Language and Statistics II

Lecture 3: Sequences (cont'd.)

Noah Smith

Quick Review

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

HMMs

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

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

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

Marginal probability of words?

Naïve algorithm: $O(2^n)$

Inference with HMMs

- Many inference problems can be solved 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

Deriving the Backward Algorithm

$$p\left(s_1^n\right) = p\left(s_1^n \mid C_0 = \text{start}\right) \leftarrow \alpha(0, \text{start})$$

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

$$= \sum_{c_1 \in \Lambda} \sum_{c_2^n \in \Lambda^{n-1}} \left(\prod_{i=1}^{n} \eta\left(s_i \mid c_i\right) \cdot \gamma\left(c_i \mid c_{i-1}\right) \right) \gamma\left(\text{stop} \mid c_n\right)$$

$$= \sum_{c_1 \in \Lambda} \eta\left(s_1 \mid c_1\right) \cdot \gamma\left(c_1 \mid c_0\right) \sum_{c_2^n \in \Lambda^{n-1}} \left(\prod_{i=2}^{n} \eta\left(s_i \mid c_i\right) \cdot \gamma\left(c_i \mid c_{i-1}\right) \right) \gamma\left(\text{stop} \mid c_n\right)$$

$$= \sum_{c_1 \in \Lambda} \eta\left(s_1 \mid c_1\right) \cdot \gamma\left(c_1 \mid \text{start}\right) \cdot p\left(s_2^n \mid c_1\right)$$

$$= \sum_{c \in \Lambda} \eta\left(s_1 \mid c\right) \cdot \gamma\left(c \mid \text{start}\right) \cdot p\left(s_2^n \mid C_1 = c\right)$$

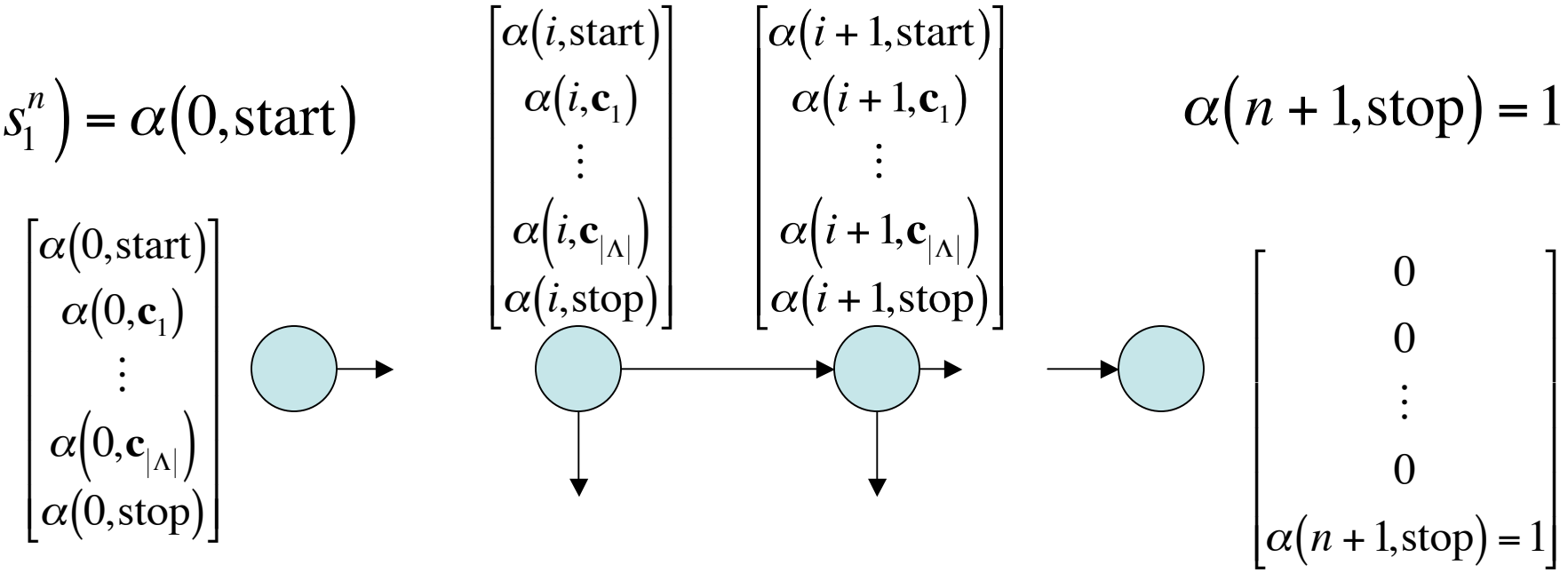
$$\alpha(1, c) \rightarrow$$

Backward Algorithm (Bigram HMM Equations)

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

$$p(s_1^n) = \alpha(0,\text{start})$$

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

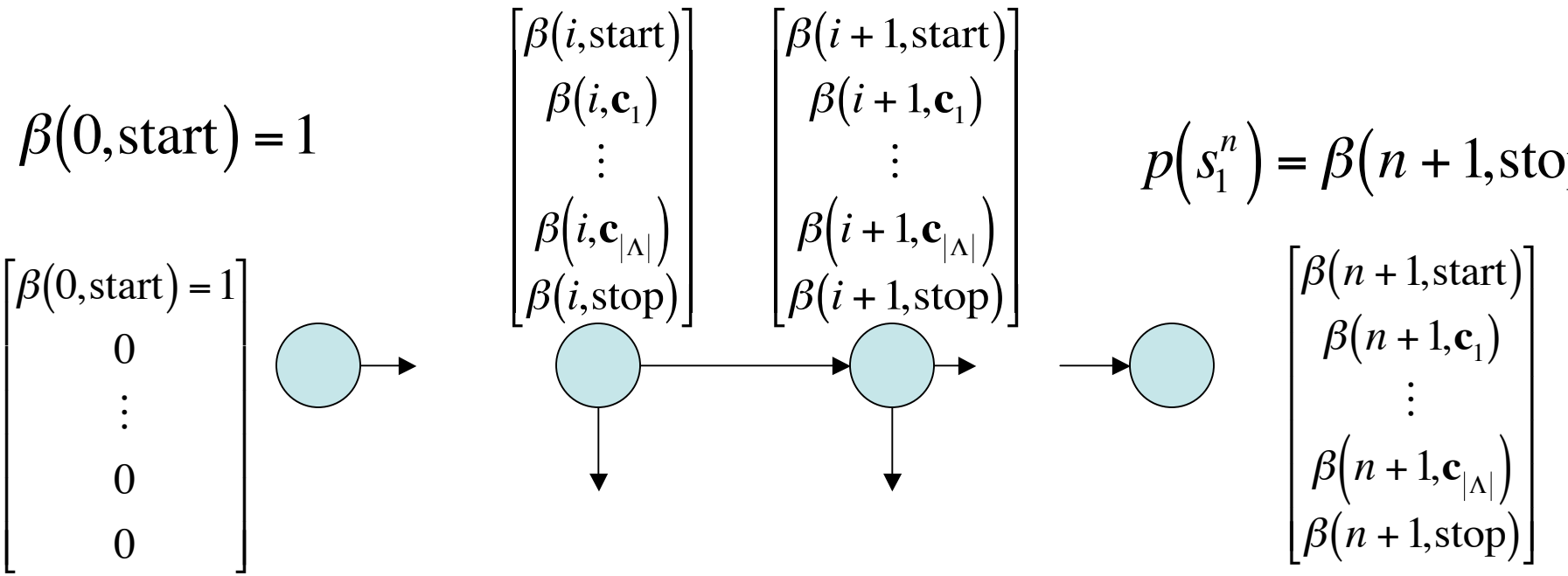


Forward Algorithm (Bigram HMM Equations)

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

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

$$p\left(s_1^n\right) = \beta(n+1,\text{stop})$$



Forward and Backward Probabilities

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

"backward" probability

$$\beta(i,c) = p\left(s_{1}^{i}, C_i = c\right)$$

"forward" probability

$$\alpha(i,c) \cdot \beta(i,c) = p\left(s_{1}^{n}, C_i = c\right)$$

$$\frac{\alpha(i,c) \cdot \beta(i,c)}{\beta(n+1,\text{stop})} = p\left(C_i = c \mid s_{1}^{n}\right)$$

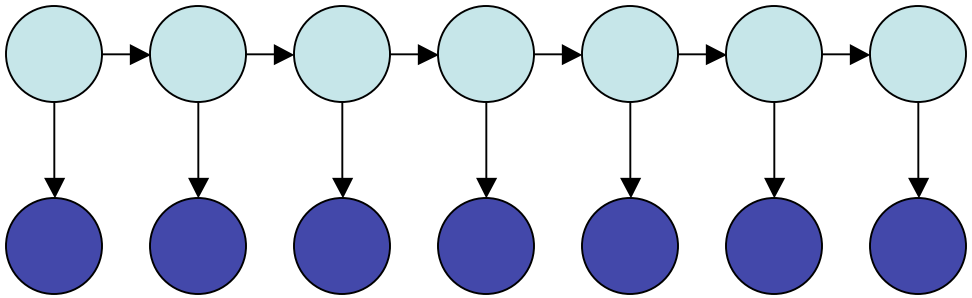
posterior probability that s_i is labeled with class c

$$\sum_{i=1}^{n} \frac{\alpha(i,c) \cdot \beta(i,c)}{\beta(n+1,\text{stop})} = E\left[\left|\left\{i : C_i = c\right\}\right| \mid s_{1}^{n}\right]$$

expected count of class c

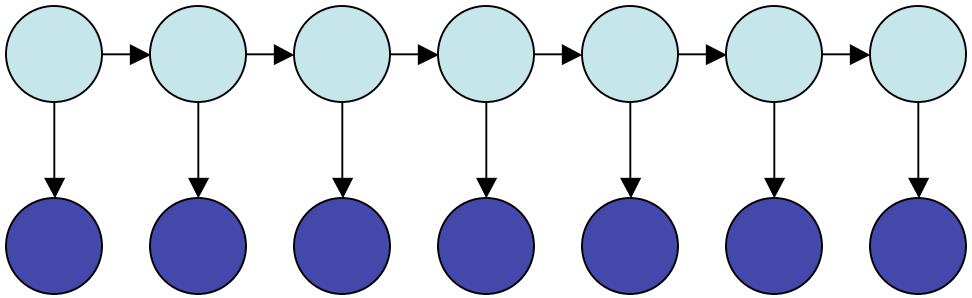
Why it Works

Nothing to the right of c_i can influence the distribution over c_{i-1}.



Why it Works

Nothing to the left of c_i can influence the distribution over c_{i+1}.

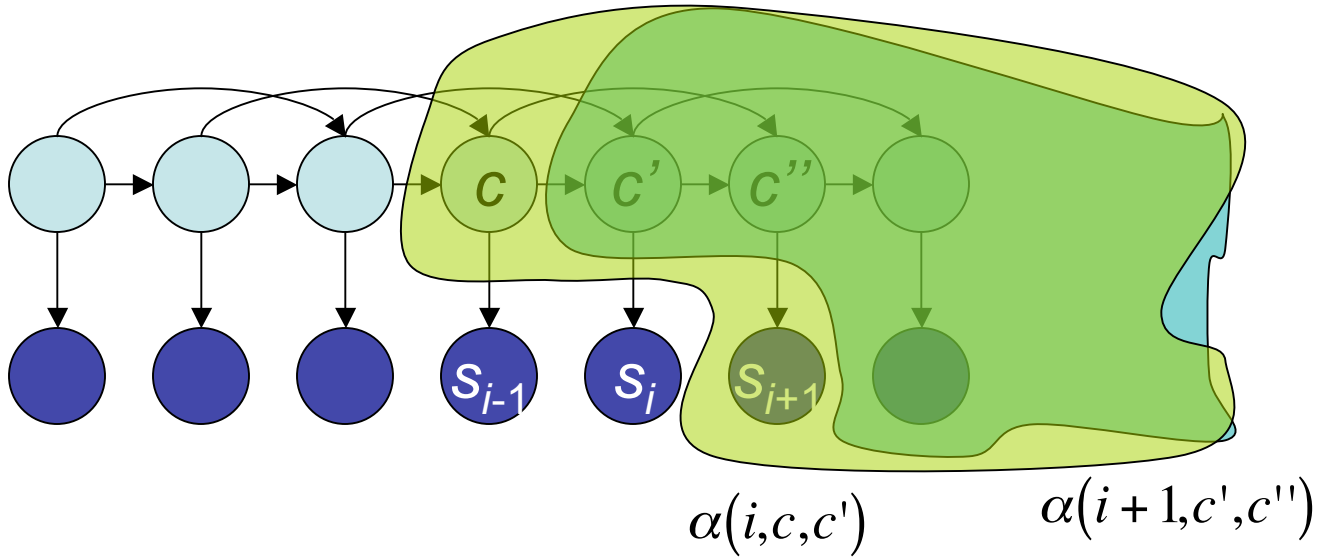


What about a **tri**gram HMM?

$$\forall c, \quad \alpha(n+1, c, \text{stop}) = 1$$

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

$$p(s_1^n) = \alpha(0, \text{start}, \text{start})$$



HMM Problem 2: Most Probable Path

$$\beta^*(0,\text{start}) = 1$$

$$\beta^*(i,c') = \max_{c\in\Lambda} \eta(s_i \mid c') \cdot \gamma(c' \mid c) \cdot \beta^*(i-1,c)$$

$$\max_{c_1^n \in \Lambda^n} p(s_1^n, c_1^n) = \beta^*(n+1,\text{stop})$$

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

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

$$p(s_1^n) = \beta(n+1,\text{stop})$$

How to recover the path itself?

Is it necessary to go left to right?

HMM Problem 3: Minimum Expected Label Loss Path

$$\hat{c}_i = \arg\max_{c \in \Lambda} p\left(C_i = c \mid s_1^n\right) = \arg\max_{c \in \Lambda} \alpha(i,c)\beta(i,c)$$

$$\hat{c}_1^n = \arg\min_{c_1^n \in \Lambda^n} E\left[\left|\left\{i : C_i \neq c_i\right\}\right|\right]$$

Same as Viterbi?

Always a valid path?

HMM Problem 4: Most Probable Path Through a Lattice

- Lattice unweighted?
  - No problem! Slight generalization of Viterbi: index states, not word positions.
- Lattice weighted?
  - NP-hard!
  - Casacuberta & de la Higuera (2000); Lyngsoe & Pedersen (2002)

Dynamic Programming

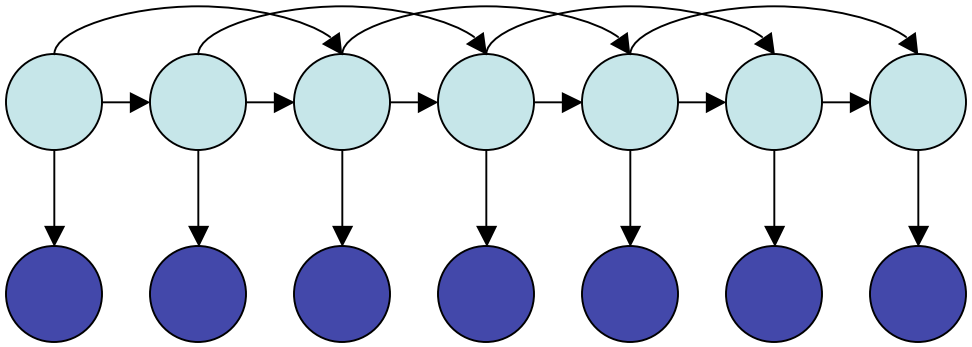
	n-gram	HMM
Best sequence in an unweighted lattice	✔	✔
Best sequence in a weighted lattice (product score)	✔	✖
Total probability of unweighted lattice	✔	✔
Total probability of weighted lattice (product score)	✔	✔

Extensions to HMMs

- Higher *n* (how are algorithms affected?)
- Factored states, multiple levels
  - Lots of neat work by Bilmes (UW).
  - Also a toolkit.
- Mixture with Markov model
- Alternative estimation criteria (coming soon)

Hidden Markov Model

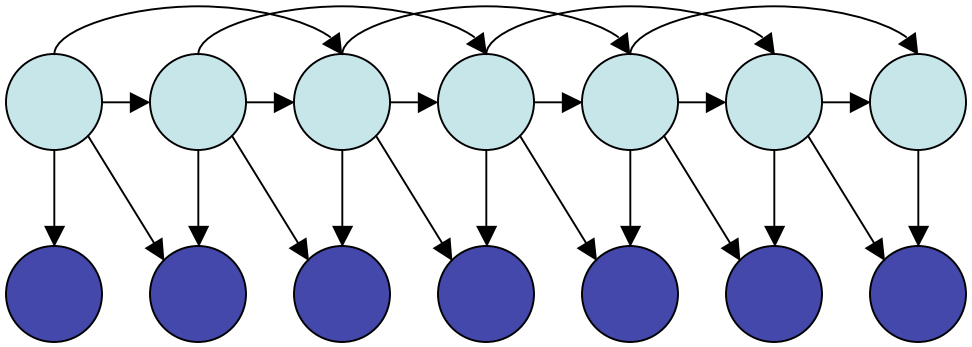
$\gamma(c_i \mid c_{i-2}, c_{i-1})$



$\eta(s_i \mid c_i)$

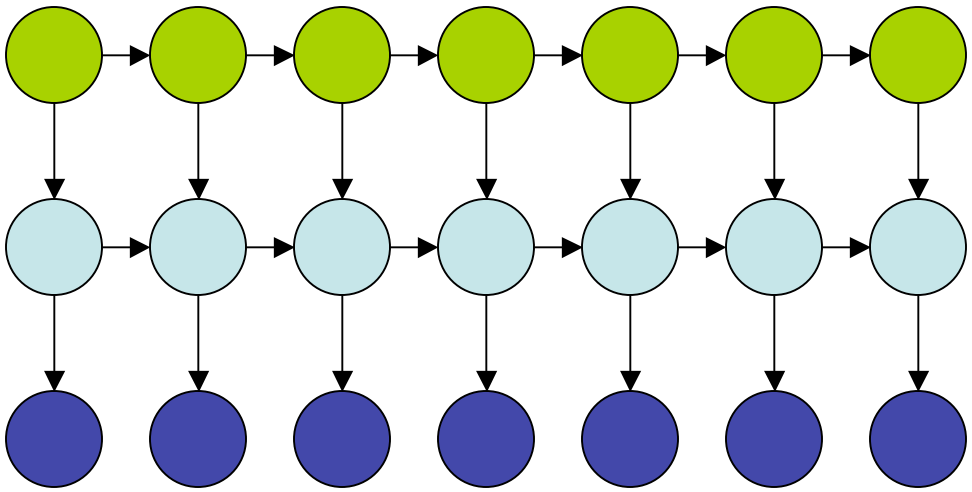
Hidden Markov Model (Variant with more conditioning)

$\gamma(c_i \mid c_{i-2}, c_{i-1})$



$\eta(s_i \mid c_{i-1}, c_i)$

Hidden Markov Model (Factored-state variant)



$\phi(q_i \mid q_{i-1})$

$\gamma(c_i \mid c_{i-1}, q_i)$

$\eta(s_i \mid c_i)$

Summary So Far

- Tradeoffs in modeling
- Model ≠ application ≠ inference algorithm
- Review of HMMs (model, well-known applications, common algorithms)
- Lots of dynamic programming tricks

Next:

- Sequence labeling alternatives (features, estimation) … log-linear models
- Weighted finite-state NLP
- Beyond sequence labeling: parsing