Language and Statistics II

Lecture 7: Forward-Backward

Quick Review

- Markov/(m+1)-gram models
- Can be a source model (e.g., ASR) or a channel model (e.g., text categorization)
- (Weighted) lattices and (m+1)-gram models
  - Finding the best path
- Adding classes deterministically (Brown et al., 1990) and stochastically (HMMs)

Important, Often Missed Point

- Why stopping probabilities?

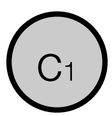
Hidden Markov Model

N-gram model
over states
(bigram shown)

Hidden Markov Model

$$\gamma(c_1 \mid \emptyset)$$

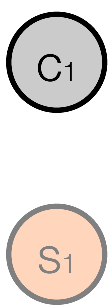
N-gram model
over states
(bigram shown)



Hidden Markov Model

$\gamma(c_1 \mid \emptyset) \qquad \times \eta(s_1 \mid c_1)$

N-gram model over states (bigram shown)

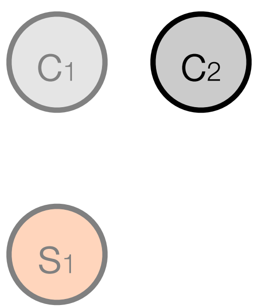


Hidden Markov Model

$$\gamma(c_1 \mid \emptyset) \quad \times \eta(s_1 \mid c_1)$$

$$\times \gamma(c_2 \mid c_1)$$

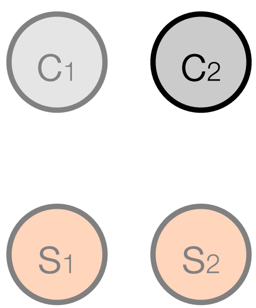
N-gram model over states (bigram shown)



Hidden Markov Model

$$
\begin{aligned}
&\gamma(c_1 \mid \emptyset) && \times \eta(s_1 \mid c_1) \\
&\quad \times \gamma(c_2 \mid c_1) && \times \eta(s_2 \mid c_2)
\end{aligned}
$$

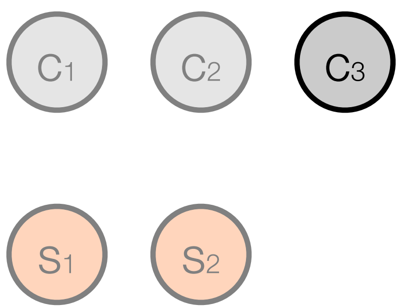
N-gram model over states (bigram shown)



Hidden Markov Model

$$
\begin{aligned}
&\gamma(c_1 \mid \emptyset) && \times\eta(s_1 \mid c_1) \\
&\quad\times\gamma(c_2 \mid c_1) && \times\eta(s_2 \mid c_2) \\
&\quad\times\gamma(c_3 \mid c_2) &&
\end{aligned}
$$

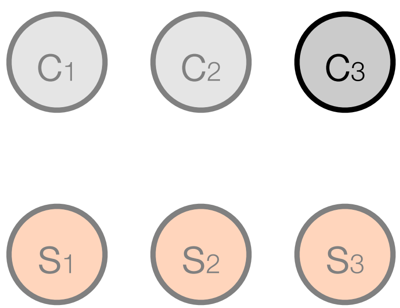
N-gram model over states (bigram shown)



Hidden Markov Model

$$\begin{aligned}
&\gamma(c_1 \mid \emptyset) && \times\eta(s_1 \mid c_1) \\
&\quad\times\gamma(c_2 \mid c_1) && \times\eta(s_2 \mid c_2) \\
&\quad\times\gamma(c_3 \mid c_2) && \times\eta(s_3 \mid c_3)
\end{aligned}$$

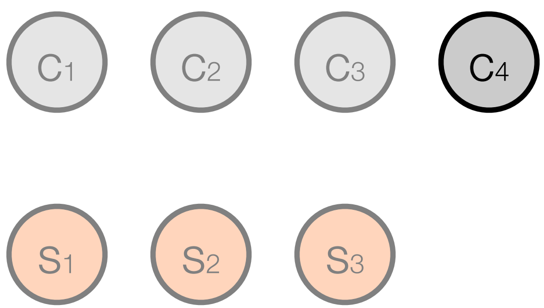
N-gram model
over states
(bigram shown)



Hidden Markov Model

$$
\begin{aligned}
&\gamma(c_1 \mid \emptyset) && \times\eta(s_1 \mid c_1) \\
&\quad\times\gamma(c_2 \mid c_1) && \times\eta(s_2 \mid c_2) \\
&\quad\times\gamma(c_3 \mid c_2) && \times\eta(s_3 \mid c_3) \\
&\quad\times\gamma(c_4 \mid c_3) &&
\end{aligned}
$$

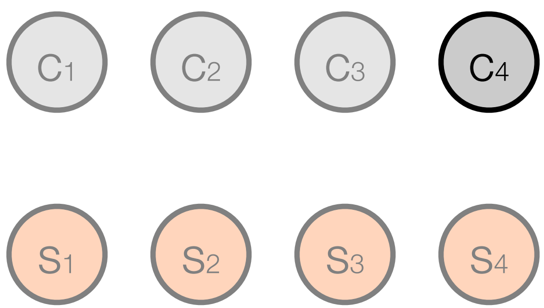
N-gram model over states (bigram shown)



Hidden Markov Model

$$
\begin{aligned}
&\gamma(c_1 \mid \emptyset) && \times\eta(s_1 \mid c_1) \\
&\quad\times\gamma(c_2 \mid c_1) && \times\eta(s_2 \mid c_2) \\
&\quad\times\gamma(c_3 \mid c_2) && \times\eta(s_3 \mid c_3) \\
&\quad\times\gamma(c_4 \mid c_3) && \times\eta(s_4 \mid c_4)
\end{aligned}
$$

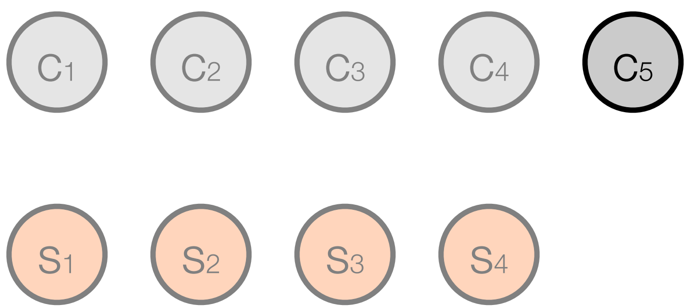
N-gram model over states (bigram shown)



Hidden Markov Model

$$
\begin{aligned}
&\gamma(c_1 \mid \emptyset) && \times \eta(s_1 \mid c_1) \\
&\quad \times \gamma(c_2 \mid c_1) && \times \eta(s_2 \mid c_2) \\
&\quad \times \gamma(c_3 \mid c_2) && \times \eta(s_3 \mid c_3) \\
&\quad \times \gamma(c_4 \mid c_3) && \times \eta(s_4 \mid c_4)
\end{aligned}
$$

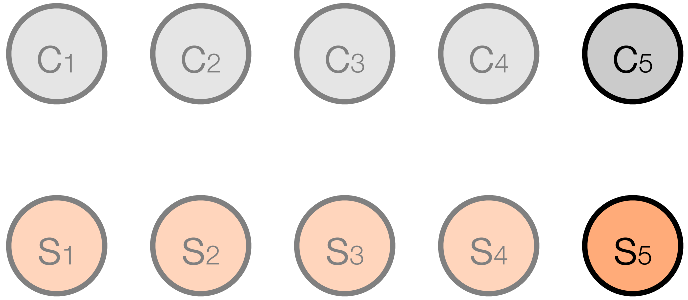
N-gram model over states (bigram shown)



Hidden Markov Model

$$\begin{aligned}
&\gamma(c_1 \mid \emptyset) && \times\eta(s_1 \mid c_1)\\
&\quad\times\gamma(c_2 \mid c_1) && \times\eta(s_2 \mid c_2)\\
&\quad\times\gamma(c_3 \mid c_2) && \times\eta(s_3 \mid c_3)\\
&\quad\times\gamma(c_4 \mid c_3) && \times\eta(s_4 \mid c_4)
\end{aligned}$$

N-gram model over states (bigram shown)



HMMs

Joint probability of **classes** and **words** is easy.

$$p(c_1^n, s_1^n) = \left(\prod_{i=1}^{n} \eta(s_i \mid c_i) \cdot \gamma(c_i \mid c_{i-m}^{i-1}) \right) \cdot \gamma(\text{stop} \mid c_{n-m+1}^n)$$

What about the marginal? Naïve algorithm: $O(2^n)$

HMMs

Joint probability of **classes** and **words** is easy.

$$p(c_1^n, s_1^n) = \left(\prod_{i=1}^{n} \eta(s_i \mid c_i) \cdot \gamma(c_i \mid c_{i-m}^{i-1}) \right) \cdot \gamma(\text{stop} \mid c_{n-m+1}^n)$$

$$p(s_1^n) = \sum_{c_1^n \in \Lambda^n} \left(\prod_{i=1}^{n} \eta(s_i \mid c_i) \cdot \gamma(c_i \mid c_{i-m}^{i-1}) \right) \cdot \gamma(\text{stop} \mid c_{n-m+1}^n)$$

What about the marginal? Naïve algorithm: $O(2^n)$

“Inference”

- noun; conclusion reached on the basis of **evidence** and reasoning
- Here, we mean **probabilistic/statistical** inference.
  - Cf. logical inference
  - Cf. automated inference systems in AI
- We know some things (evidence), and our model gives us a framework for reasoning probabilistically about the other things.

Inference with HMMs

Inference with HMMs

- Many inference problems can be solved exactly in polynomial time!
  - Unlike general graphical models (why?)
  - Dynamic programming (a.k.a. sum-product or max-product algorithms)

Inference with HMMs

- Many inference problems can be solved exactly in polynomial time!
  - Unlike general graphical models (why?)
  - Dynamic programming (a.k.a. sum-product or max-product algorithms)

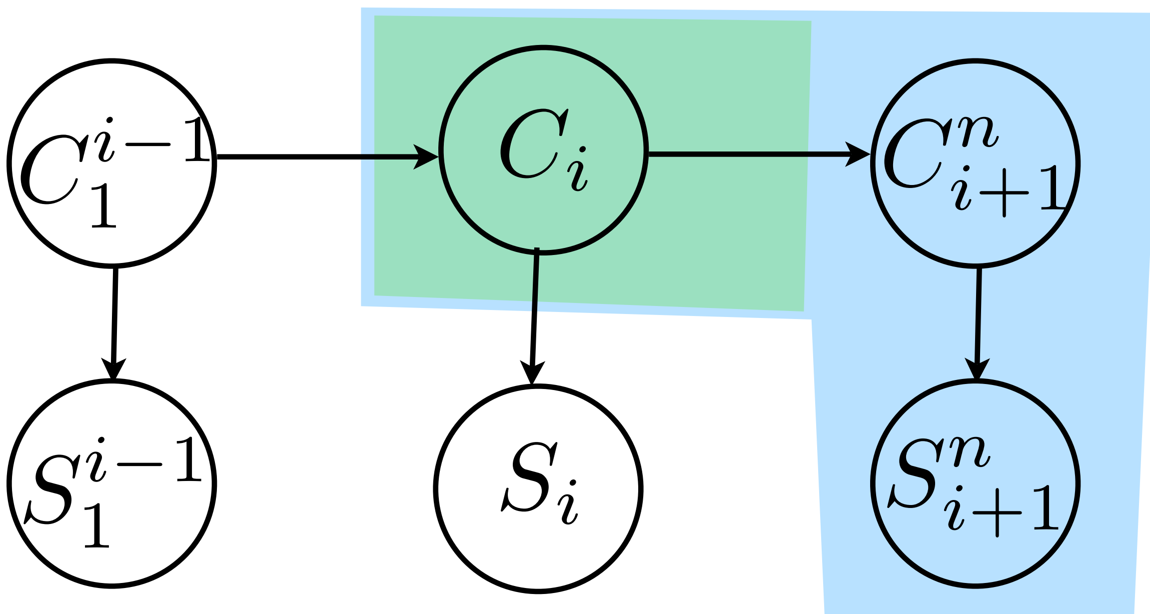
- Probability of a sequence:
  - **forward** algorithm
  - **backward** algorithm

Backward Probabilities

$$back(i, c) = p(s_{i+1}^n \mid C_i = c)$$

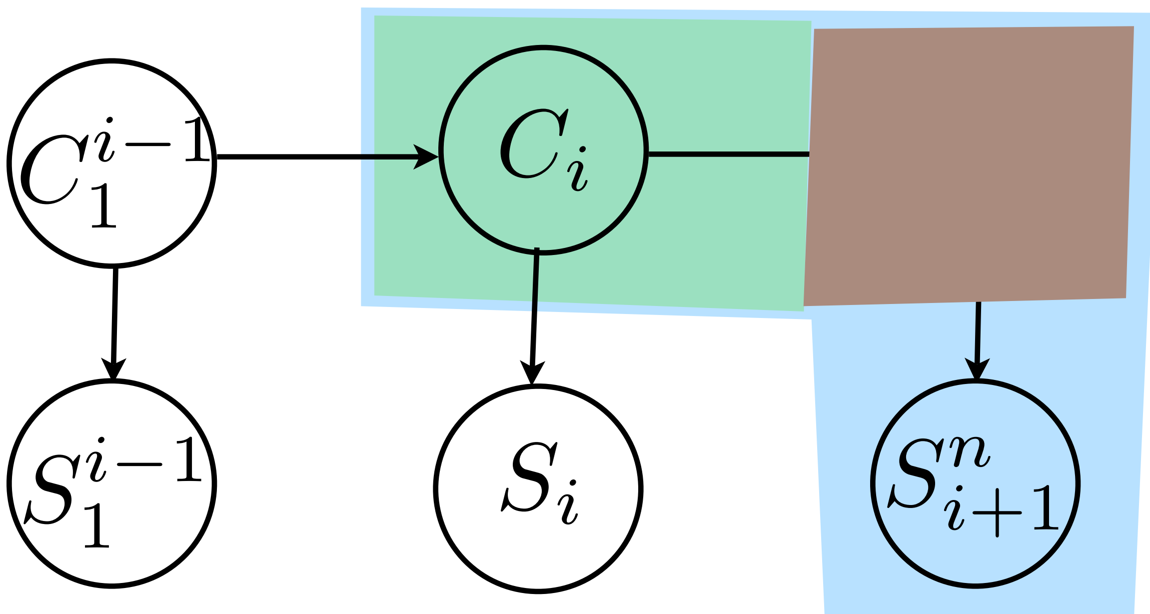
$$p(s_1^n) = p(s_1^n \mid C_0 = start) = back(0, start)$$

Visualizing Backward Probabilities



$$
\begin{aligned}
p(s_{i+1}^n \mid c_i) &= \sum_{c_{i+1}^n} p(s_{i+1}^n, c_{i+1}^n \mid c_i) \\
&= \sum_{c_{i+1}, c_{i+2}^n} p(c_{i+1}^n \mid c_i) \cdot p(s_{i+1}^n \mid c_{i+1}^n) \\
&= \sum_{c_{i+1}} p(c_{i+1} \mid c_i) \cdot p(s_{i+1} \mid c_{i+1}) \cdot p(s_{i+2}^n \mid c_{i+1})
\end{aligned}
$$

Visualizing Backward Probabilities



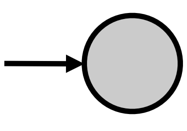
$$
\begin{aligned}
p(s_{i+1}^n \mid c_i) &= \sum_{c_{i+1}^n} p(s_{i+1}^n, c_{i+1}^n \mid c_i) \\
&= \sum_{c_{i+1}, c_{i+2}^n} p(c_{i+1}^n \mid c_i) \cdot p(s_{i+1}^n \mid c_{i+1}^n) \\
&= \sum_{c_{i+1}} p(c_{i+1} \mid c_i) \cdot p(s_{i+1} \mid c_{i+1}) \cdot p(s_{i+2}^n \mid c_{i+1})
\end{aligned}
$$

Backward Algorithm (Bigram HMM Equations)

(simplification here)

Backward Algorithm (Bigram HMM Equations)

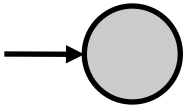
$$back(n+1, \mathrm{stop}) = 1$$



(simplification here)

Backward Algorithm (Bigram HMM Equations)

$$back(n+1, \text{stop}) = 1$$

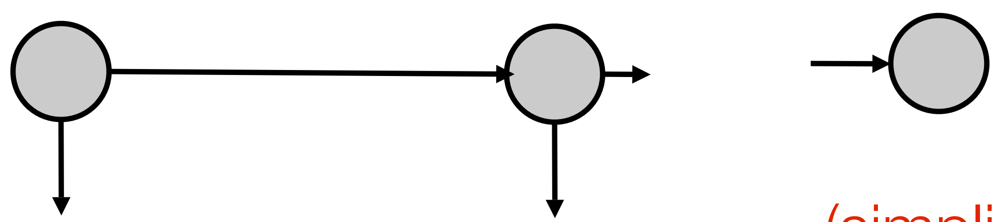


$$\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

(simplification here)

Backward Algorithm (Bigram HMM Equations)

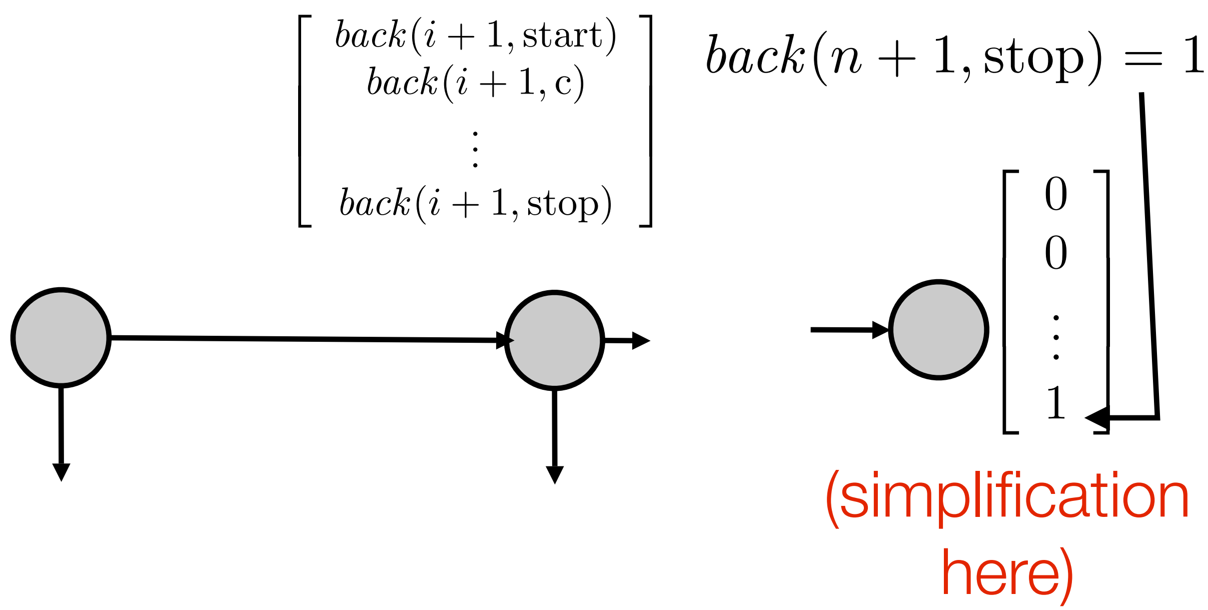
$$back(n+1, \text{stop}) = 1$$



$$\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

(simplification here)

Backward Algorithm
(Bigram HMM Equations)



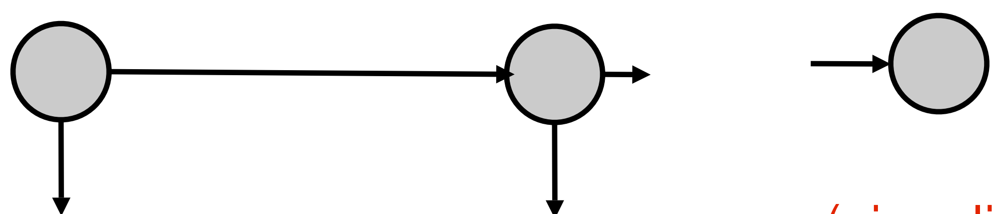
Backward Algorithm (Bigram HMM Equations)

$$back(i, c') = \sum_{c \in \Lambda} \eta(s_{i+1} \mid c) \cdot \gamma(c \mid c') \cdot back(i+1, c)$$

$$\begin{bmatrix} back(i+1, \text{start}) \\ back(i+1, \text{c}) \\ \vdots \\ back(i+1, \text{stop}) \end{bmatrix}$$

$$back(n+1, \text{stop}) = 1$$

$$\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$



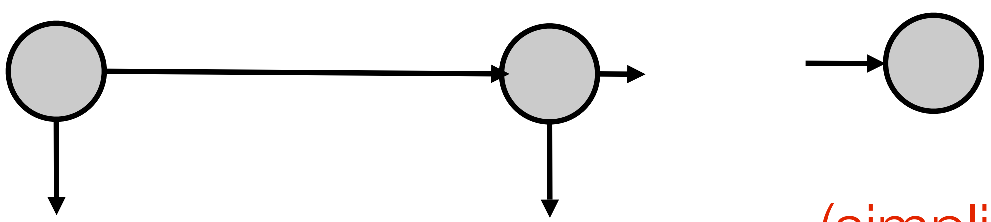
(simplification here)

Backward Algorithm (Bigram HMM Equations)

$$back(i, c') = \sum_{c \in \Lambda} \eta(s_{i+1} \mid c) \cdot \gamma(c \mid c') \cdot back(i+1, c)$$

$$\begin{bmatrix} back(i, \text{start}) \\ back(i, \text{c}) \\ \vdots \\ back(i, \text{stop}) \end{bmatrix} \quad \begin{bmatrix} back(i+1, \text{start}) \\ back(i+1, \text{c}) \\ \vdots \\ back(i+1, \text{stop}) \end{bmatrix} \quad back(n+1, \text{stop}) = 1$$

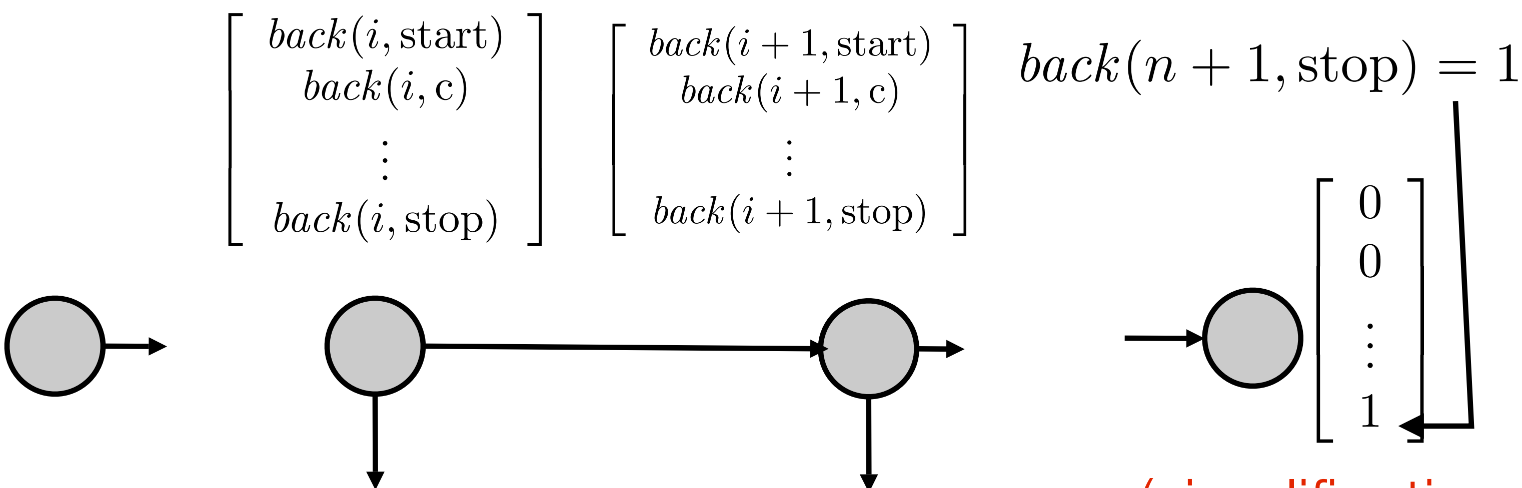
$$\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$



(simplification here)

Backward Algorithm (Bigram HMM Equations)

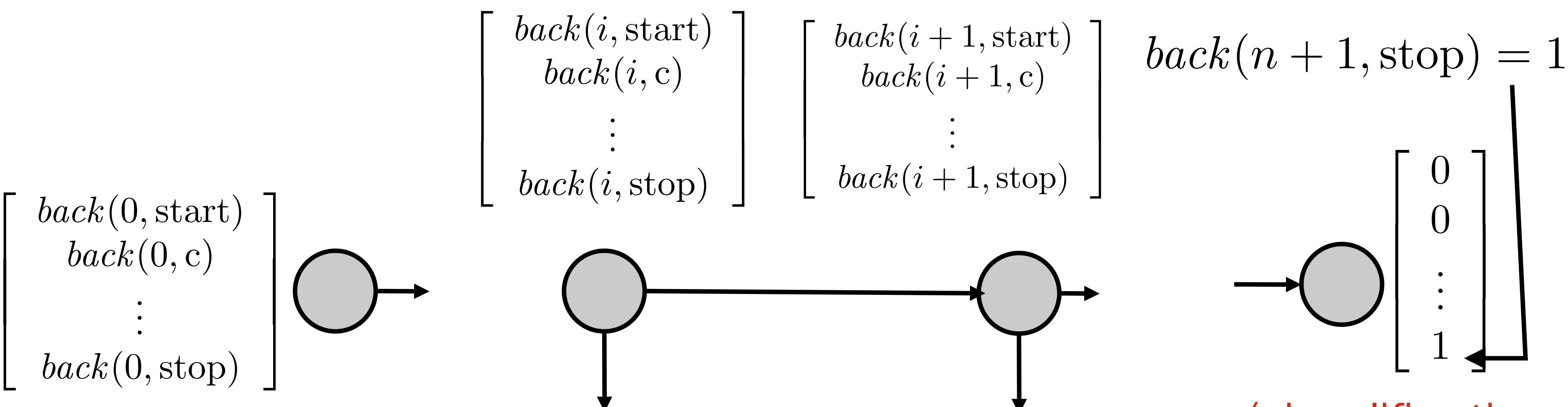
$$back(i, c') = \sum_{c \in \Lambda} \eta(s_{i+1} \mid c) \cdot \gamma(c \mid c') \cdot back(i+1, c)$$



(simplification here)

Backward Algorithm (Bigram HMM Equations)

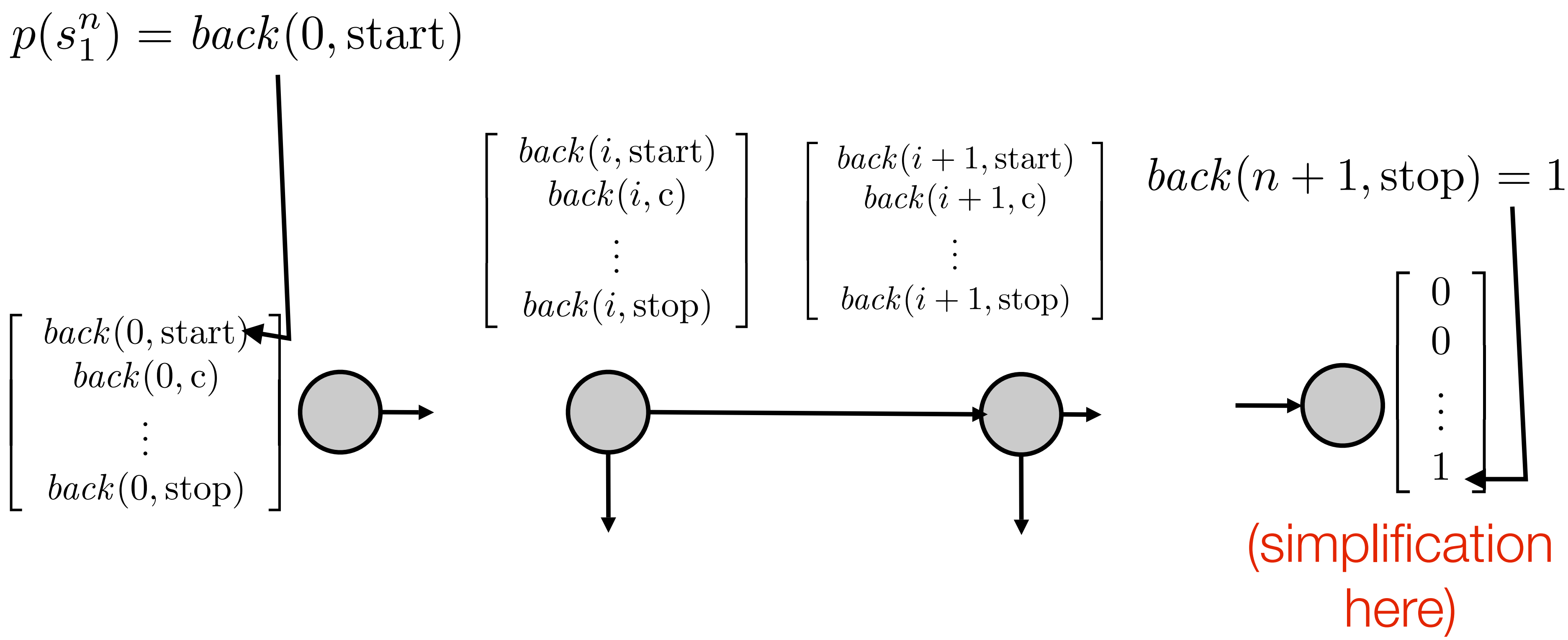
$$back(i, c') = \sum_{c \in \Lambda} \eta(s_{i+1} \mid c) \cdot \gamma(c \mid c') \cdot back(i+1, c)$$



(simplification here)

Backward Algorithm (Bigram HMM Equations)

$$back(i, c') = \sum_{c \in \Lambda} \eta(s_{i+1} \mid c) \cdot \gamma(c \mid c') \cdot back(i+1, c)$$



Semiring Weighted Logic Program

- back(I, D) += $\underline{\eta}$(A | C) × $\underline{\gamma}$(C | D) × $\underline{s}$(A, I+1) × back(I+1, C).
- back(N, S) += $\underline{\gamma}$(stop | S) × $\underline{\text{length}}$(N).
- goal += back(0, start)

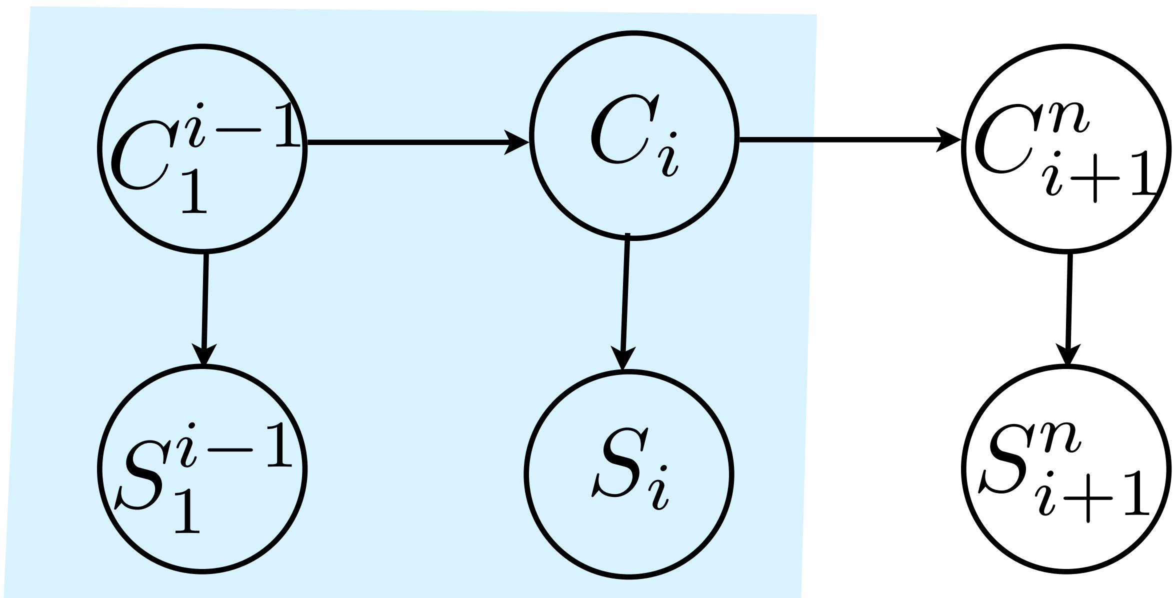
- Semiring?
- Graph/hypergraph?
- Runtime?
- Execution strategy?

Forward Probabilities

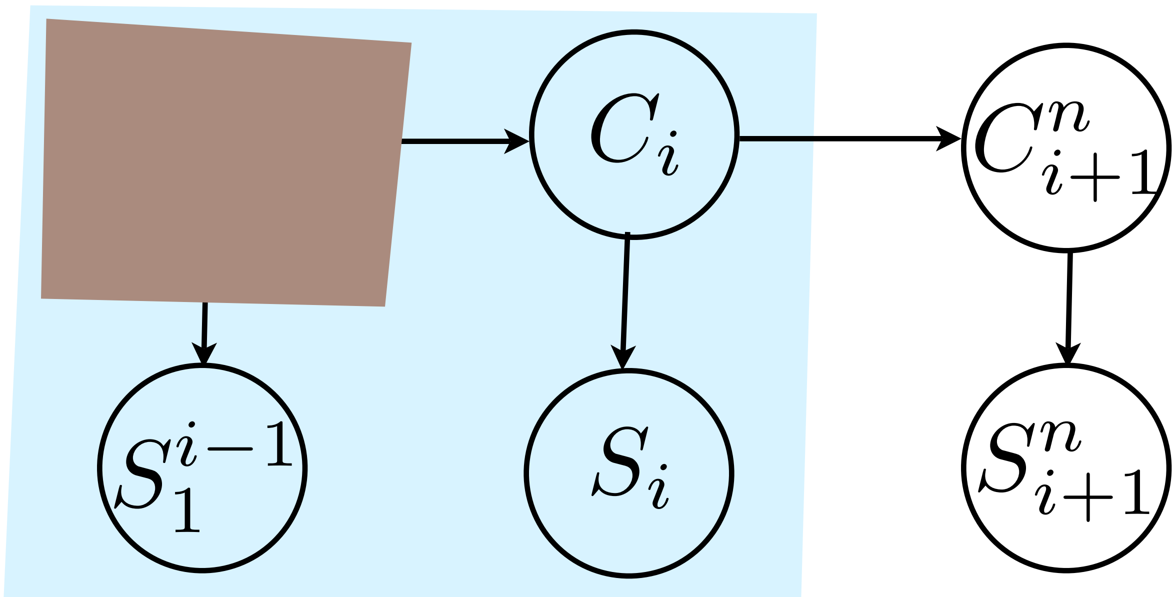
$$forw(i, c) = p(s_1^i, C_i = c)$$

$$p(s_1^n) = p(s_1^n, C_{n+1} = stop) = forw(n + 1, stop)$$

Visualizing Forward Probabilities

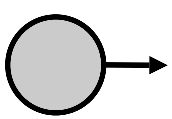


Visualizing Forward Probabilities



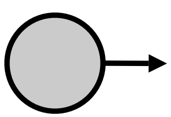
Forward Algorithm (Bigram HMM Equations)

Forward Algorithm
(Bigram HMM Equations)



Forward Algorithm (Bigram HMM Equations)

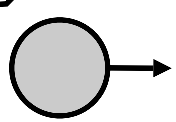
$$\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$



Forward Algorithm (Bigram HMM Equations)

$$forw(0, \text{start}) = 1$$

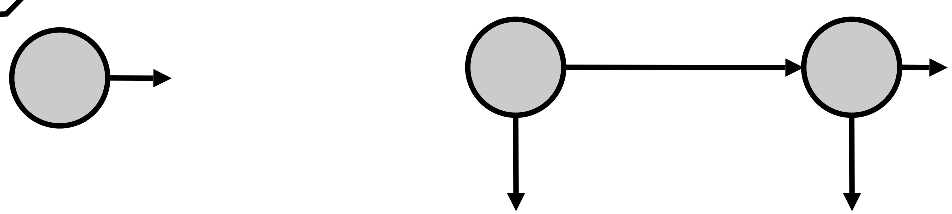
$$\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$



Forward Algorithm
(Bigram HMM Equations)

$$forw(0, \text{start}) = 1$$

$$\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

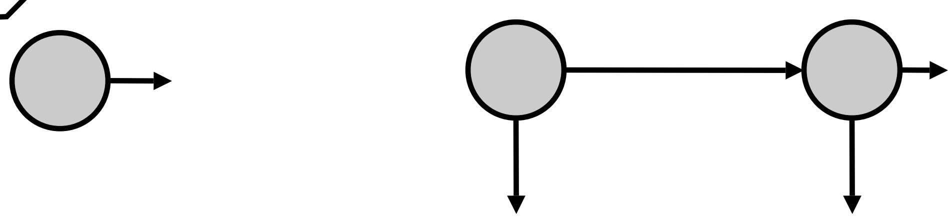


Forward Algorithm (Bigram HMM Equations)

$$forw(0, \text{start}) = 1$$

$$\begin{bmatrix} forw(i, \text{start}) \\ forw(i, \text{c}) \\ \vdots \\ forw(i, \text{stop}) \end{bmatrix}$$

$$\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$



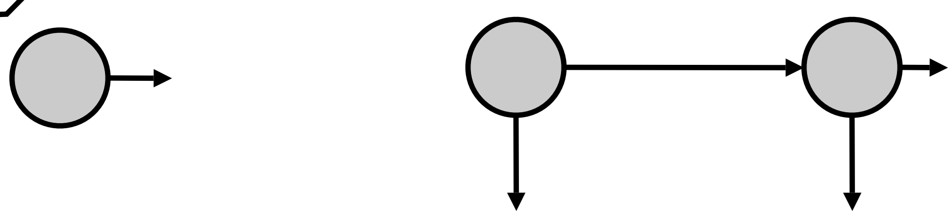
Forward Algorithm (Bigram HMM Equations)

$$forw(i, c') = \sum_{c \in \Lambda} \eta(s_i \mid c') \cdot \gamma(c' \mid c) \cdot forw(i-1, c)$$

$$forw(0, \text{start}) = 1$$

$$\begin{bmatrix} forw(i, \text{start}) \\ forw(i, \text{c}) \\ \vdots \\ forw(i, \text{stop}) \end{bmatrix}$$

$$\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

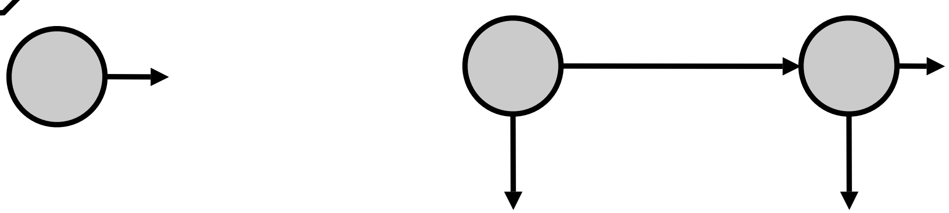


Forward Algorithm (Bigram HMM Equations)

$$forw(i, c') = \sum_{c \in \Lambda} \eta(s_i \mid c') \cdot \gamma(c' \mid c) \cdot forw(i-1, c)$$

$$forw(0, \text{start}) = 1$$

$$\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \begin{bmatrix} forw(i, \text{start}) \\ forw(i, \text{c}) \\ \vdots \\ forw(i, \text{stop}) \end{bmatrix} \quad \begin{bmatrix} forw(i, \text{start}) \\ forw(i, \text{c}) \\ \vdots \\ forw(i, \text{stop}) \end{bmatrix}$$



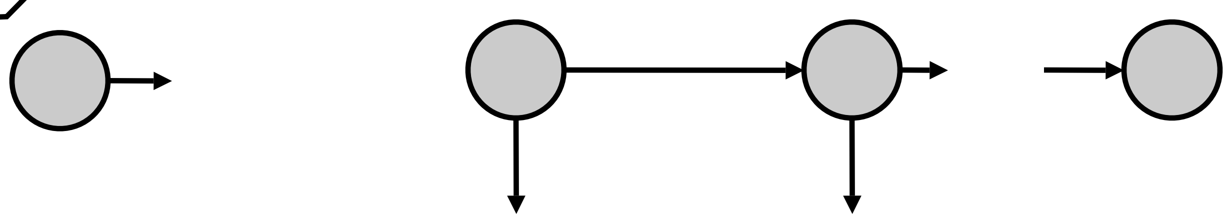
Forward Algorithm (Bigram HMM Equations)

$$forw(i, c') = \sum_{c \in \Lambda} \eta(s_i \mid c') \cdot \gamma(c' \mid c) \cdot forw(i-1, c)$$

$$forw(0, \text{start}) = 1$$

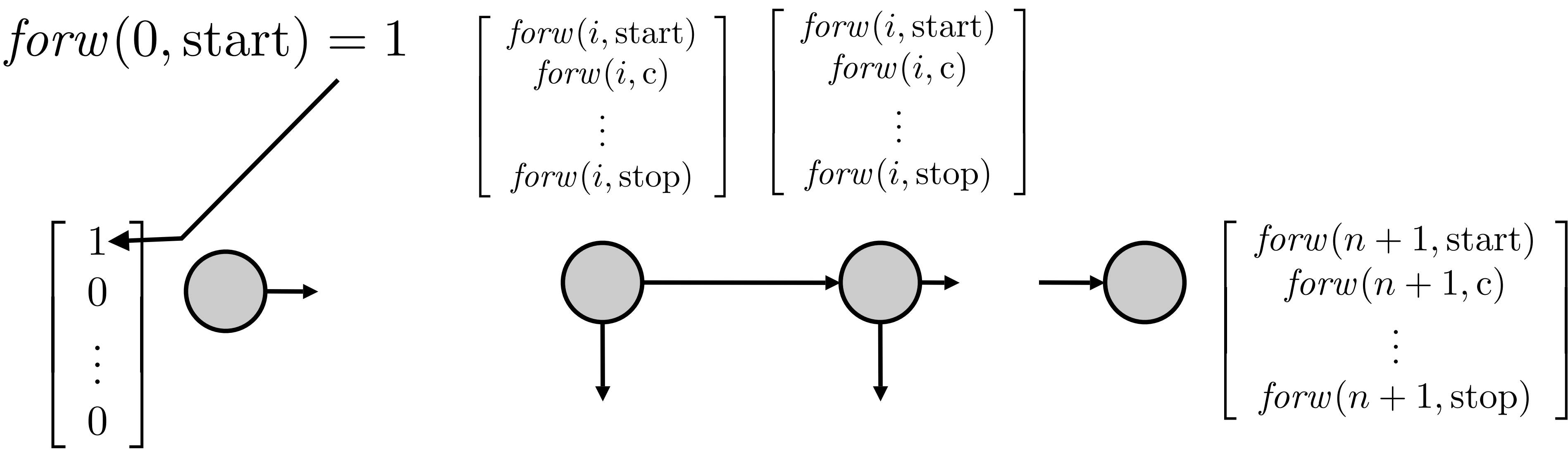
$$\begin{bmatrix} forw(i, \text{start}) \\ forw(i, \text{c}) \\ \vdots \\ forw(i, \text{stop}) \end{bmatrix} \quad \begin{bmatrix} forw(i, \text{start}) \\ forw(i, \text{c}) \\ \vdots \\ forw(i, \text{stop}) \end{bmatrix}$$

$$\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$



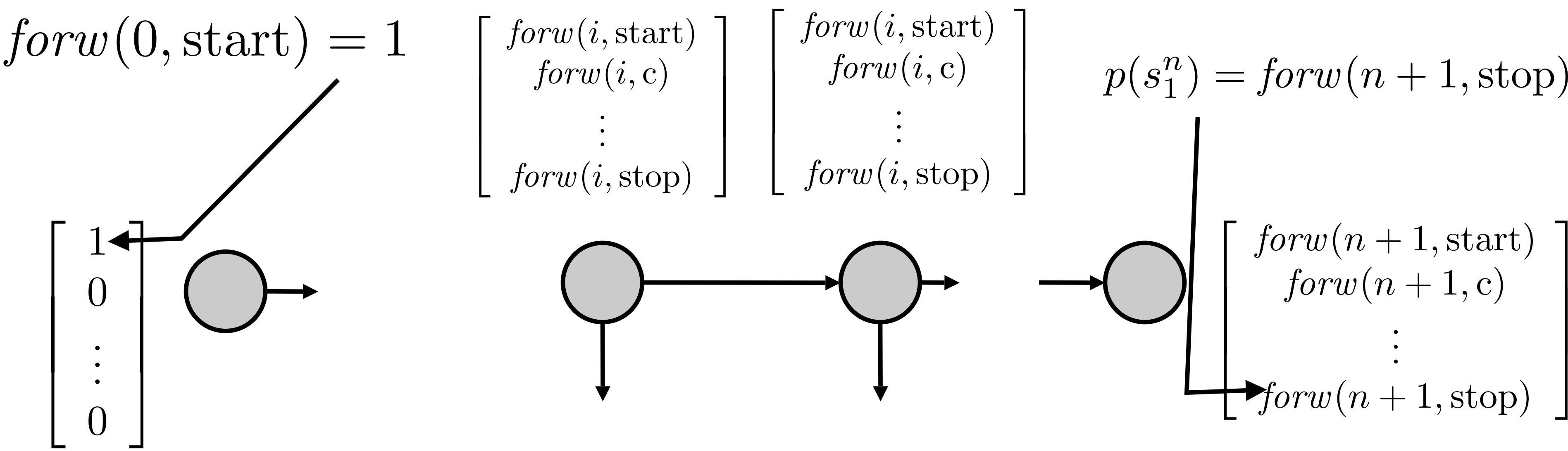
Forward Algorithm (Bigram HMM Equations)

$$forw(i, c') = \sum_{c \in \Lambda} \eta(s_i \mid c') \cdot \gamma(c' \mid c) \cdot forw(i-1, c)$$



Forward Algorithm (Bigram HMM Equations)

$$forw(i, c') = \sum_{c \in \Lambda} \eta(s_i \mid c') \cdot \gamma(c' \mid c) \cdot forw(i-1, c)$$



Semiring Weighted Logic Program

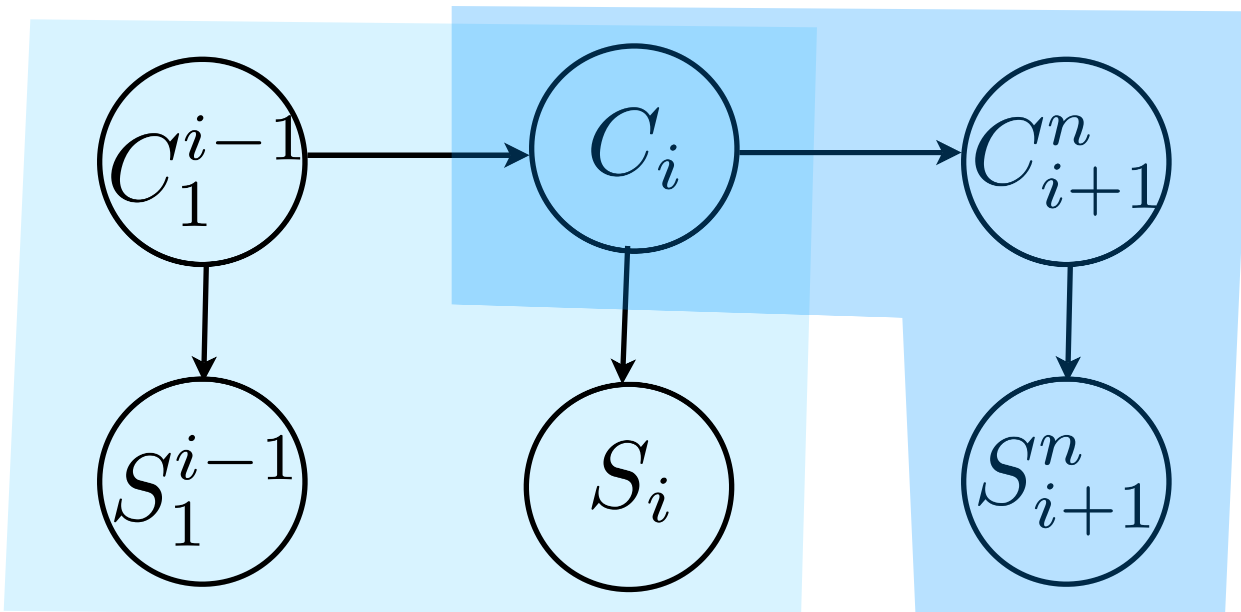
- $\text{forw}(I, D) \mathrel{+}= \underline{\eta}(A \mid D) \times \underline{\gamma}(D \mid C) \times \underline{s}(A, I) \times \text{forw}(I-1, C).$
- $\text{forw}(0, \text{start}) := 1.$
- $\text{goal} \mathrel{+}= \underline{\gamma}(\text{stop} \mid C) \times \underline{\text{length}}(N) \times \text{forw}(N, C).$

- Semiring?
- Graph/hypergraph?
- Runtime?
- Execution strategy?

Putting Forward and Backward Together

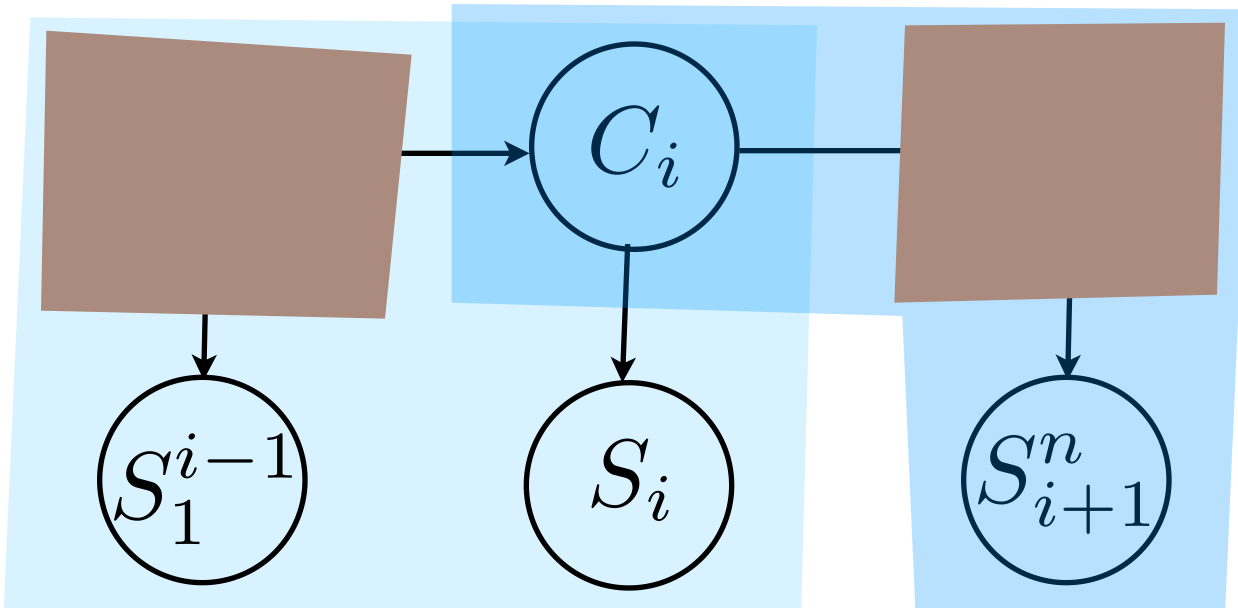
$$
\begin{aligned}
back(i,c) &= p(s_{i+1}^n \mid C_i = c) \\
forw(i,c) &= p(s_1^i, C_i = c) \\
back(0, start) &= p(s_1^n) \\
forw(n+1, \text{stop}) &= p(s_1^n) \\
back(i,c) \times forw(i,c) &= p(s_1^n, C_i = c) \\
\frac{back(i,c) \times forw(i,c)}{p(s_1^n)} &= p(C_i = c \mid s_1^n) = \mathbb{E}[\delta(C_i, c) \mid s_1^n] \\
\sum_{i=1}^{n} \frac{back(i,c) \times forw(i,c)}{p(s_1^n)} &= \sum_{i=1}^{n} p(C_i = c \mid s_1^n) = \mathbb{E}[count(c) \mid s_1^n]
\end{aligned}
$$

Visualizing Forward-Backward Products



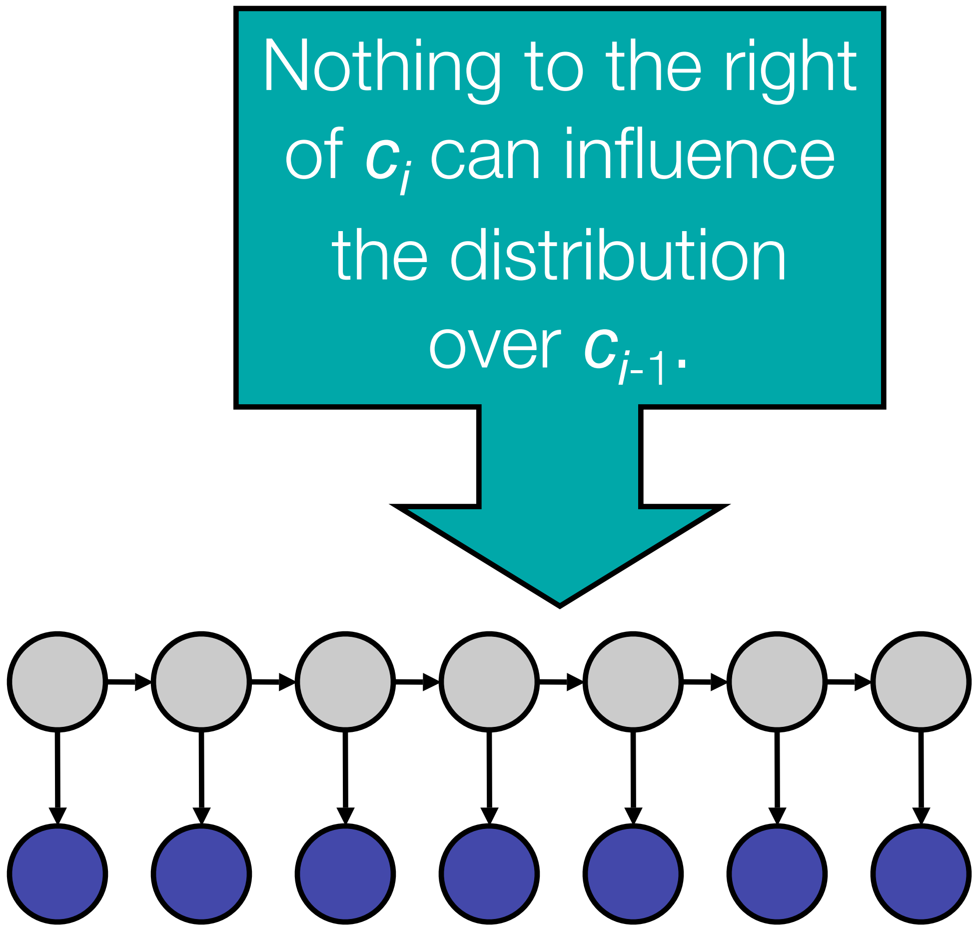
$$\begin{aligned} back(i,c) \cdot forw(i,c) &= p(s_{i+1}^n \mid C_i = c) \cdot p(s_1^i, C_i = c) \\ &= p(s_1^n, C_i = c) \end{aligned}$$

Visualizing Forward-Backward Products

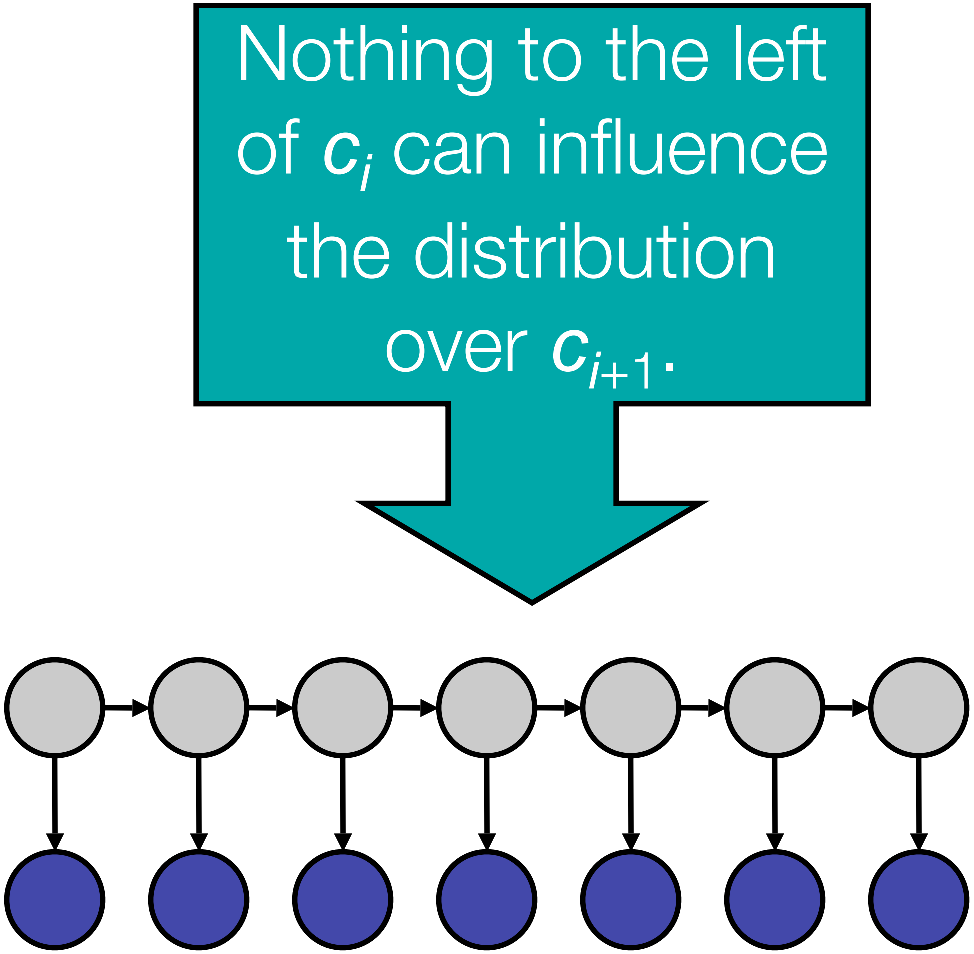


$$
\begin{aligned}
back(i,c) \cdot forw(i,c) &= p(s_{i+1}^n \mid C_i = c) \cdot p(s_1^i, C_i = c) \\
&= p(s_1^n, C_i = c)
\end{aligned}
$$

Why the dynamic programming works



Why the dynamic programming works



Adapting the Programs

- Trigram version?
- Condition η on current *and* previous state?
- Probability distribution over start state?
- Silent states?
- Factored states?

Viterbi Algorithm (Most Likely Label Sequence)

- vit(I, D) max= $\underline{\eta}$(A | D) × $\underline{\gamma}$(D | C) × $\underline{s}$(A, I) × vit(I-1, C).
- vit(0, start) := 1.
- goal max= $\underline{\gamma}$(stop | C) × $\underline{\text{length}}$(N) × vit(N, C).

- Semiring?
- Recovering the path?
- Do you have to go left to right?

Minimum Expected Label-Loss Sequence

$$\hat{c}_i = \arg\max_c p(C_i = c \mid s_1^n) = \arg\min_c \mathbb{E}[\delta(C_i, c)]$$

Minimum Expected Label-Loss Sequence

$$\hat{c}_i = \arg\max_c p(C_i = c \mid s_1^n) = \arg\min_c \mathbb{E}[\delta(C_i, c)]$$

$$p(C_i = c \mid s_1^n) = back(i, c) \cdot forw(i, c)$$

Minimum Expected Label-Loss Sequence

$$\hat{c}_i = \arg\max_c p(C_i = c \mid s_1^n) = \arg\min_c \mathbb{E}[\delta(C_i, c)]$$

$$p(C_i = c \mid s_1^n) = back(i, c) \cdot forw(i, c)$$

$$\hat{c}_1^n = \arg\min_{c_1^n} \sum_{i=1}^{n} \mathbb{E}[\delta(C_i, c_i)] = \arg\min_{c_1^n} \mathbb{E}\left[\sum_{i=1}^{n} \delta(C_i, c_i)\right]$$

Minimum Expected Label-Loss Sequence

$$\hat{c}_i = \arg\max_c p(C_i = c \mid s_1^n) = \arg\min_c \mathbb{E}[\delta(C_i, c)]$$

$$p(C_i = c \mid s_1^n) = back(i, c) \cdot forw(i, c)$$

$$\hat{c}_1^n = \arg\min_{c_1^n} \sum_{i=1}^{n} \mathbb{E}[\delta(C_i, c_i)] = \arg\min_{c_1^n} \mathbb{E}\left[\sum_{i=1}^{n} \delta(C_i, c_i)\right]$$

Same as Viterbi?

Minimum Expected Label-Loss Sequence

$$\hat{c}_i = \arg\max_c p(C_i = c \mid s_1^n) = \arg\min_c \mathbb{E}[\delta(C_i, c)]$$

$$p(C_i = c \mid s_1^n) = back(i, c) \cdot forw(i, c)$$

$$\hat{c}_1^n = \arg\min_{c_1^n} \sum_{i=1}^{n} \mathbb{E}[\delta(C_i, c_i)] = \arg\min_{c_1^n} \mathbb{E}\left[\sum_{i=1}^{n} \delta(C_i, c_i)\right]$$

Same as Viterbi?

Always a valid HMM path?

Random Label Sequence

$$C_1^n, S_1^n$$

Random Label Sequence

- The HMM defines a distribution over C_1^n, S_1^n

Random Label Sequence

- The HMM defines a distribution over C_1^n, S_1^n
- Choose a path according to that distribution

Random Label Sequence

- The HMM defines a distribution over C_1^n, S_1^n
- Choose a path according to that distribution
- Solution: Design an HMM that only emits the sequence we have observed.

Random Label Sequence

- The HMM defines a distribution over C_1^n, S_1^n
- Choose a path according to that distribution
- Solution: Design an HMM that only emits the sequence we have observed.
- Tip: use quantities you already know how to compute.

Random Label Sequence

- The HMM defines a distribution over C_1^n, S_1^n
- Choose a path according to that distribution
- Solution: Design an HMM that only emits the sequence we have observed.
- Tip: use quantities you already know how to compute.
- Question: is the resulting distribution over label sequences representable as a Markov model?