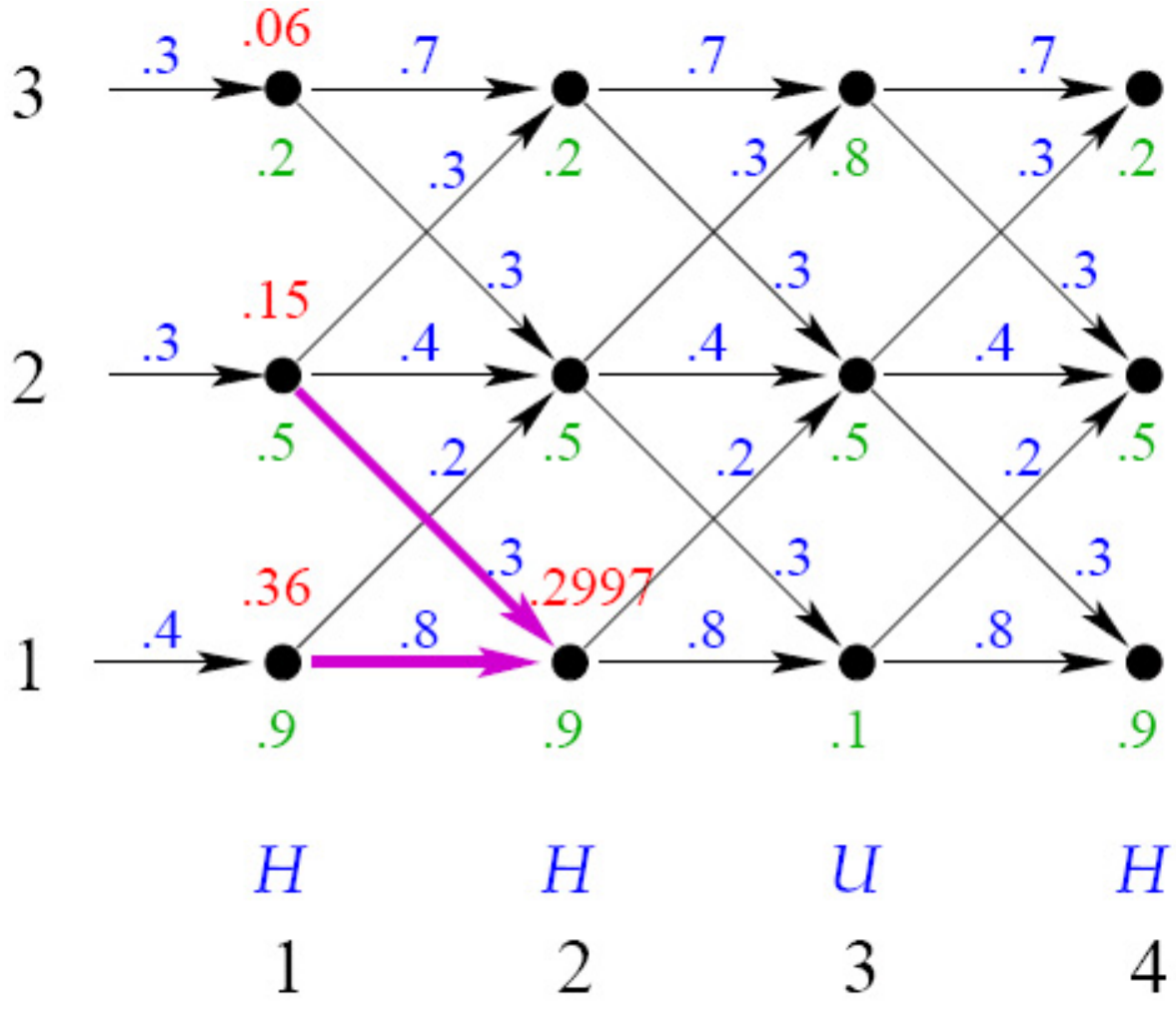
The forward algorithm: Example



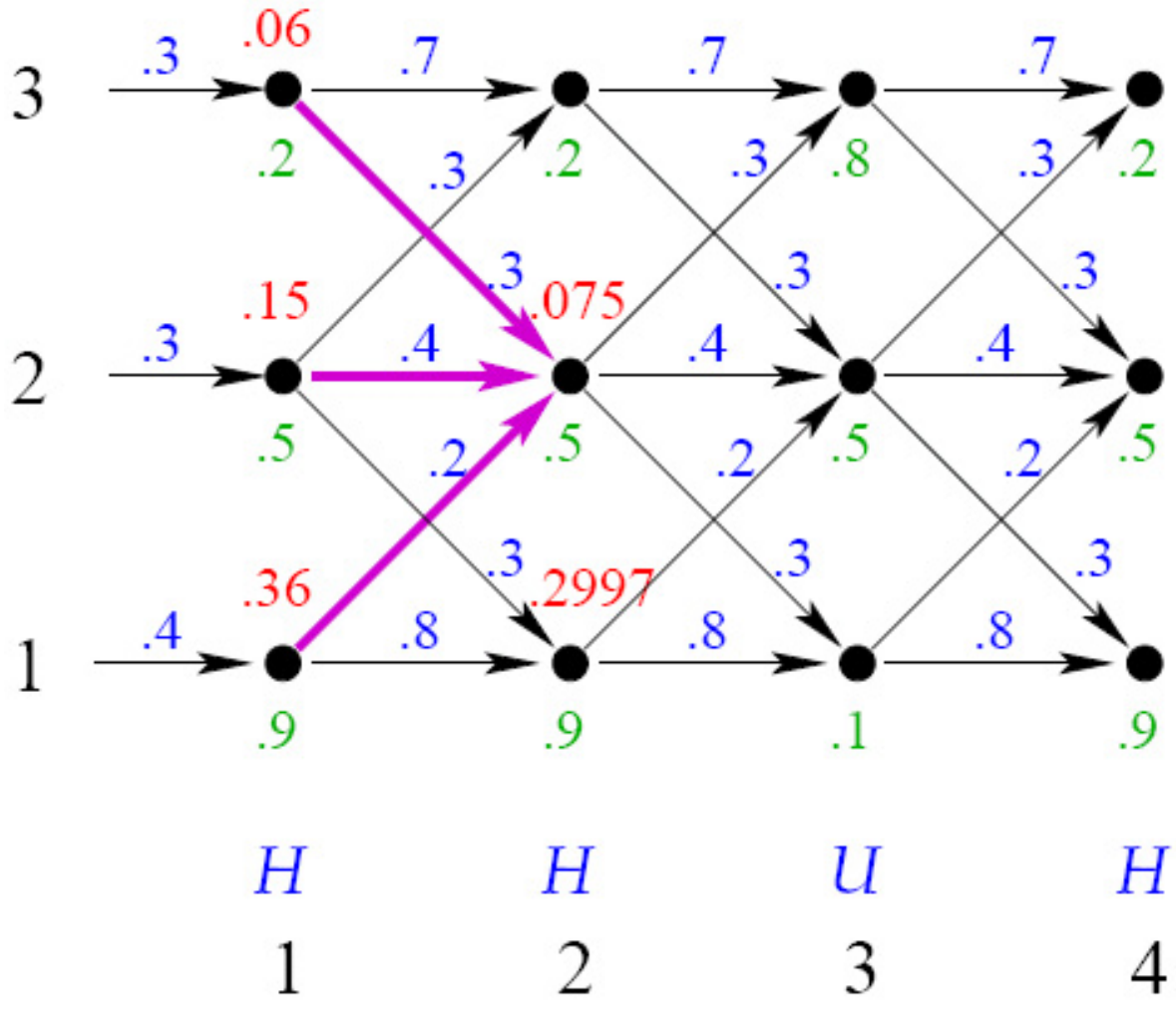
The forward algorithm: Example



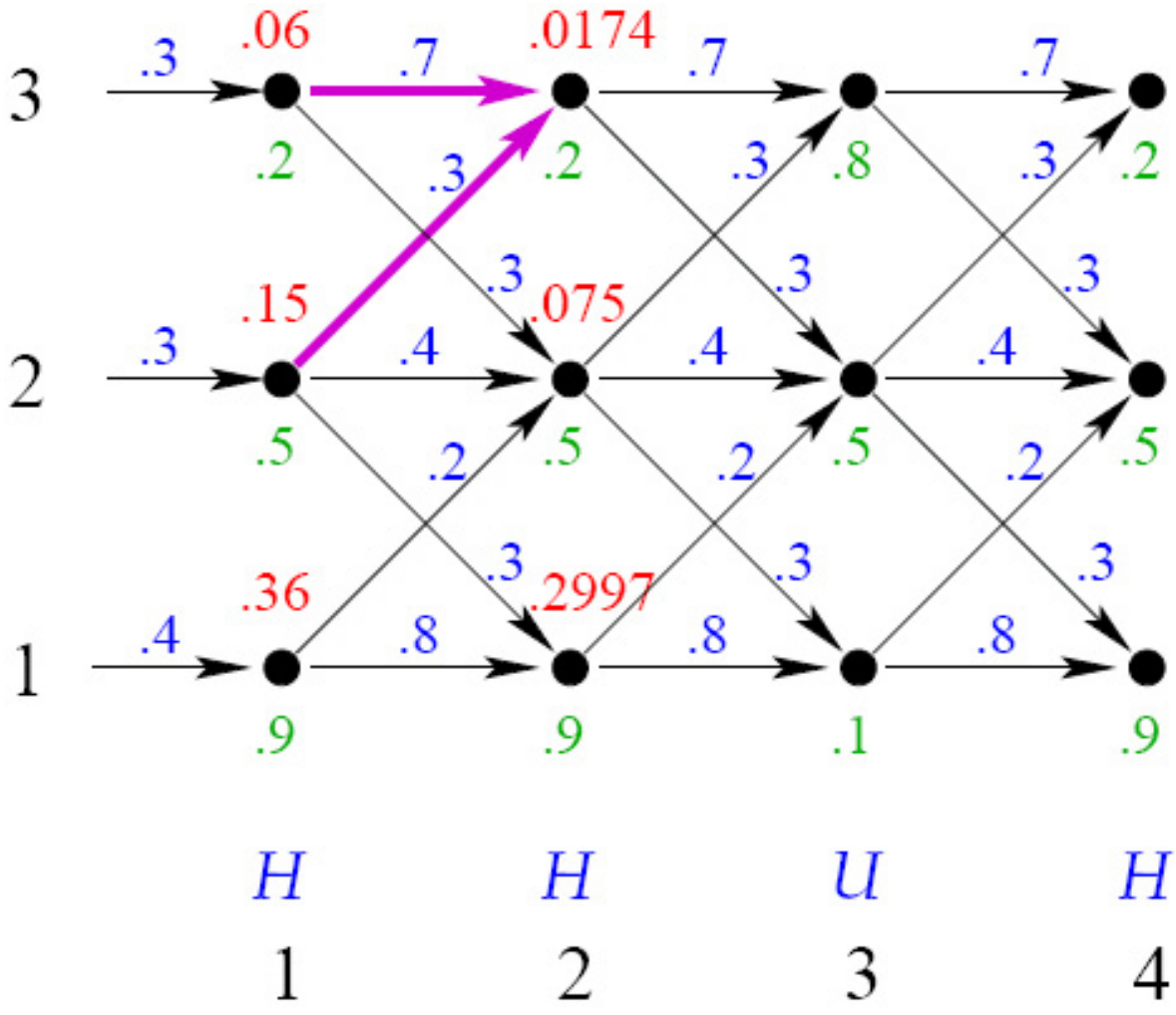
The forward algorithm: Example



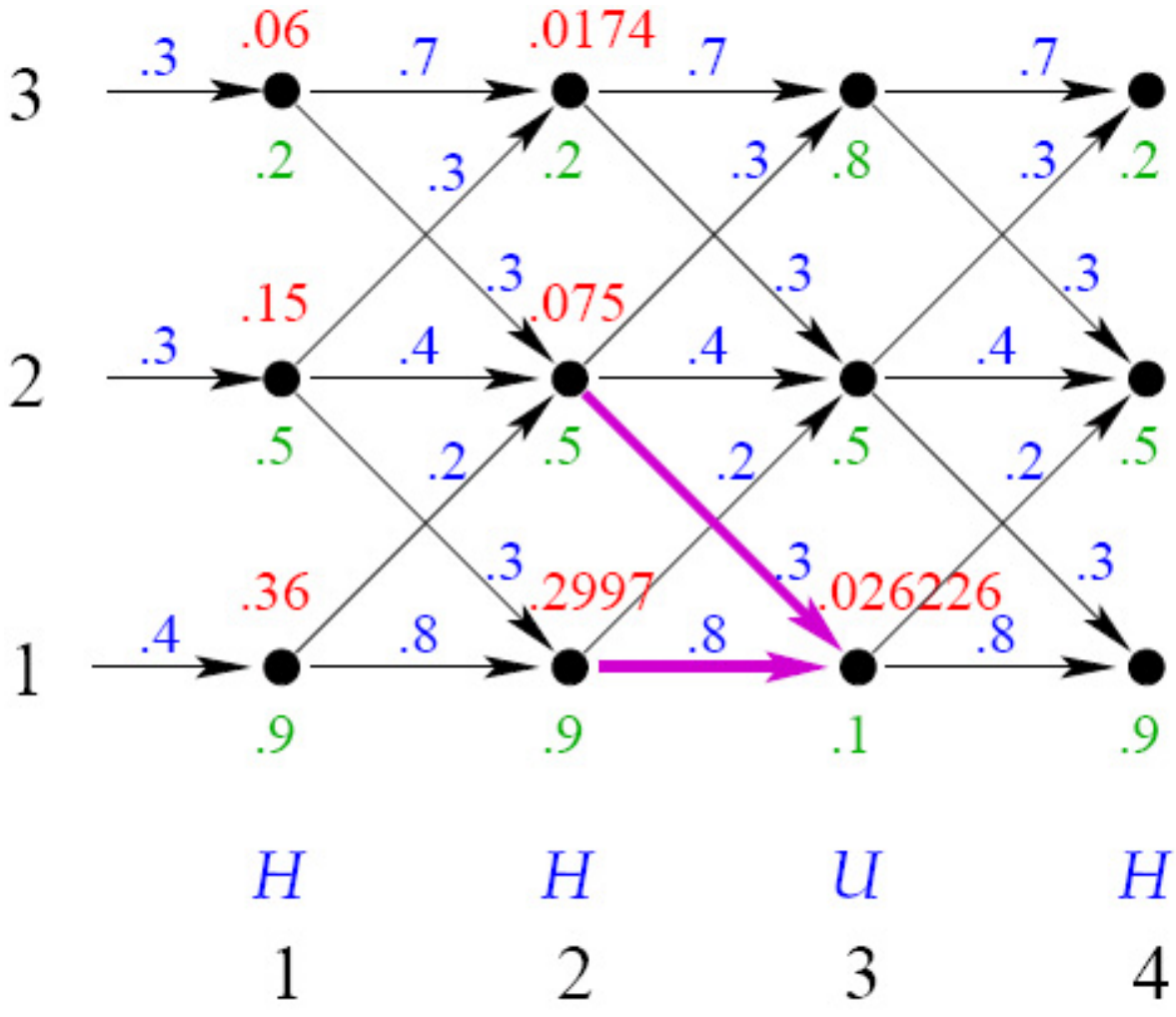
The forward algorithm: Example



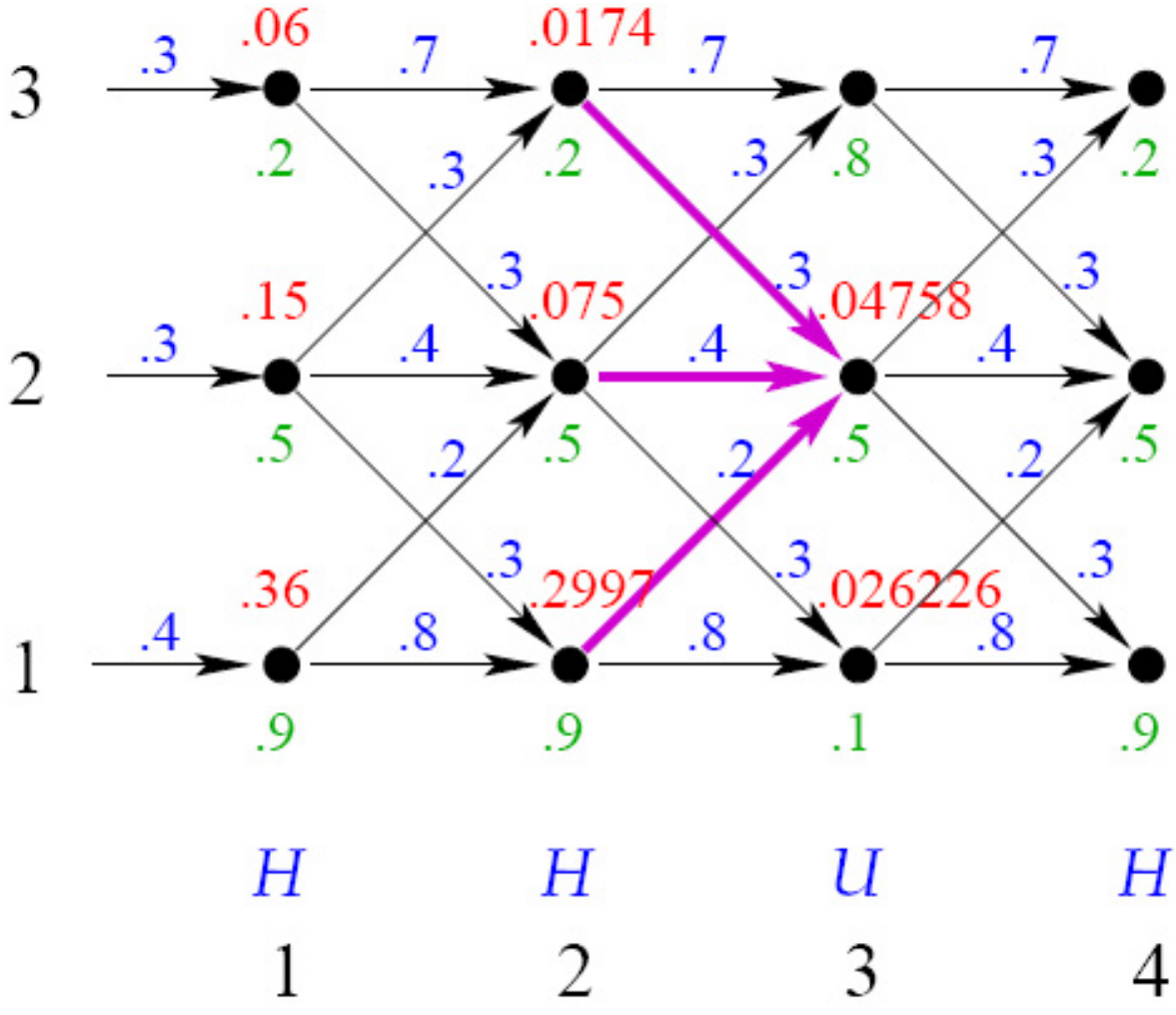
The forward algorithm: Example



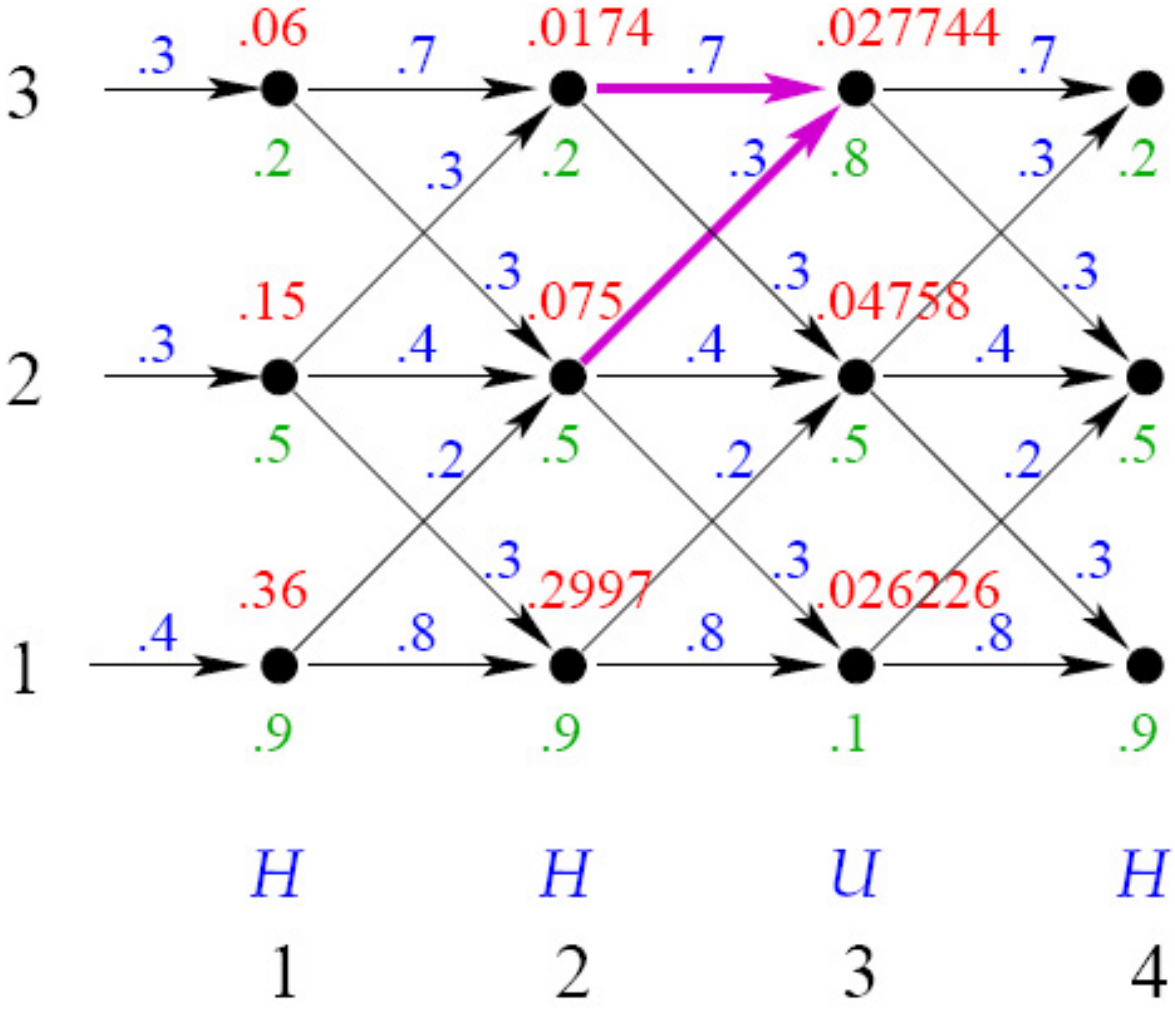
The forward algorithm: Example



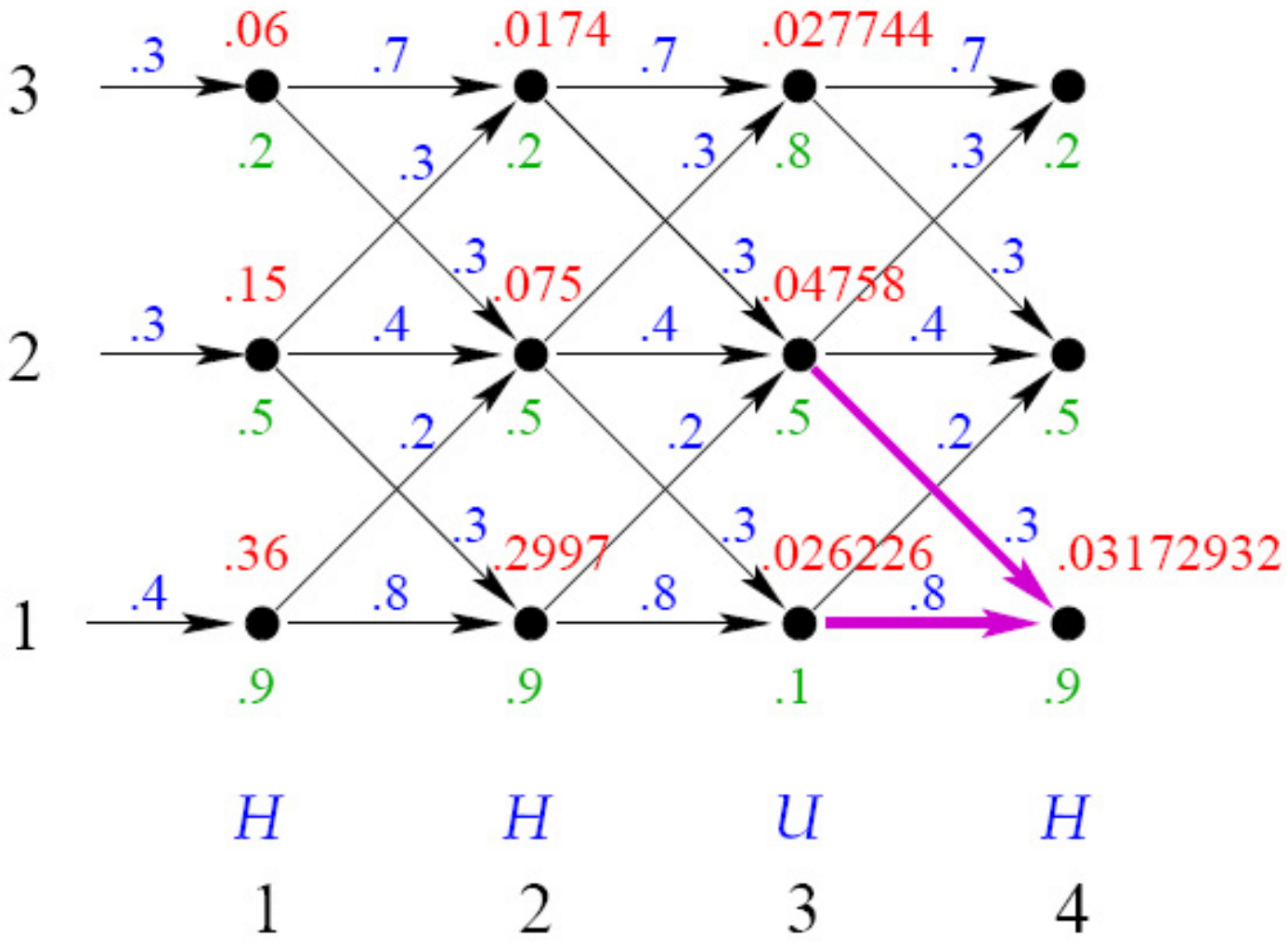
The forward algorithm: Example



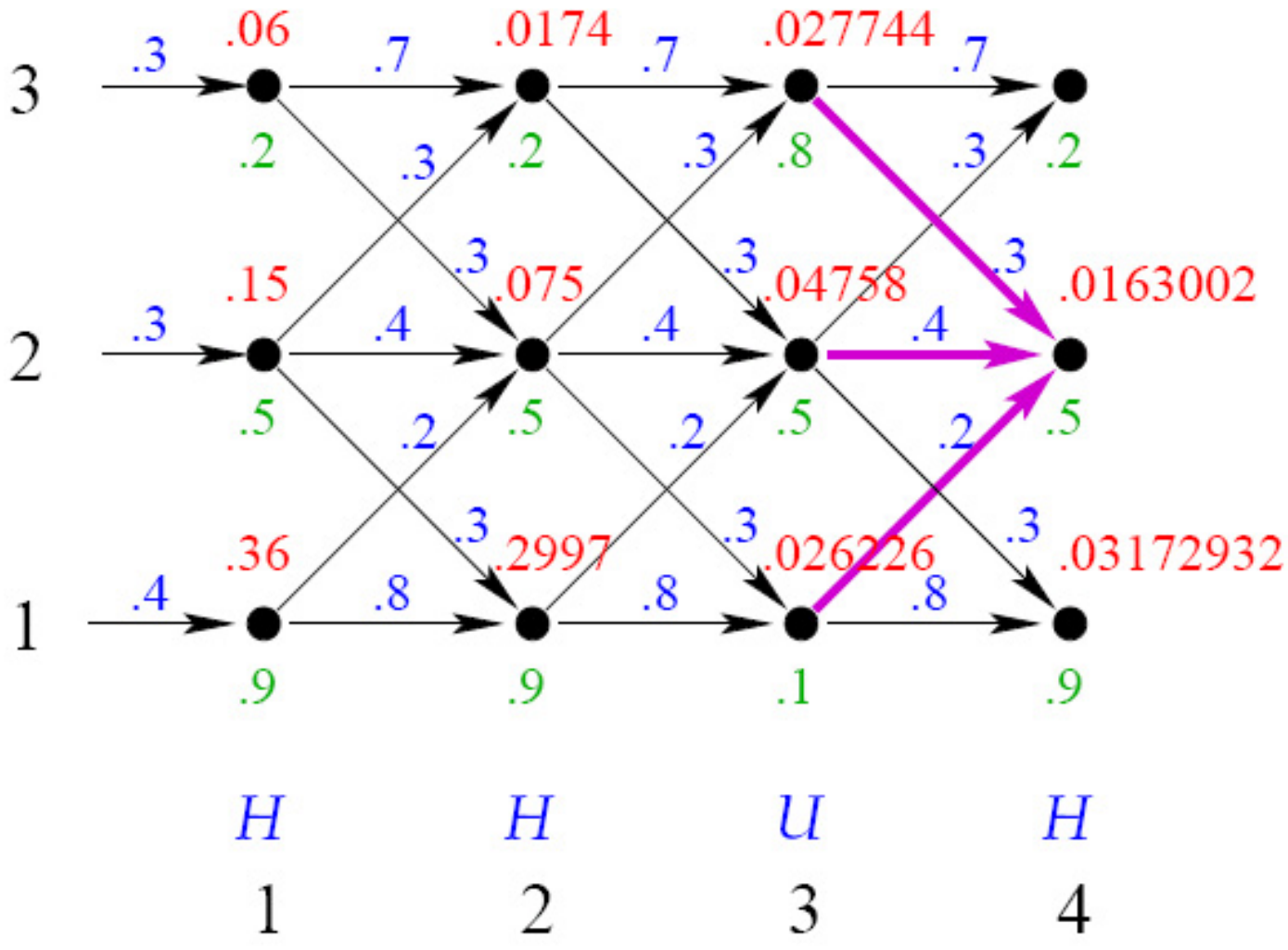
The forward algorithm: Example



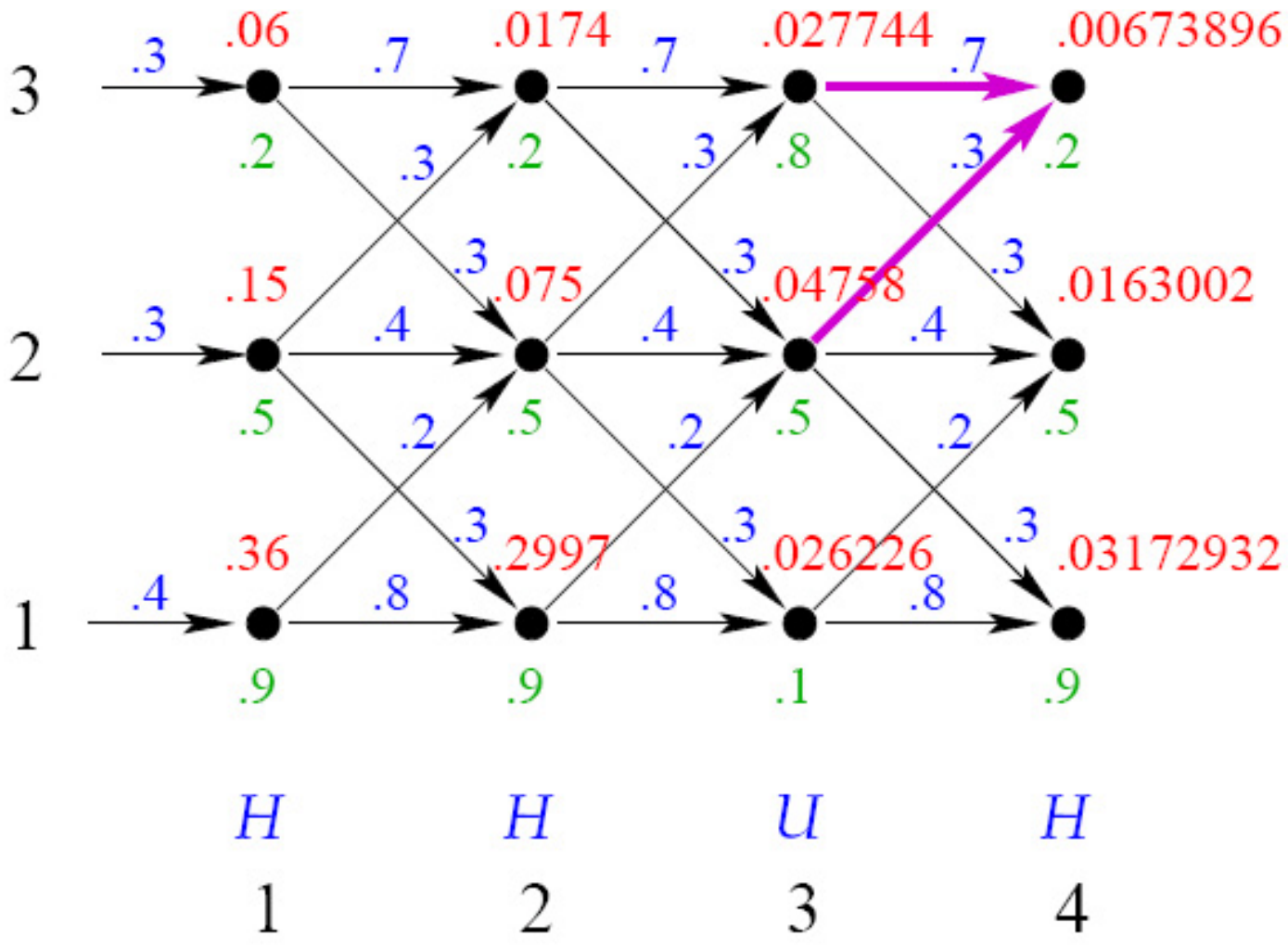
The forward algorithm: Example



The forward algorithm: Example



The forward algorithm: Example



Computing $p(O|\lambda)$: The backward algorithm

- Define the "backward" variable $\beta_t(i)$:

$$\beta_t(i) = P(\mathbf{o}_{t+1}\mathbf{o}_{t+2}\ldots\mathbf{o}_T|q_t = i, \lambda), \quad \begin{array}{rcl} 1 \leq & t & \leq T-1, \\ 1 \leq & i & \leq N \end{array}$$

- Initialization: $\beta_T(i) = 1$
- Recursion: $\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(\mathbf{o}_{t+1}) \beta_{t+1}(j)$
- Finally: $p(\mathbf{O}|\lambda) = \sum_{i=1}^{N} \pi_i b_i(\mathbf{o}_1) \beta_1(i)$
- As in forward algorithm, computation is N^2T
- Either algorithm alone can be used to compute $p(\mathbf{O}|\lambda)$, but both αs and βs will be needed for the training problem

CONGRATULATIONS!

- We've solved Problem 1 (Scoring)!
- Questions? Comments?
- We are ready to apply HMMs to an ASR problem
- But first...

An extension to the example

Your friend may be in one of three cities – Chicago, Seattle, or Los Angeles. Could you guess which city he is in based on $\mathbf{O}$?

An extension to the example

Your friend may be in one of three cities – Chicago, Seattle, or Los Angeles. Could you guess which city he is in based on $\mathbf{O}$?

Chicago model λ_C:

$$\pi = [0.4\ 0.3\ 0.3], \mathbf{A} = \begin{bmatrix} 0.8 & 0.2 & 0 \\ 0.3 & 0.4 & 0.3 \\ 0 & 0.3 & 0.7 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \\ 0.2 & 0.8 \end{bmatrix}$$

Seattle model λ_S:

$$\pi = [0.1\ 0.4\ 0.5], \mathbf{A} = \begin{bmatrix} 0.9 & 0.1 & 0 \\ 0.1 & 0.6 & 0.3 \\ 0 & 0.2 & 0.8 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \\ 0.2 & 0.8 \end{bmatrix}$$

Los Angeles model λ_L:

$$\pi = [0.7\ 0.2\ 0.1], \mathbf{A} = \begin{bmatrix} 0.95 & 0.05 & 0 \\ 0.2 & 0.7 & 0.1 \\ 0 & 0.3 & 0.7 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \\ 0.2 & 0.8 \end{bmatrix}$$

An extension to the example (2)

Choose the city C^* that satisfies

$$\begin{aligned} C^* &= \underset{C}{\operatorname{argmax}}\, p(\lambda_C|\mathbf{O}) \\ &= \underset{C}{\operatorname{argmax}}\, p(\mathbf{O}|\lambda_C)p(\lambda_C) \end{aligned}$$

If the cities are equally likely *a priori*, then pick the city with the highest $p(\mathbf{O}|\lambda_C)$

Isolated-word recognition with whole-word HMMs

- Analogy with the weather-mood model:

model component	weather model	ASR
observation o_t	mood on day t	acoustic feature vector (or VQ value) in frame t
state q_t	weather condition	"state"
model	city	word

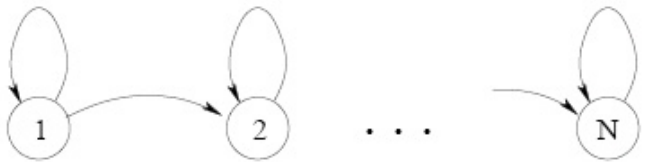
- Hypothesized word w^* is the one with the highest $p(\mathbf{O}|\lambda_w)$ (assuming words are equally likely)

Isolated-word recognition with whole-word HMMs

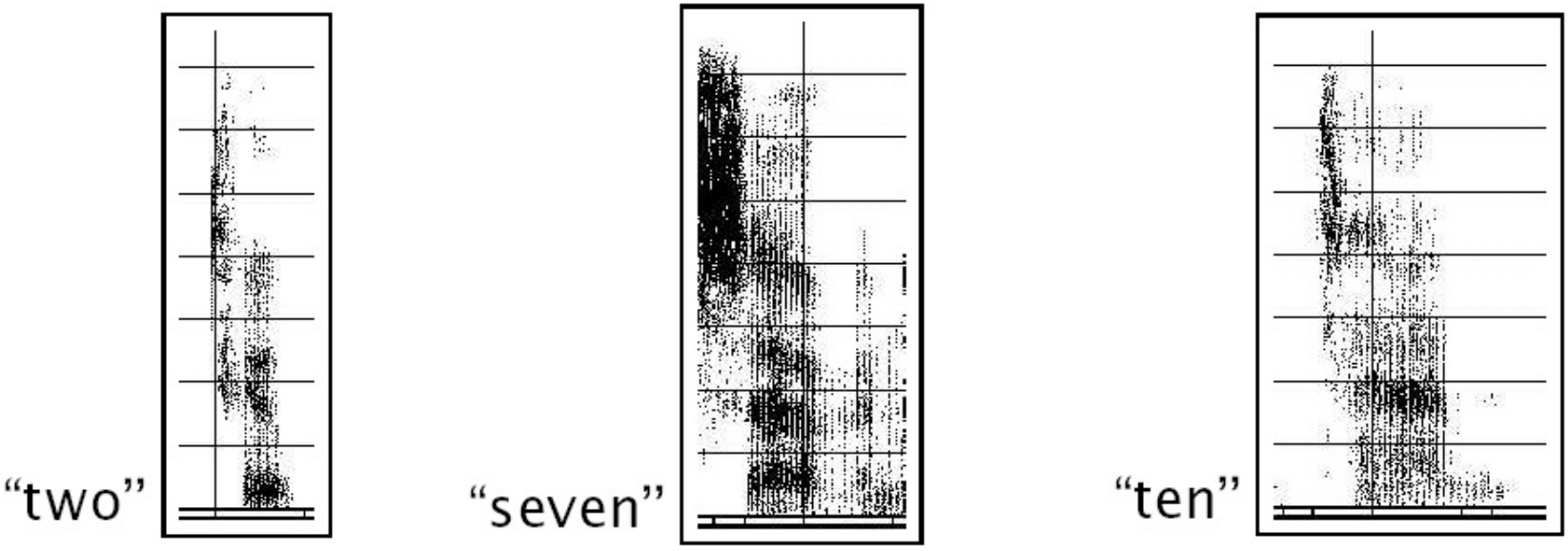
- Analogy with the weather-mood model:

model component	weather model	ASR
observation o_t	mood on day t	acoustic feature vector (or VQ value) in frame t
state q_t	weather condition	"state"
model	city	word

- Hypothesized word w^* is the one with the highest $p(\mathbf{O}|\lambda_w)$ (assuming words are equally likely)
- Typical HMMs for ASR are left-to-right:

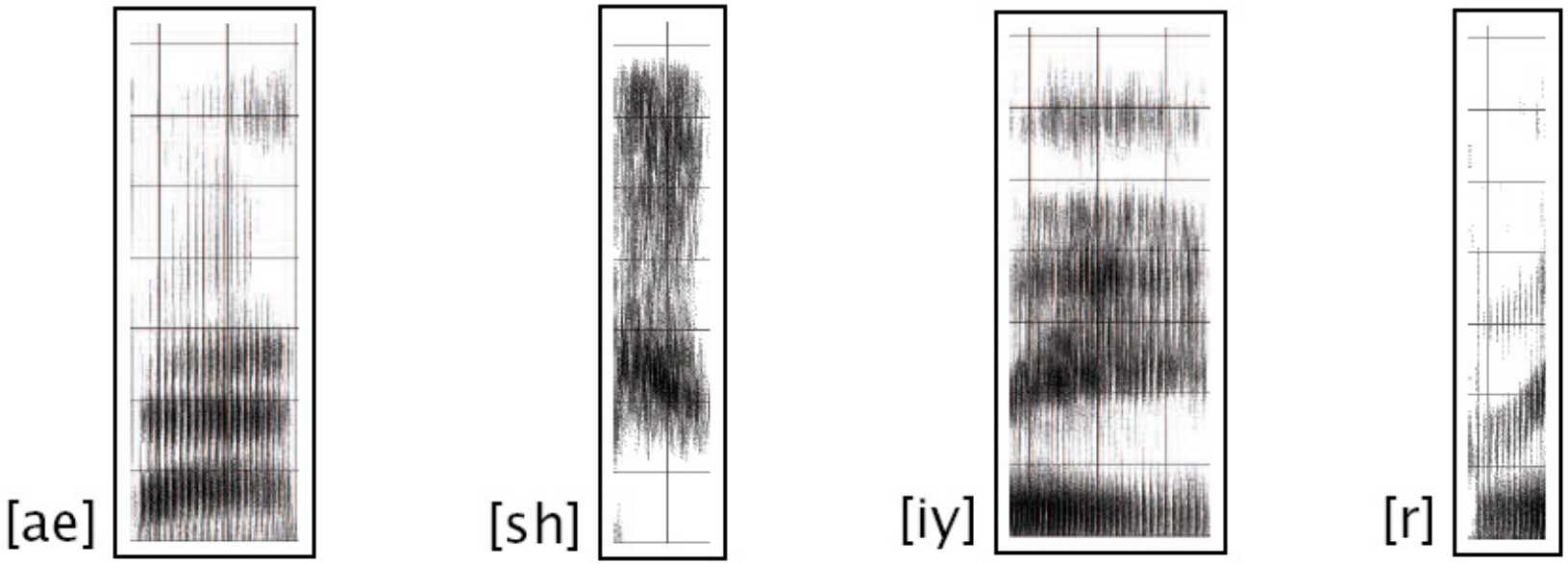


How many states per word model? (What IS a state?)



Phone classification with HMMs

Same approach as isolated-word recognition. How many states per phone?



HMMs for ASR: A historical perspective

Year	Authors	Contribution
1966	Baum et al.	Statistical Formulation
1975	Baker	Discrete HMM
1976	Jelinek et al.	Discrete HMM
1985	Juang et al.	Continuous HMM
1988	Lee	Speaker-independent continuous speech ASR using HMMs
1990s	Young	HMM Toolkit (HTK)

Looking ahead: Continuous-speech (multiword sequence) recognition with HMMs

- One approach: String together whole-word HMMs to make whole-sentence HMMs
  - Works well for small-vocabulary tasks (enough training data per HMM), e.g. digit string recognition
  - For large-vocabulary tasks, string together sub-word HMMs (e.g. phone HMMs) instead – more on this later...

What are the observation (emission) distributions?

How do we define $\{b_i(k)\} = P(o_t = v_k | q_t = i)$?

- For weather-mood HMM, we had a vector of probabilities for each state
- Makes sense when observations are *discrete* variables
- But speech observations are continuous-valued multidimensional feature vectors!

What are the observation (emission) distributions?

How do we define $\{b_i(k)\} = P(o_t = v_k | q_t = i)$?

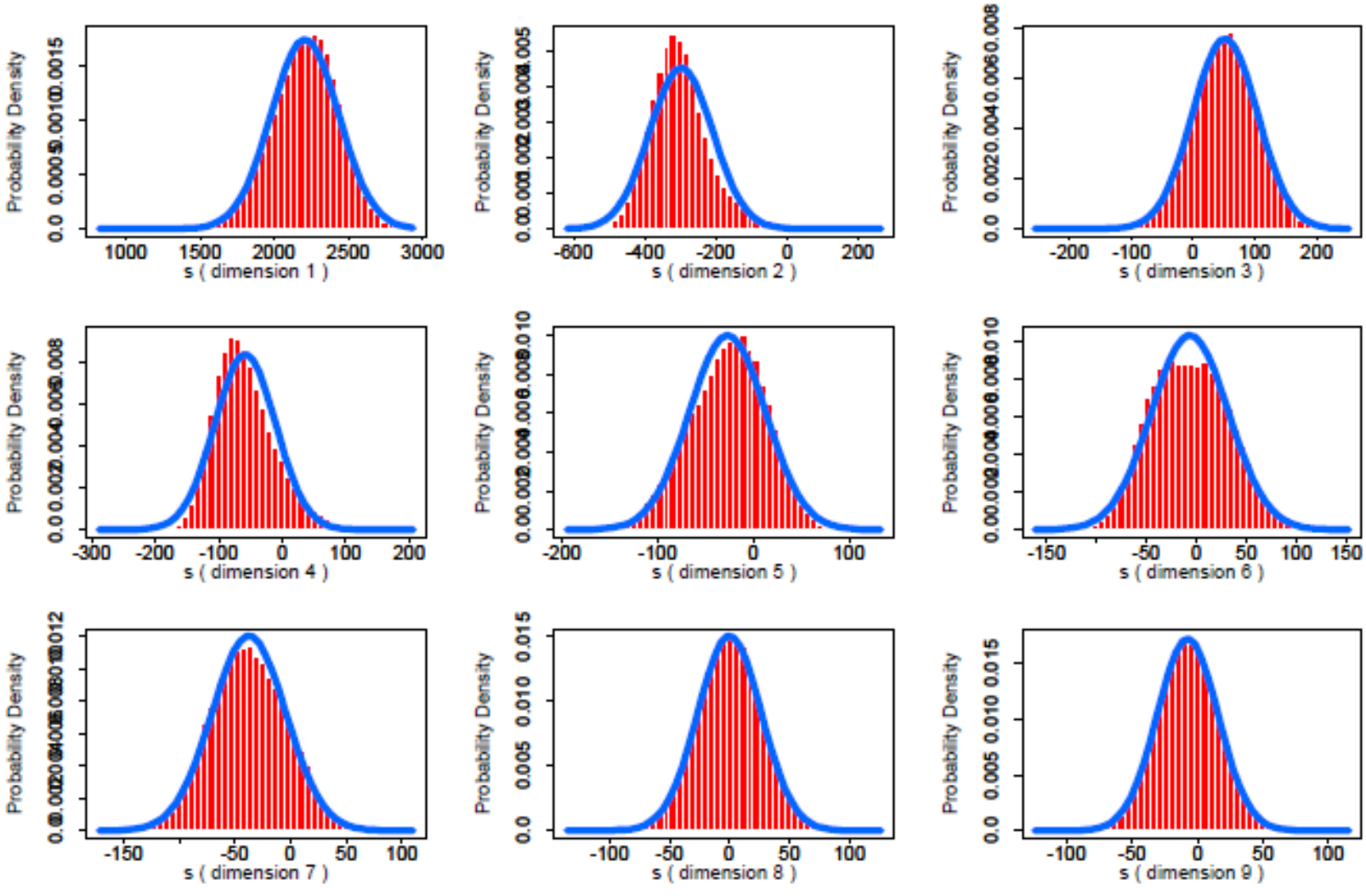
- For weather-mood HMM, we had a vector of probabilities for each state
- Makes sense when observations are *discrete* variables
- But speech observations are continuous-valued multidimensional feature vectors!
- Two choices:
  - Use vector quantization (VQ) to convert feature vectors to discrete scalars via clustering (see Tutorial 2)
  - (More common) Use a continuous density for the observation distribution

HMMs with continuous observation distributions

- In a *continuous-density* HMM (CD-HMM), the discrete observation probabilities, $b_i(k)$, are replaced by continuous densities $b_i(\mathbf{o})$
- E.g., the observation distribution can be a Gaussian or mixture of Gaussians: $b_i(\mathbf{o}) = \sum_{k=1}^{K} c_{ik}\mathcal{N}(\mathbf{o}|\mu_{ik}, \mathbf{\Sigma}_{ik}), \;\; 1 \leq i \leq N$
- $c_{ik}, \mu_{ik}, \mathbf{\Sigma}_{ik}$ are the parameters of Gaussian component k in state i
- Doesn't change anything in the algorithms; just replace $b_i(k)$ by the value of the corresponding density

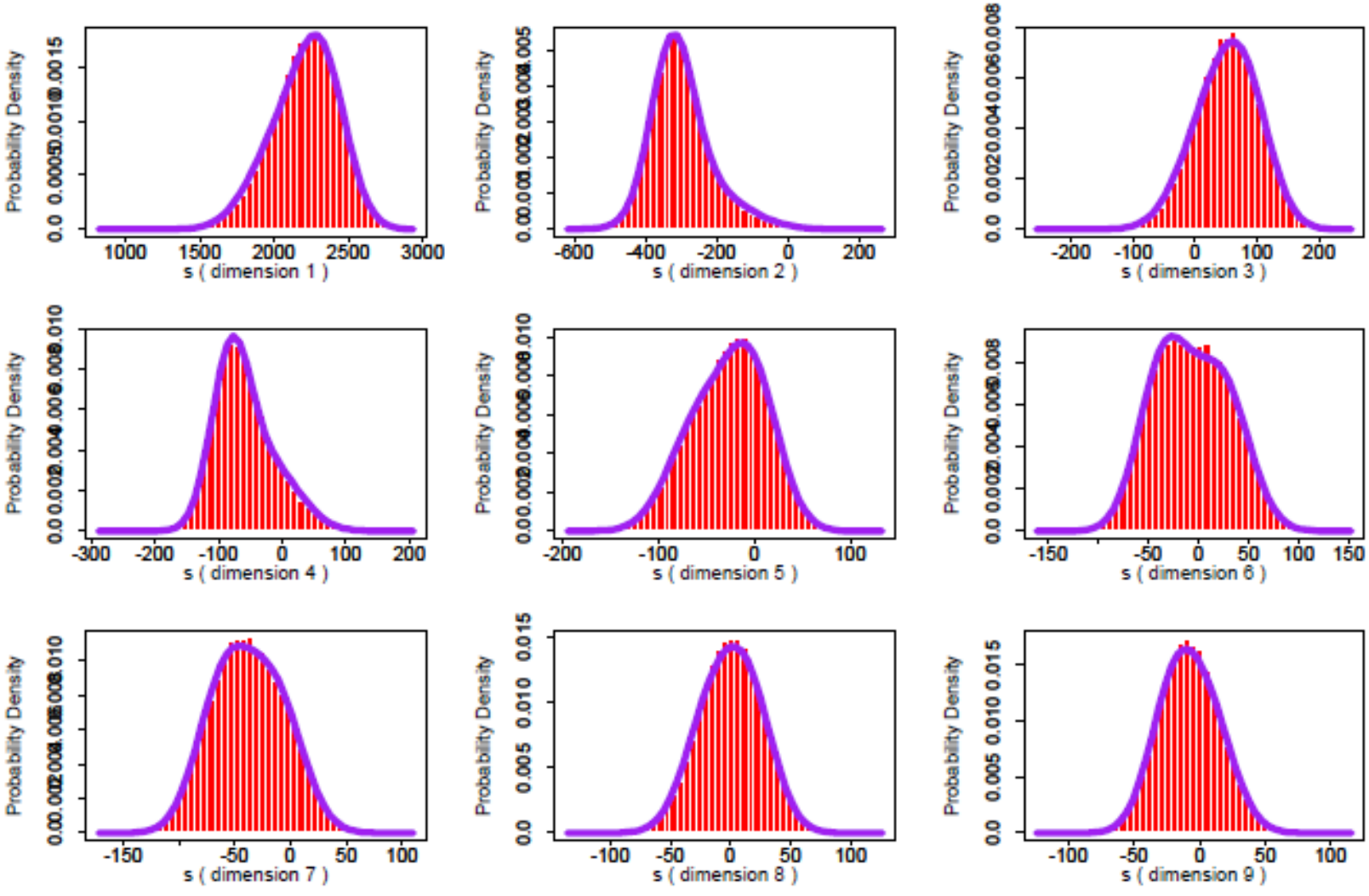
Example: Gaussian density for MFCCs

First 9 MFCC's from [s]: Gaussian PDF

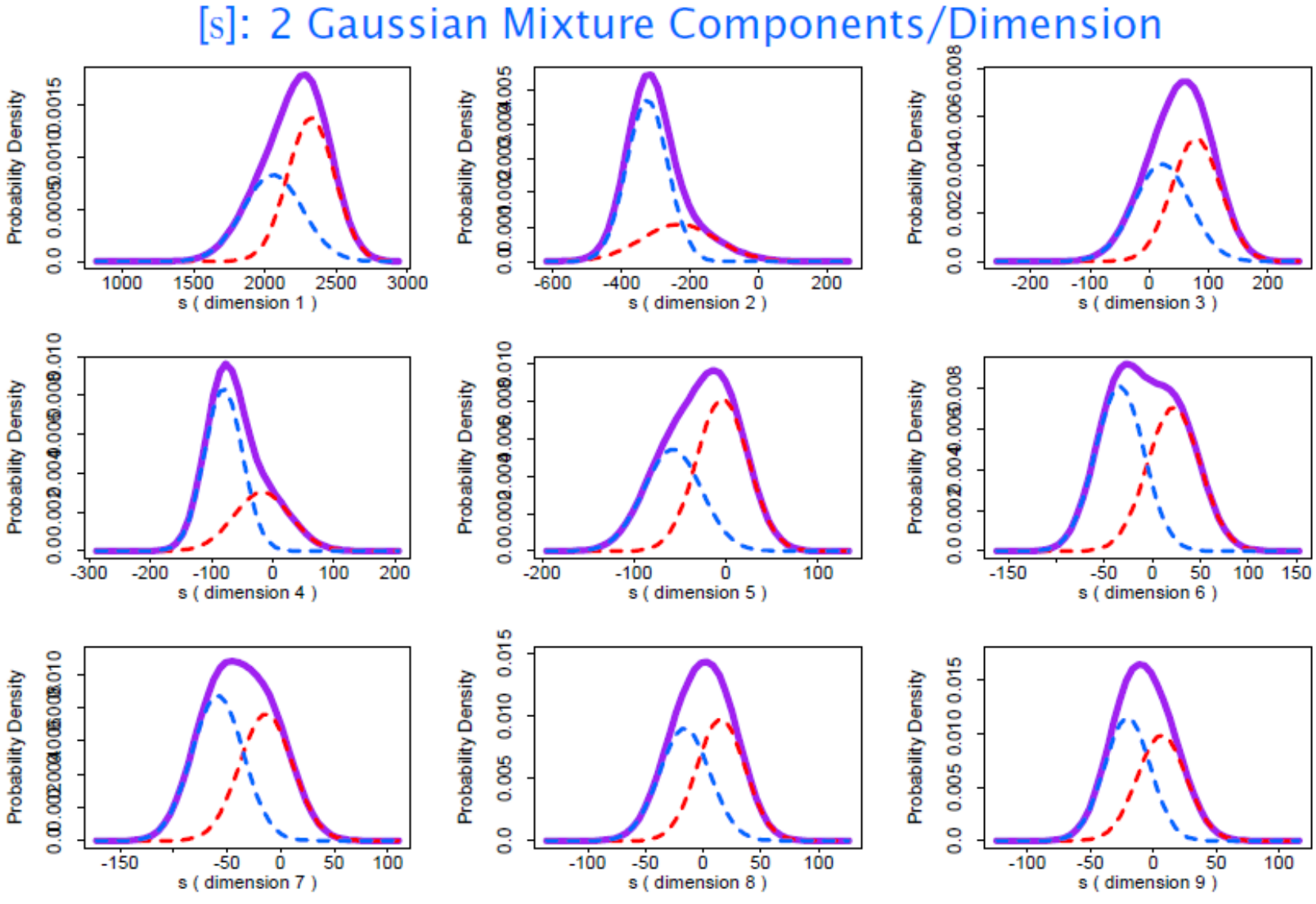


Example: Gaussian density for MFCCs

[s]: 2 Gaussian Mixture Components/Dimension



Example: Gaussian density for MFCCs



Outline

HMM problem 2: Decoding

- Given an observation sequence $\mathbf{O}$, what is the most likely state sequence $\mathbf{q}^* = \{q_1^*, \ldots, q_T^*\}$?
- Consider this criterion: Choose the states $\{q_1, \ldots, q_T\}$ that are *individually* most probable given $\mathbf{O}$
- Define $\gamma_t(i)$ as the probability of being in state i at time t, given the observation sequence:
  $\gamma_t(i) = P(q_t = i|\mathbf{O}, \lambda)$
- Then the individually most likely state q_t at time t is:
  $q_t = \underset{1 \le i \le N}{\operatorname{argmax}} \gamma_t(i), \quad 1 \le t \le T$
- It is easy to show that $\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{p(\mathbf{O}|\lambda)}$

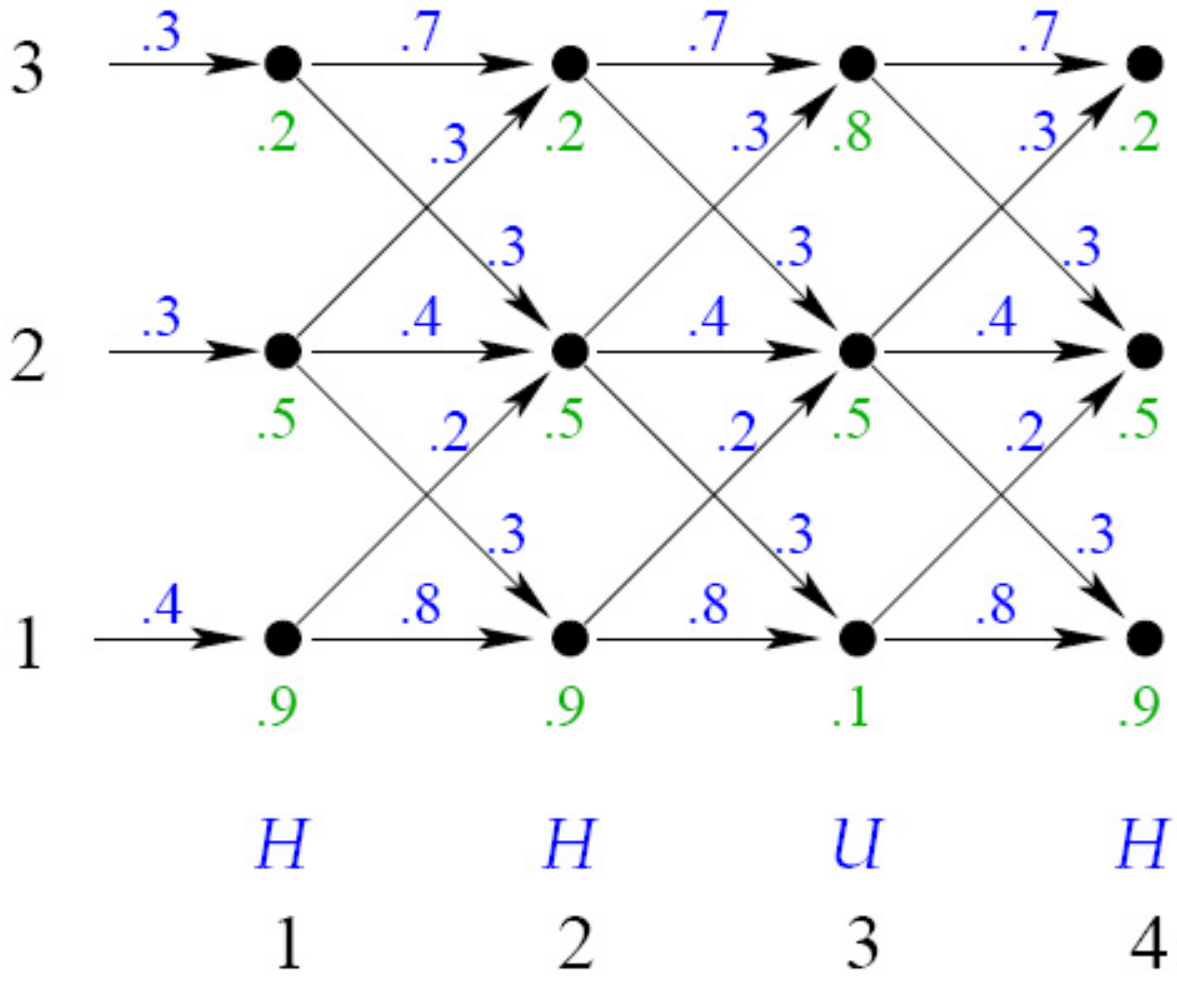
HMM problem 2: Decoding (2)

- The "individually most likely" criterion maximizes the expected number of correct states
- But it can give very unlikely (or impossible!) state sequences
- Another criterion: Choose the state sequence that maximizes $p(\mathbf{q}|\mathbf{O}, \lambda)$ (or, equivalently, that maximizes $p(\mathbf{q}, \mathbf{O}|\lambda)$)
- This can be done via the Viterbi algorithm
  - A dynamic programming algorithm
  - Analogous to DTW algorithm, but with probabilities

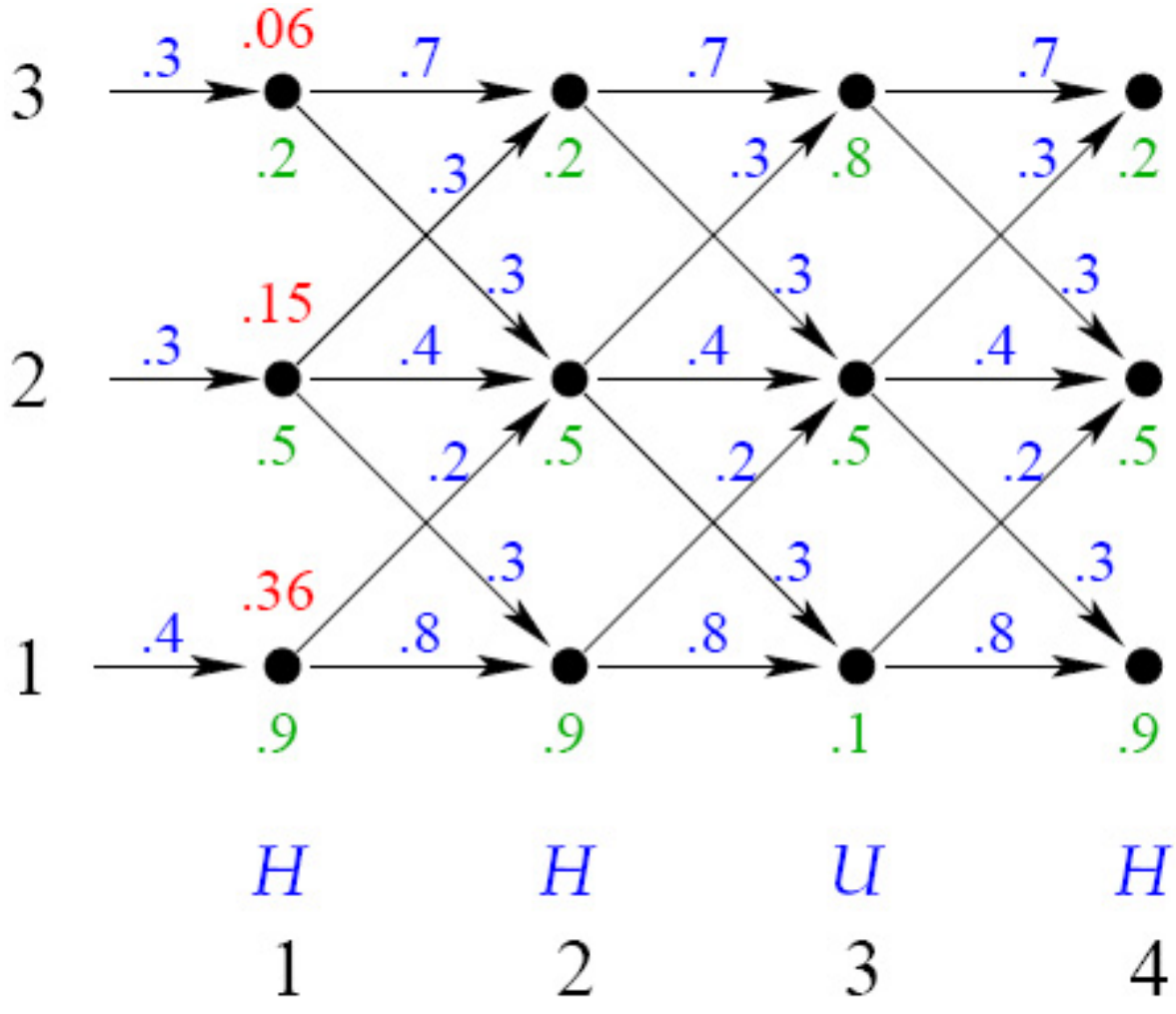
The Viterbi algorithm

- Define $\delta_t(i)$ as the highest probability along a single path to state i at time t that accounts for the first t observations:
  $\delta_t(i) = \max_{q_1,\cdots,q_{t-1}} P(q_1 q_2 \ldots q_t = i, \mathbf{o}_1 \mathbf{o}_2 \ldots \mathbf{o}_t | \lambda)$
- Initialization: $\delta_1(i) = \pi_i b_i(\mathbf{o}_1)$ (Note: this $= \alpha_1(i)$)
- Recursion: $\delta_t(j) = \max_{1 \le i \le N} [\delta_{t-1}(i) a_{ij}] \, b_j(\mathbf{o}_t), \quad 2 \le t \le T$
- Note the similarity to the α recursion:
  $$\alpha_t(j) = \left[\sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} \right] b_j(\mathbf{o}_t)$$
  - Same as forward algorithm, but sum replaced by max!
  - To retrieve the best path, also keep a back-pointer from each node to best incoming partial path (as in DTW)
- As in forward algorithm, computation is $\approx N^2 T$

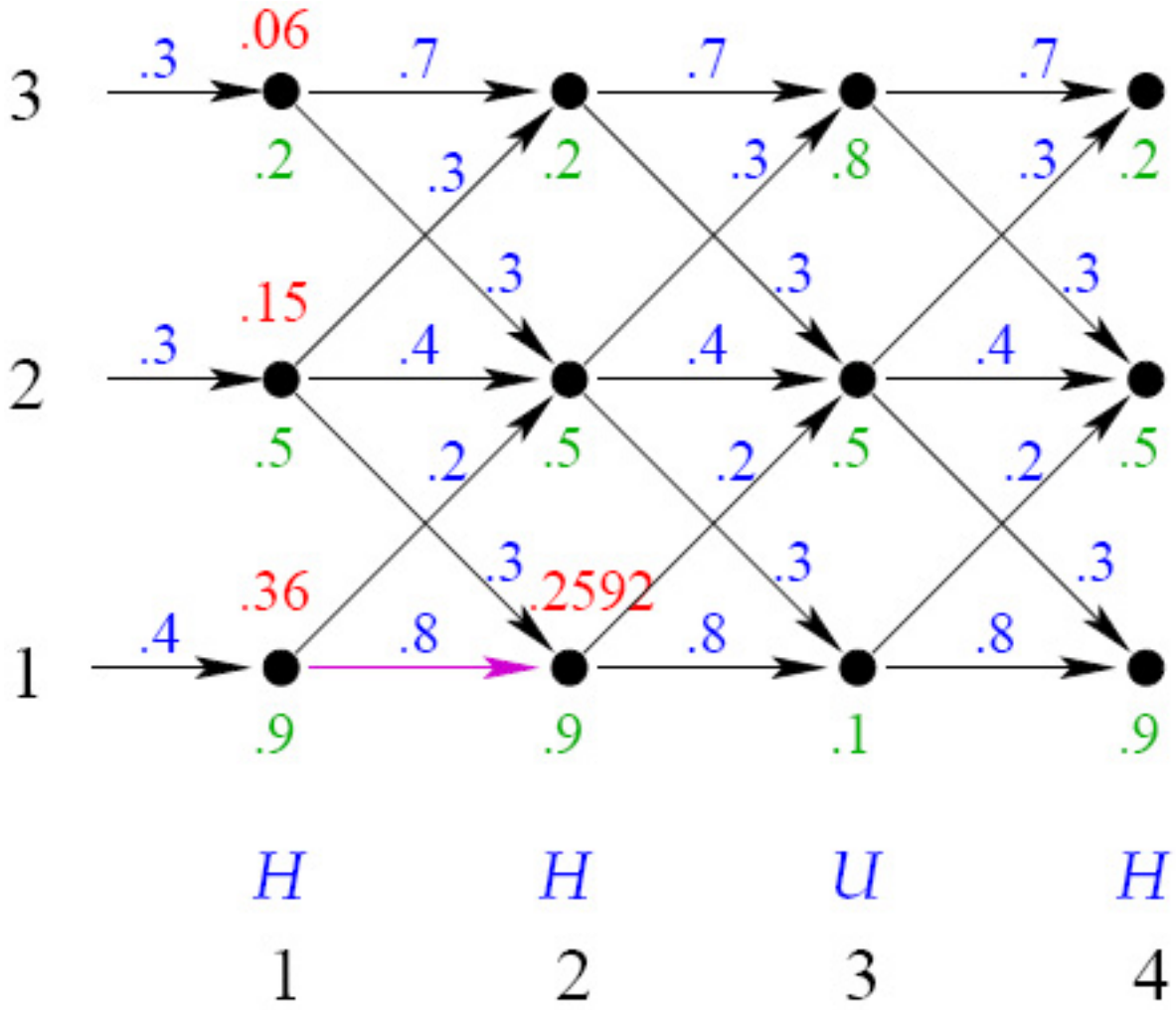
The Viterbi algorithm: Example



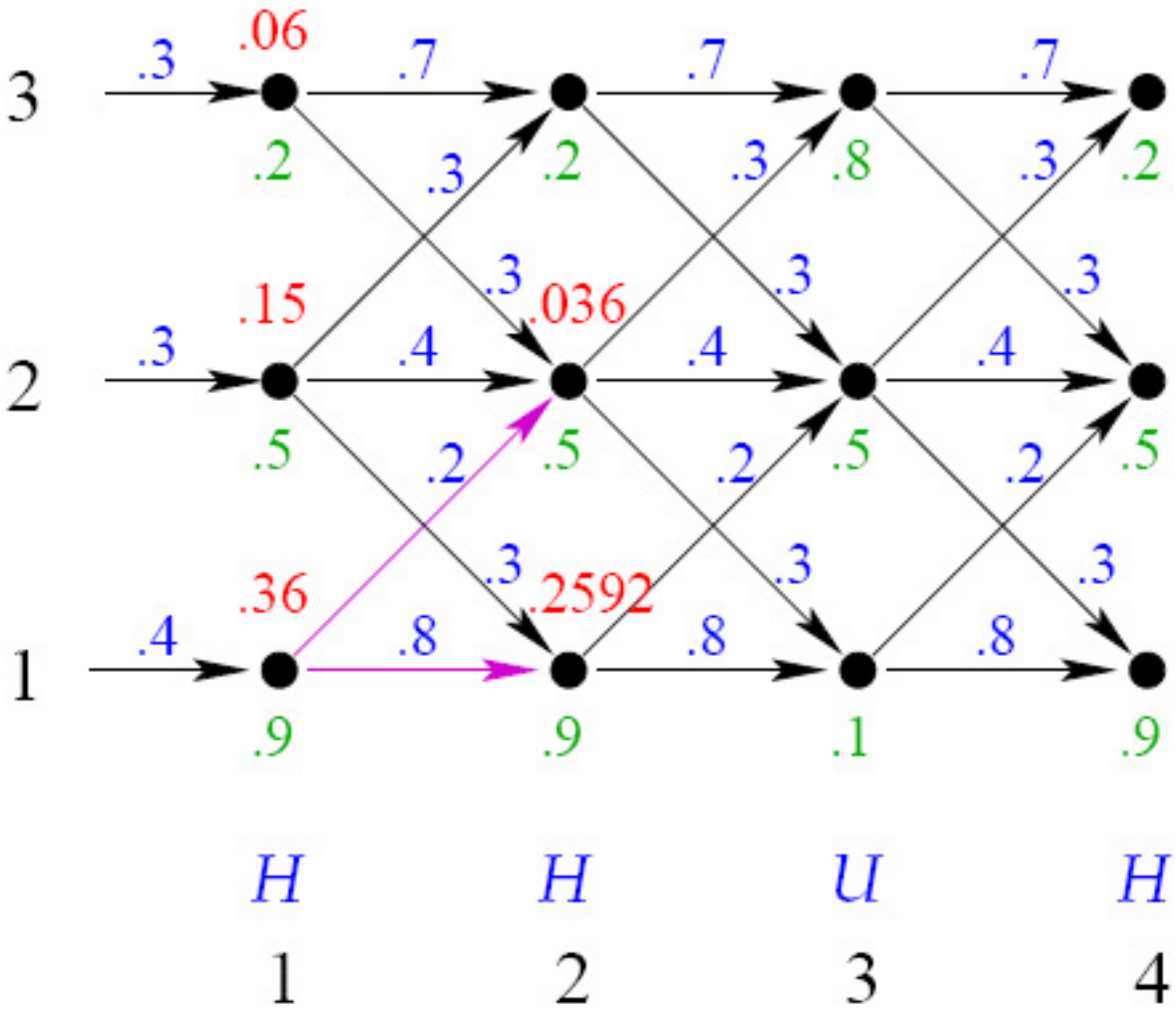
The Viterbi algorithm: Example



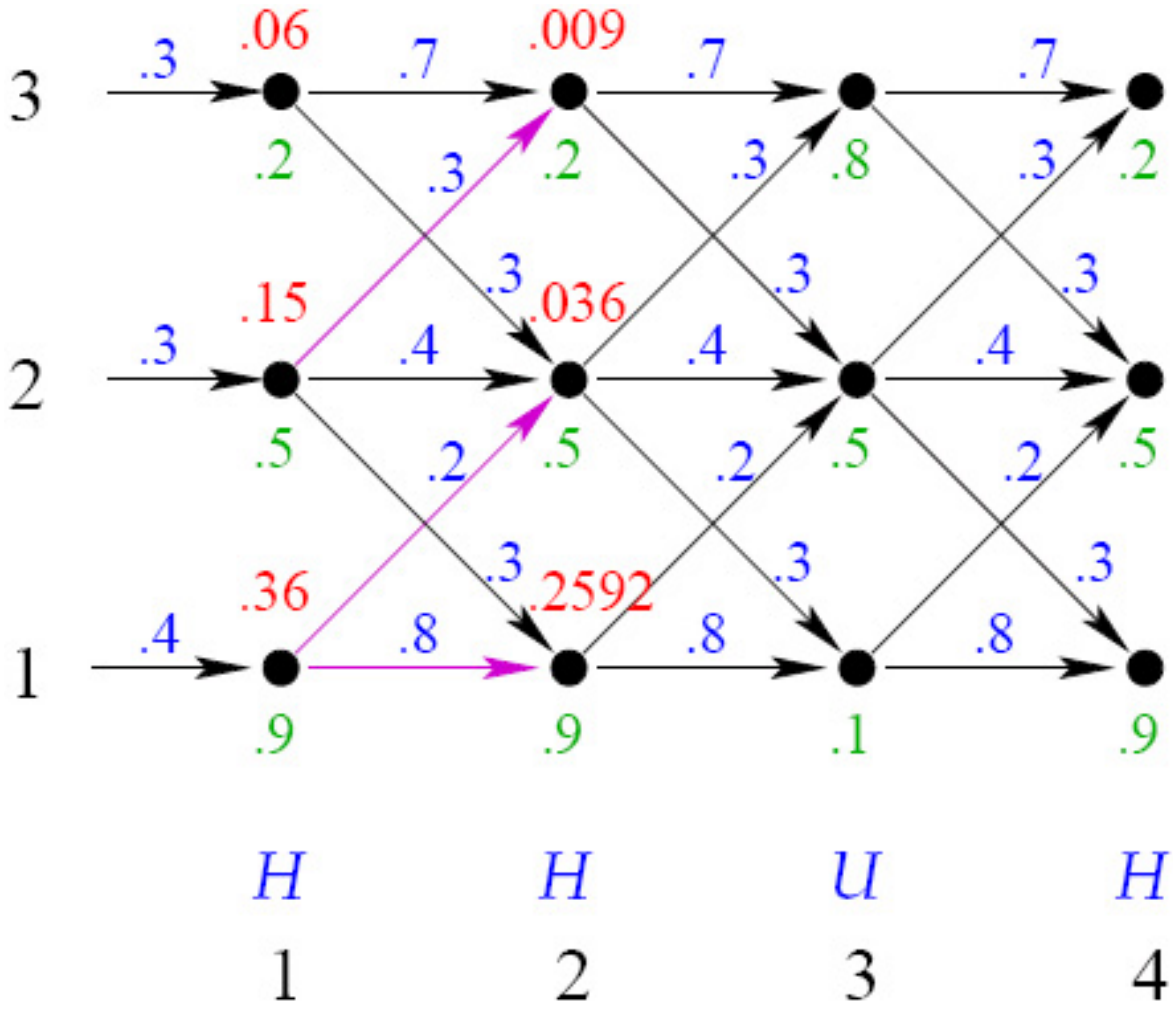
The Viterbi algorithm: Example



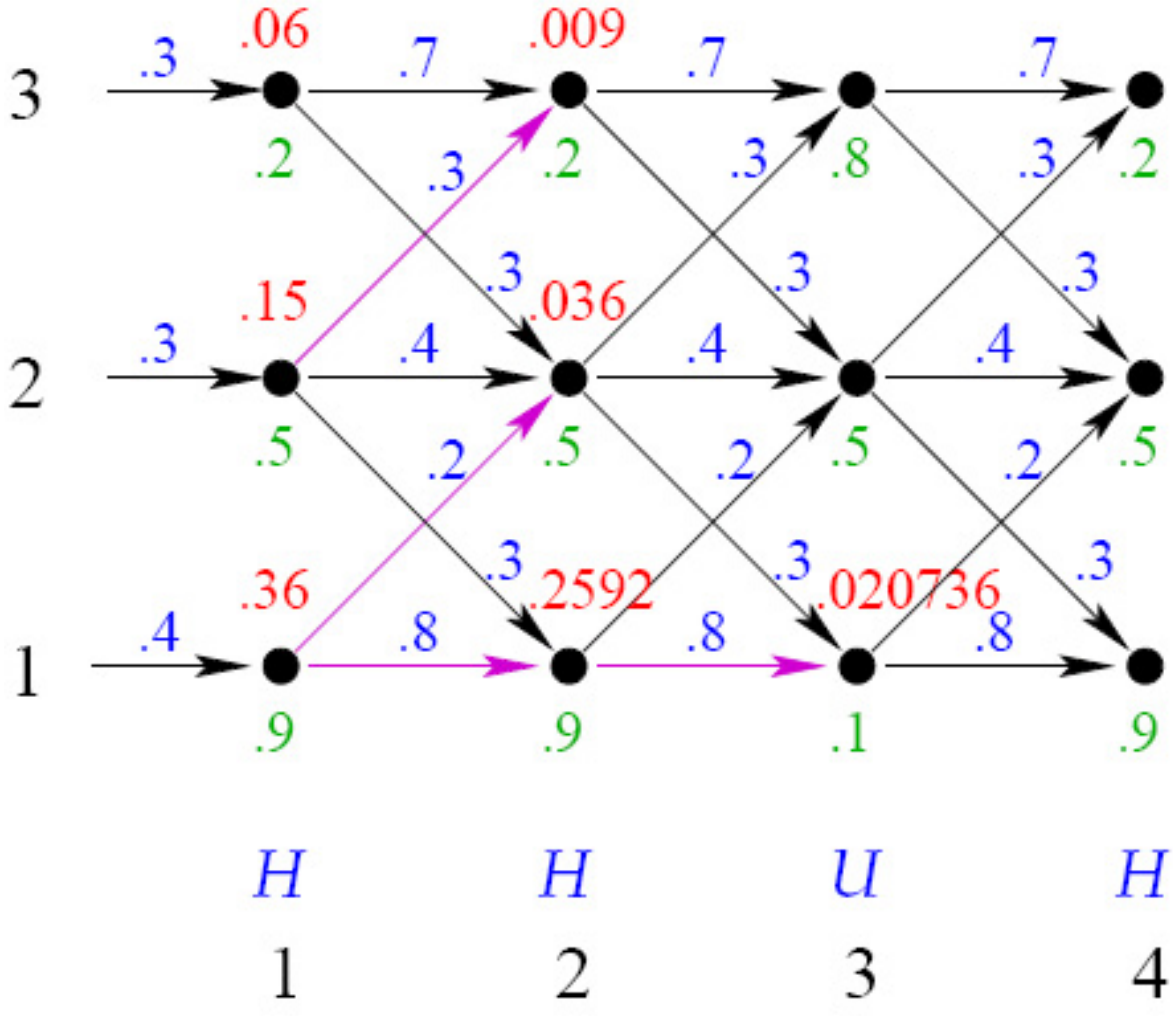
The Viterbi algorithm: Example



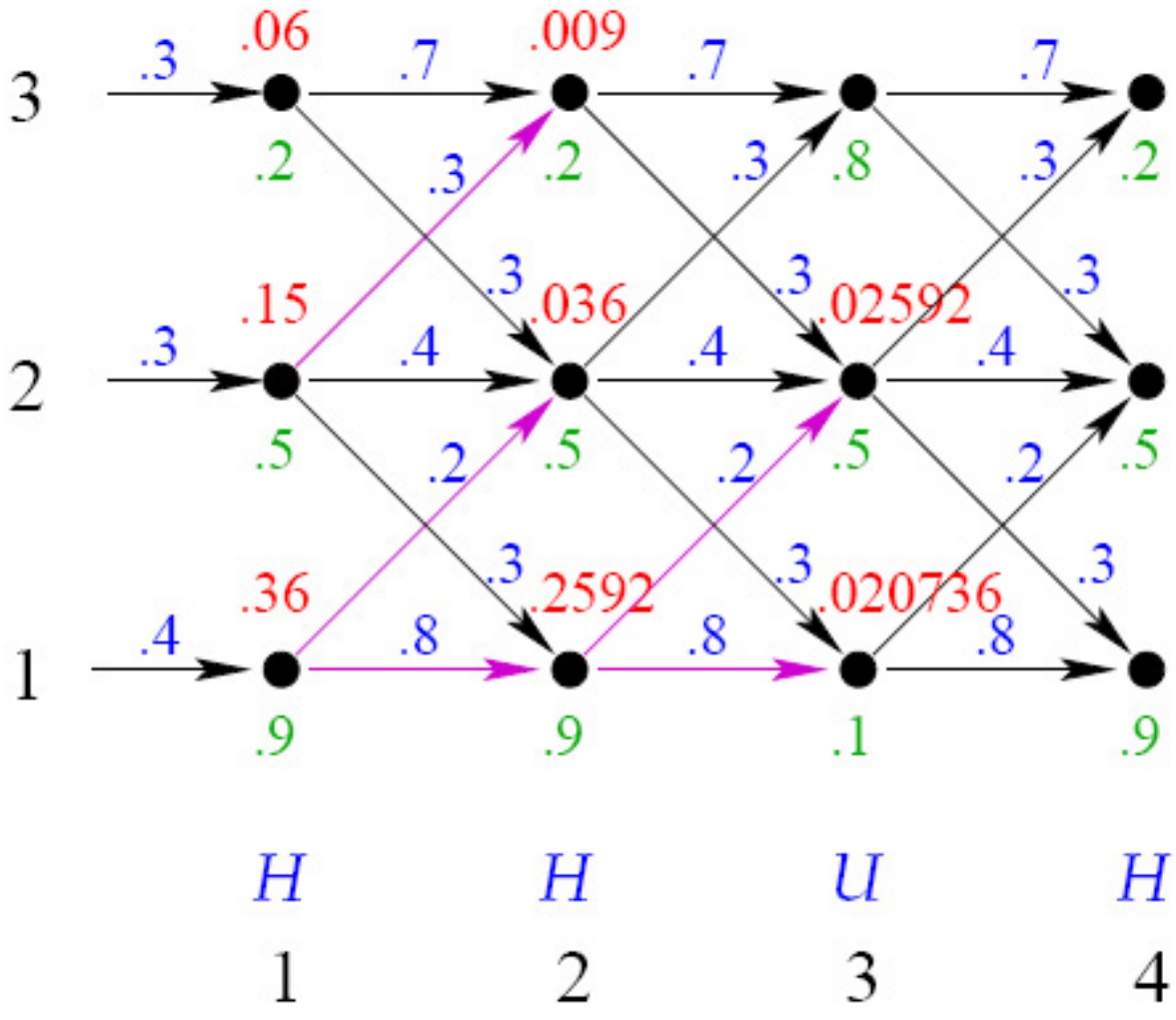
The Viterbi algorithm: Example



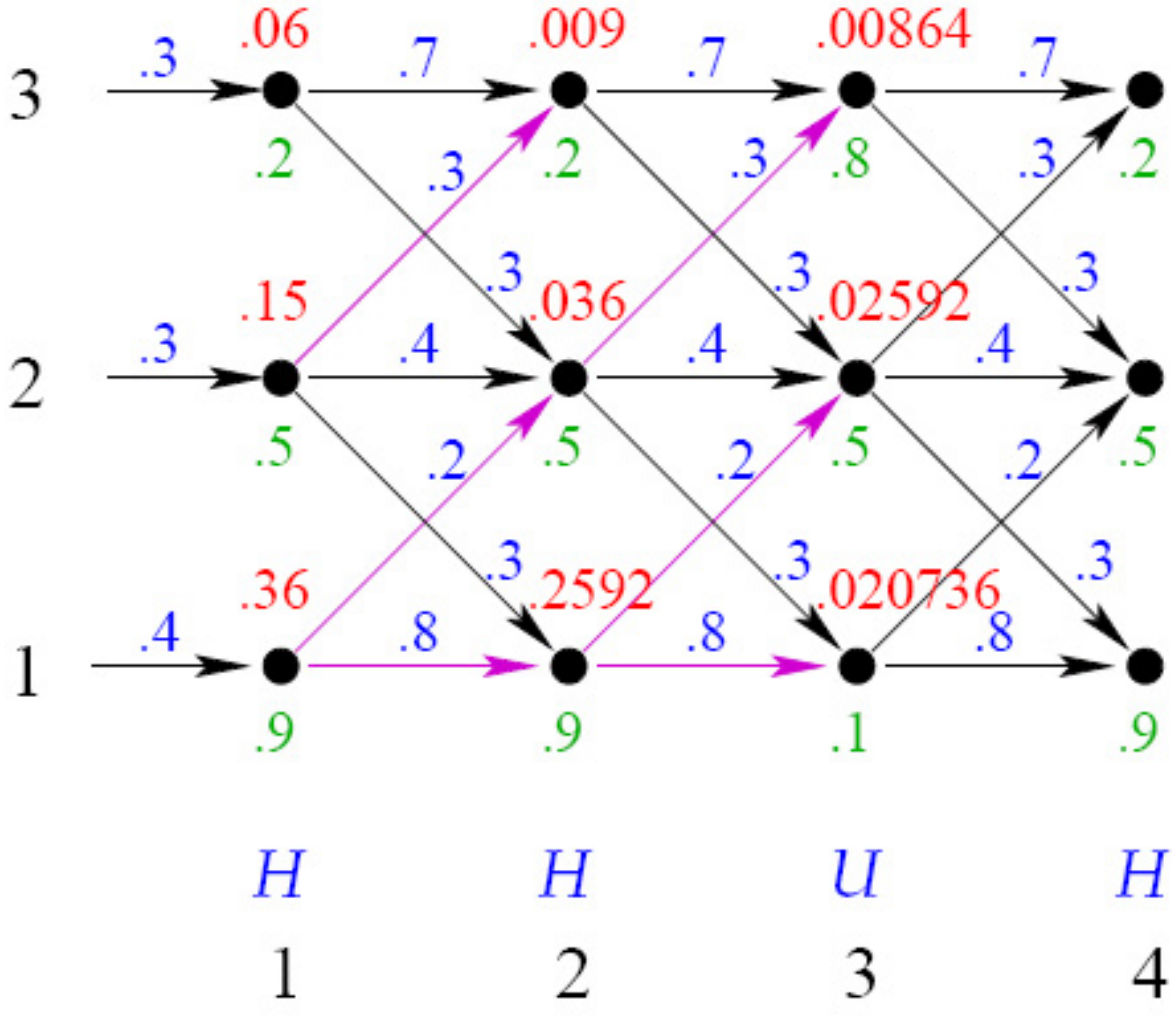
The Viterbi algorithm: Example



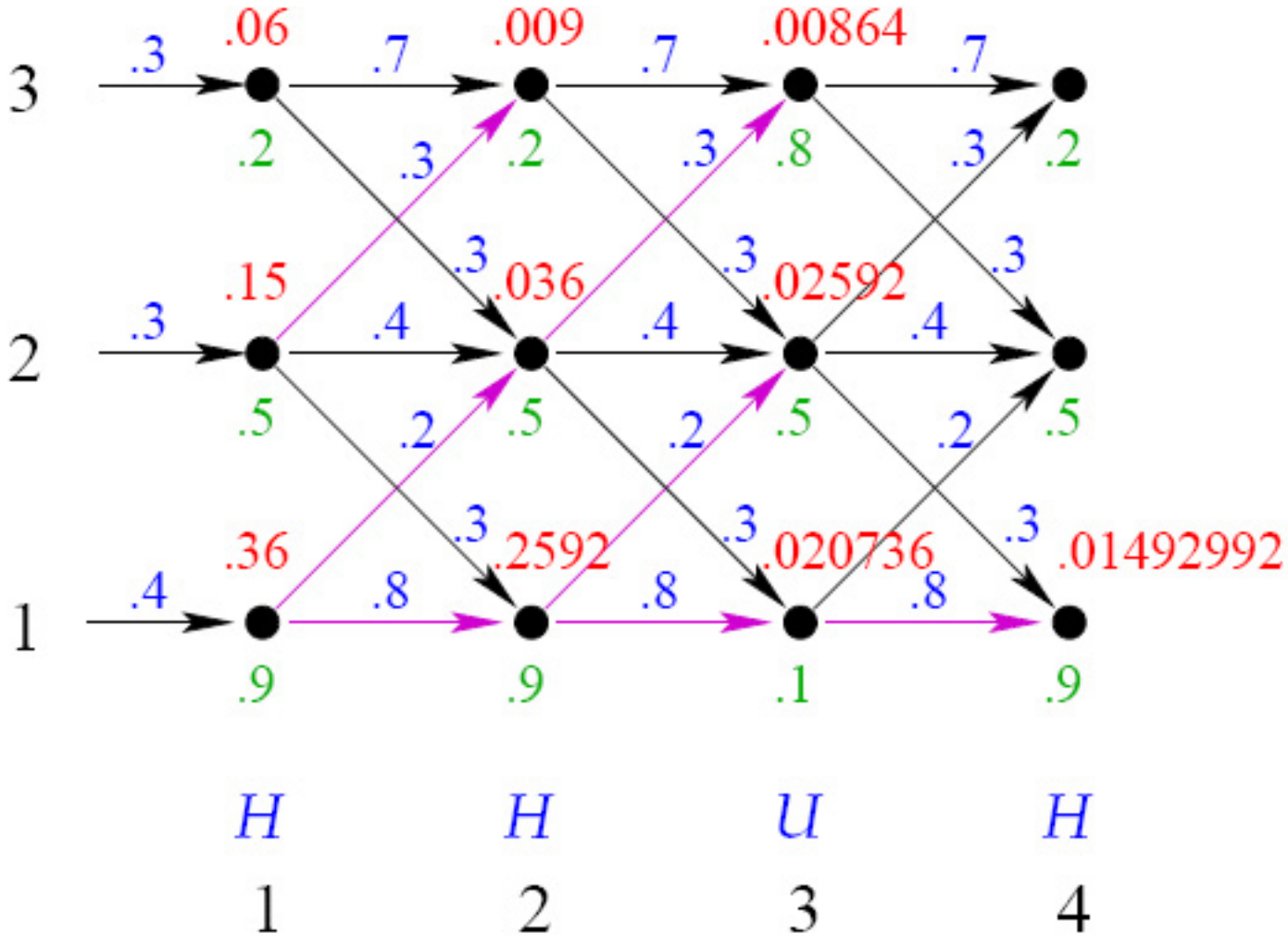
The Viterbi algorithm: Example



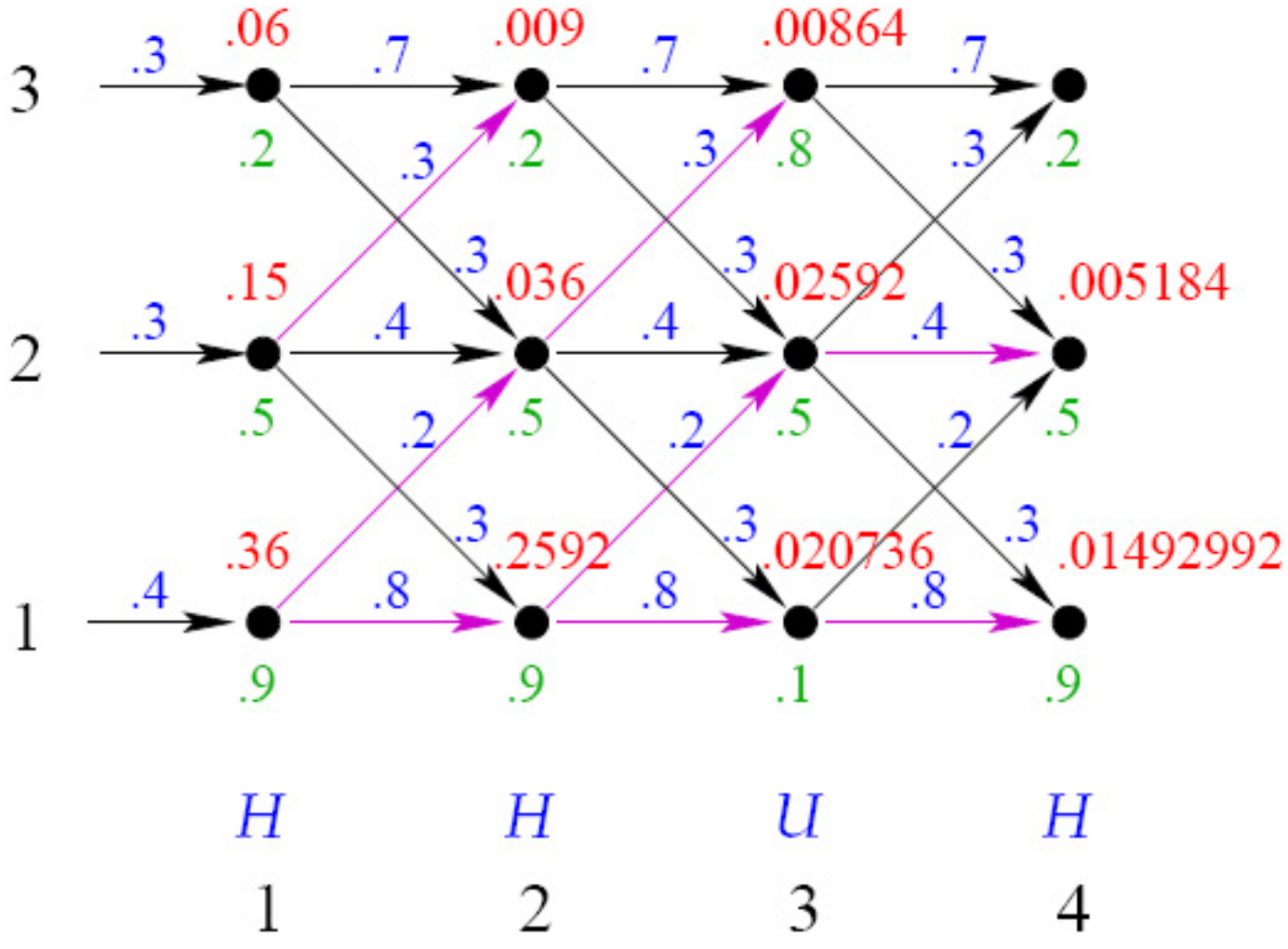
The Viterbi algorithm: Example



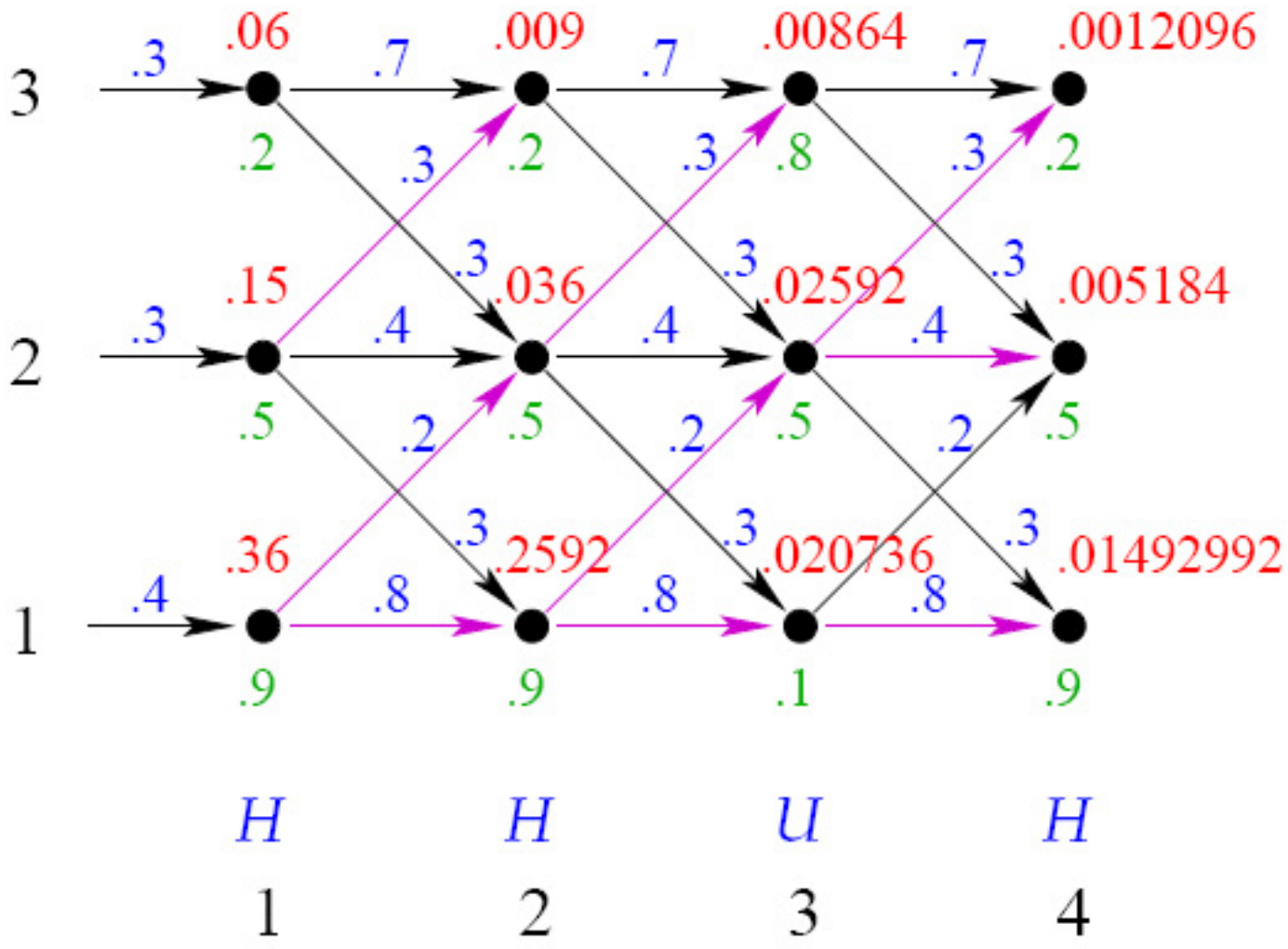
The Viterbi algorithm: Example



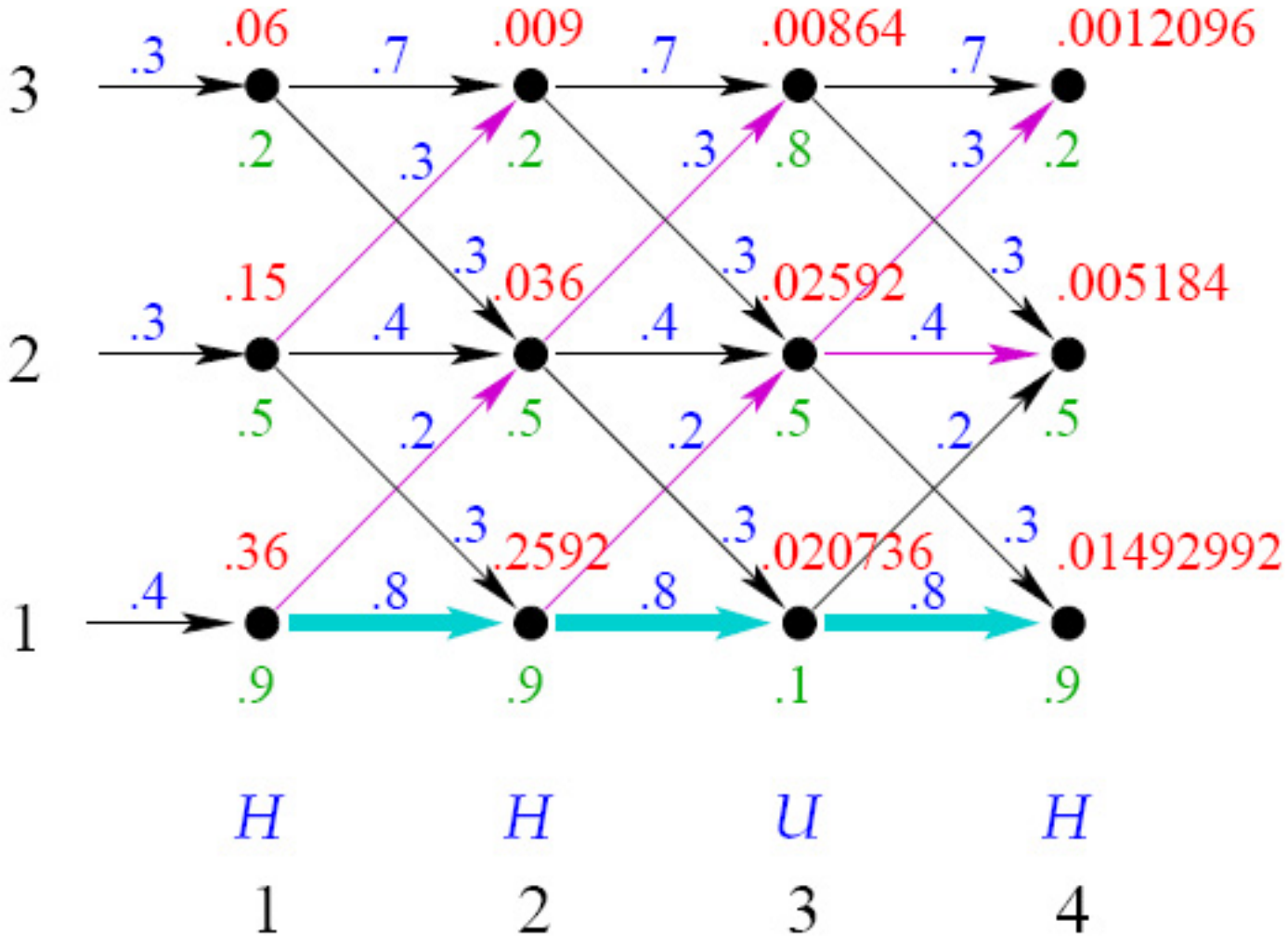
The Viterbi algorithm: Example



The Viterbi algorithm: Example



The Viterbi algorithm: Example



Summary

So far:

- Problem 1 (scoring): Find probability of a given observation sequence, given a model $\lambda = \{\pi, \mathbf{A}, \mathbf{B}\}$. Sufficient to do isolated-word (or isolated-phone) recognition
- Problem 2 (decoding): Find most likely state sequence

Independence properties

An HMM is a model of the joint distribution of $\mathbf{q}$ and $\mathbf{O}$:

$$\begin{aligned} p(\mathbf{O}, \mathbf{q}) = p(\mathbf{O}_{1:T}, \mathbf{q}_{1:T}) &= p(q_1)p(\mathbf{o}_1|q_1)\prod_{t=2}^{T} p(q_t|q_{t-1})p(\mathbf{o}_t|q_t) \\ &= \pi_{q_1} b_{q_1}(\mathbf{o}_q)\prod_{t=2}^{T} a_{q_{t-1}q_t} b_{q_t}(\mathbf{o}_t) \end{aligned}$$

What independence properties does this imply?

Independence properties

An HMM is a model of the joint distribution of $\mathbf{q}$ and $\mathbf{O}$:

$$p(\mathbf{O}, \mathbf{q}) = p(\mathbf{O}_{1:T}, \mathbf{q}_{1:T}) = p(q_1)p(\mathbf{o}_1|q_1)\prod_{t=2}^{T} p(q_t|q_{t-1})p(\mathbf{o}_t|q_t)$$

$$= \pi_{q_1} b_{q_1}(\mathbf{o}_q) \prod_{t=2}^{T} a_{q_{t-1}q_t} b_{q_t}(\mathbf{o}_t)$$

What independence properties does this imply?

- Notation: $X \perp Y \Rightarrow X$ is independent of Y
- Notation: $X \perp \{Y, Z\} \Rightarrow X$ is independent of Y and Z
- Notation: $X \perp Y|W \Rightarrow X$ is independent of Y given W

Independence properties

An HMM is a model of the joint distribution of $\mathbf{q}$ and $\mathbf{O}$:

$$p(\mathbf{O}, \mathbf{q}) = p(\mathbf{O}_{1:T}, \mathbf{q}_{1:T}) = p(q_1)p(\mathbf{o}_1|q_1)\prod_{t=2}^{T} p(q_t|q_{t-1})p(\mathbf{o}_t|q_t)$$

$$= \pi_{q_1} b_{q_1}(\mathbf{o}_q)\prod_{t=2}^{T} a_{q_{t-1}q_t} b_{q_t}(\mathbf{o}_t)$$

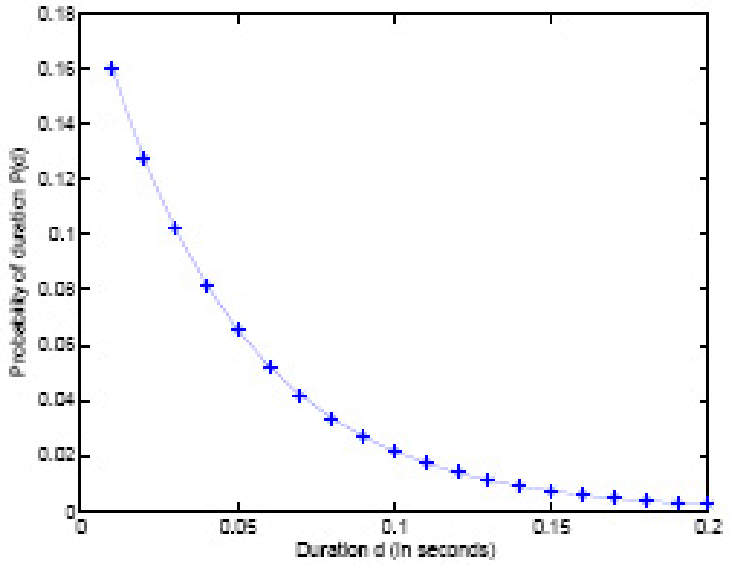
What independence properties does this imply?

- Notation: $X \perp Y \Rightarrow X$ is independent of Y
- Notation: $X \perp \{Y, Z\} \Rightarrow X$ is independent of Y and Z
- Notation: $X \perp Y|W \Rightarrow X$ is independent of Y given W
- $\mathbf{o}_t \perp \{\mathbf{q}_{\neg t}, \mathbf{O}_{\neg t}\}|q_t$, where $\mathbf{q}_{\neg t} = \{q_1, \ldots, q_{t-1}, q_{t+1}, \ldots, q_T\}$, i.e. the current observation only depends on the current state
- $\{\mathbf{q}_{1:t-1}, \mathbf{O}_{1:t-1}\} \perp \{\mathbf{q}_{t+1:T}, \mathbf{O}_{t+1:T}\}|q_t$, i.e. the future is independent of the past given the current state

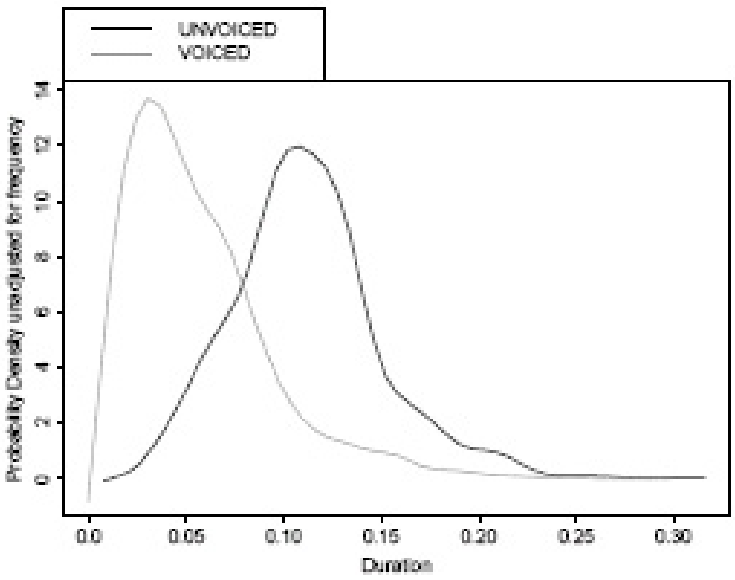
State duration

- The probability of staying in state i for d time steps (frames) is $a_{ii}^{d-1}(1 - a_{ii})$, a geometric distribution
- In contrast, phone durations look nothing like that:

HMM state duration probabilities, $a_{ii} = .8$



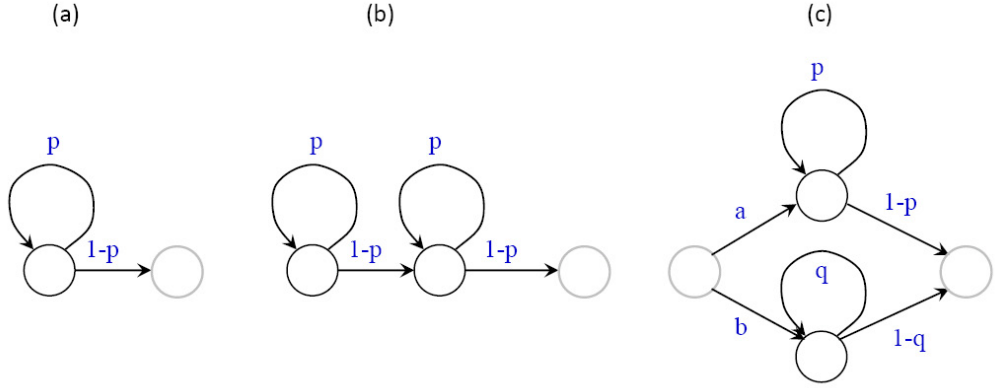
Actual fricative duration probabilities



State duration modeling

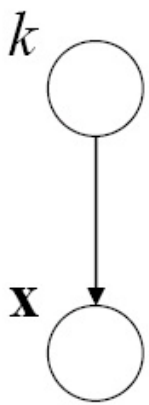
Possible solutions:

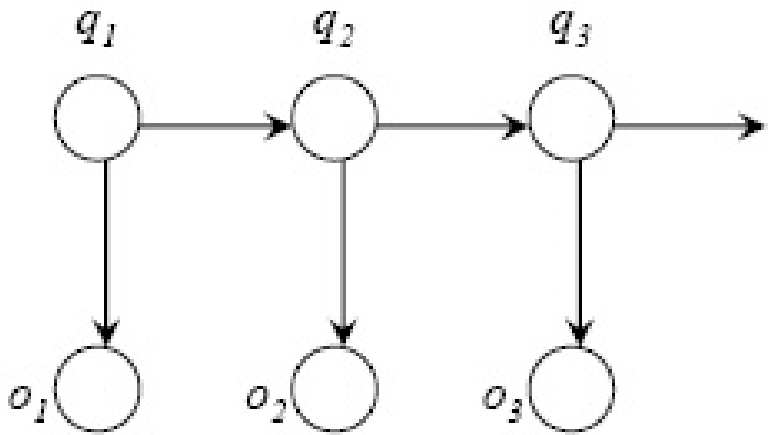
- Replace transition probabilities with explicit duration distributions [Ostendorf '96]
  - Most general approach
  - Breaks dynamic programming algorithms
- Replace single state with multiple *tied states* with identical observation distribution, and some desired transition structure
  - Simplest option: Set a minimum duration
  - Can simulate other duration distributions with transition structures



HMMs as extension of GMMs

- We've argued why HMMs are an extension of DTW
- They are also an extension of GMMs, where observation distributions are Gaussian and "components" are not independent across data points
- Graphical model representation (*not transition diagrams*!) for GMMs and HMMs:





Handling underflow in HMM computations

- The algorithms as we've described them don't actually work...
  - Path probability for a 10-second waveform $\approx .5^{1000} \approx 9 \times 10^{-302}$
  - Path probability for a 20-second waveform $\approx .5^{2000} \approx 0$

Handling underflow in HMM computations

- The algorithms as we've described them don't actually work...
  - Path probability for a 10-second waveform $\approx .5^{1000} \approx 9 \times 10^{-302}$
  - Path probability for a 20-second waveform $\approx .5^{2000} \approx 0$
- Preventing underflow in HMM computations
  - Viterbi decoding: Only multiplications involved; can sum logs instead of multiplying probabilities
  - Forward-backward: Both multiplications and additions involved; scale the probabilities at each time frame, and keep track of scale factors