A.I. in health informatics
lecture 8 structured learning

kevin small &
byron wallace

today

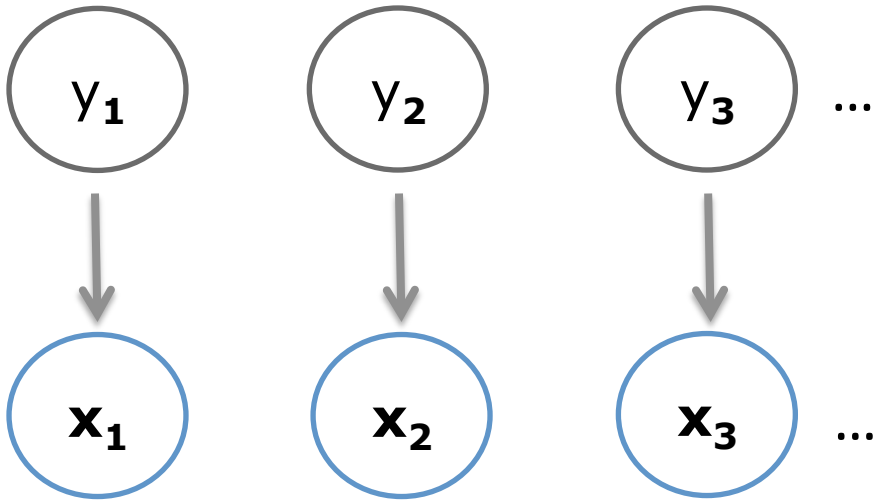
- models for structured learning: HMMs and CRFs
  - structured learning is particularly useful in biomedical applications: parsing (clinical) text; genetic data, etc.
  - we'll cover this in more detail; but need the basics first

unstructured learning

assumptions:

- we're given a set of i.i.d. instances $\{\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_N\}$ and their (univariate) labels $\{y_1, y_2, \ldots y_N\}$
- no order or *sequence* to the data

unstructured learning: graphically



structured learning

assumptions:

- there is some correlation between a label $\mathbf{y}_i$ and the *preceding* labels, in other words,

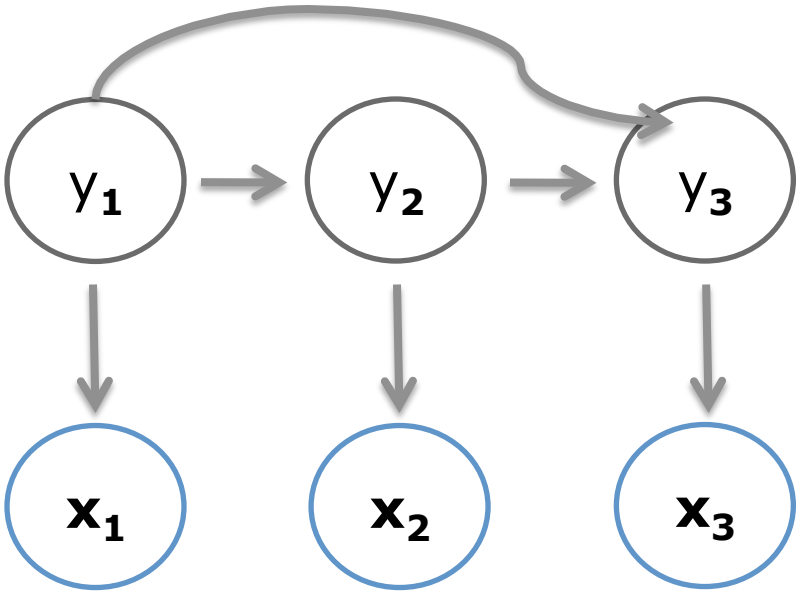
- $\mathbf{y}_i$ is **structured**, ie., y_{i+1} is affected by the previous labels y_i, y_{i-1} …

structured learning

- consider the task of part-of-speech (POS) tagging sentences
  - *nouns* tend to follow *verbs*

image from: http://www.cnts.ua.ac.be/pages/MBSP

structured learning: graphically

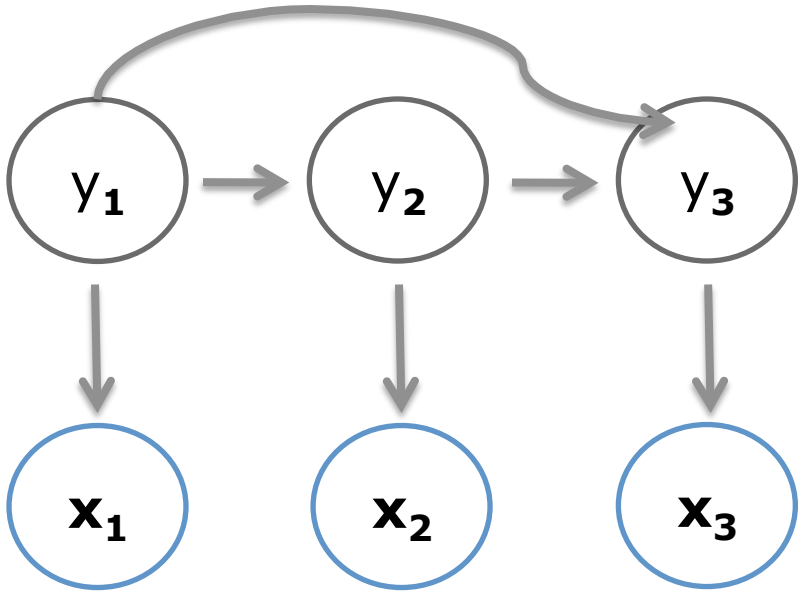


probability & structured learning

$$p(\mathrm{x}_1, \mathrm{x}_2, \ \ldots\ , \mathrm{x}_n) = \prod_{i=1}^{n} p(\mathrm{x}_i \mid \mathrm{x}_1, \ \ldots, \mathrm{x}_{\mathrm{i\text{-}1}})$$

intractable!

(first-order) HMM



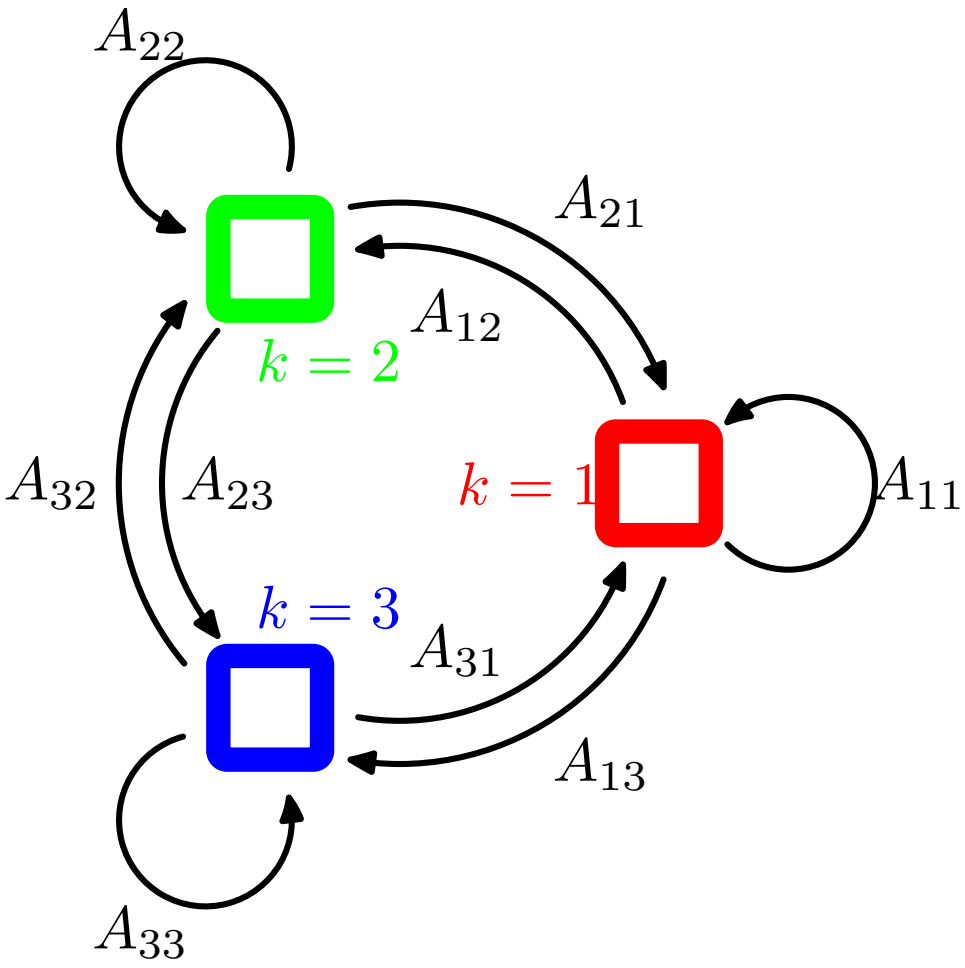
* usually assume states (ys) are *latent*; but not always (see weather example, upcoming)

(first-order) MM

$$p(x_1, x_2, \ldots, x_n) = p(x_1) \prod_{i=2}^{n} p(x_i \mid x_{i-1})$$

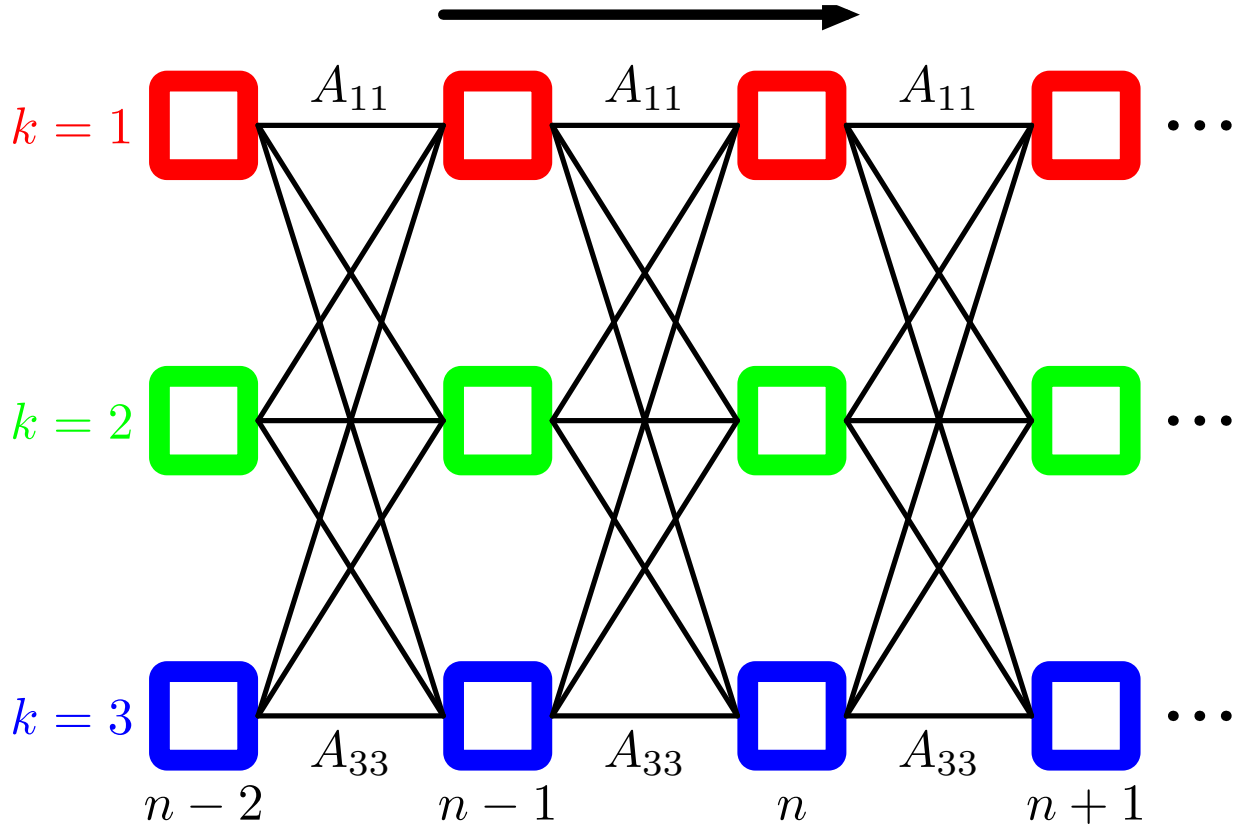
tractable!

markov model



from Bishop PMLR

markov model: unfolded



from Bishop PRML

markov model

- $\mathbf{a_{ij}}$ – probability of transitioning from state *i* to *j*
  - $\sum_j a_{ij} = 1$ (we have to go somewhere!)
  - $a_{ij} \geq 0$

let's talk about the weather

- the world has three states:

  1 *rainy*, *2 cloudy,* 3 *sunny*

- (in the weather case the markov model is not hidden)

- our transition matrix is:

$$A = \{a_{ij}\} = \begin{bmatrix} 0.4 & 0.3 & 0.3 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}$$

note that this system likes to stay where it is!

the weather

- probability that of observing the weather
  {*sunny*, *sunny*, *rainy*}?

$$A = \{a_{ij}\} = \begin{array}{c} \begin{matrix} R & C & S \end{matrix} \\ \begin{bmatrix} 0.4 & 0.3 & 0.3 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{bmatrix} \end{array}$$

= 1.0 * P(S|S)*P(R|S) = .8*.1

the weather

- it's sunny today. what's the probability that it remains so for *k* days?

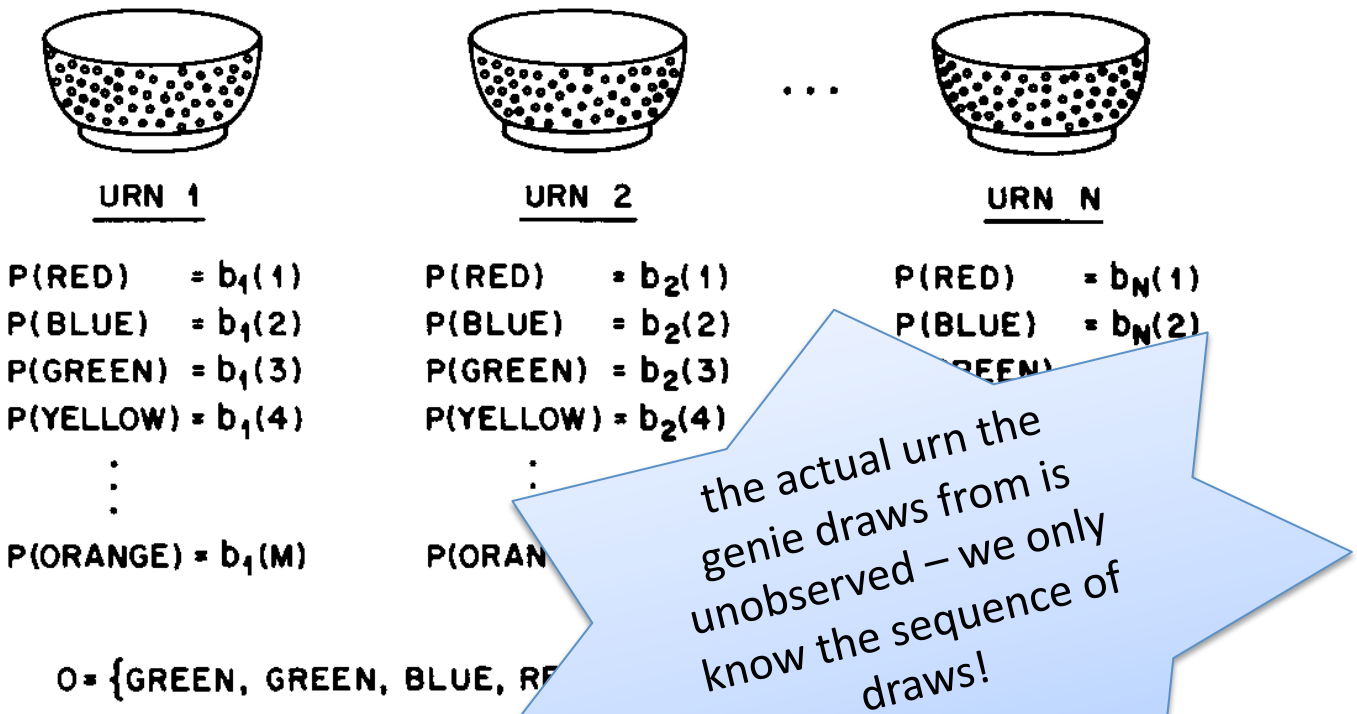
$$A = \{a_{ij}\} = \begin{array}{c} \\ R \\ C \\ S \end{array} \begin{array}{c} \begin{array}{ccc} R & C & S \end{array} \\ \begin{bmatrix} 0.4 & 0.3 & 0.3 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{bmatrix} \end{array}$$

$$= (.8)^k$$

hidden markov models

- in weather example, we only care about *transitions*
  - the (weather) states were the observations
- often we need to model data in which an observation is generated conditional on some latent, underlying state
  - e.g., bias coin example; urn-genie example
- enter the HMM

the urn example



a ***genie (?!?)*** is in the room choosing [illegible] to draw balls from – each earn contains different proportions of the various colored balls

hidden markov models: sufficient parameters

N states (latent), *M* symbols (observed): symbols are observed conditioned on the current state

A - transition probabilities (from urn to urn)

B – symbol emission probabilities (color proportions in each urn)

π – initial state distribution (initial urn likelihood)

$\lambda = (A, B, \pi)$ specifies our model

hidden markov models: the three problems

1 given a set of observations **o** = $\{o_1, o_2 \ldots o_T\}$ and a model λ, compute $P(\mathbf{o}|\lambda)$

2 given **o**, λ, calculate most likely latent states (usually thought of as labels) **q** = $\{q_1, q_2 \ldots q_T\}$

3 given **o**, calculate λ

hidden markov models: problem 1

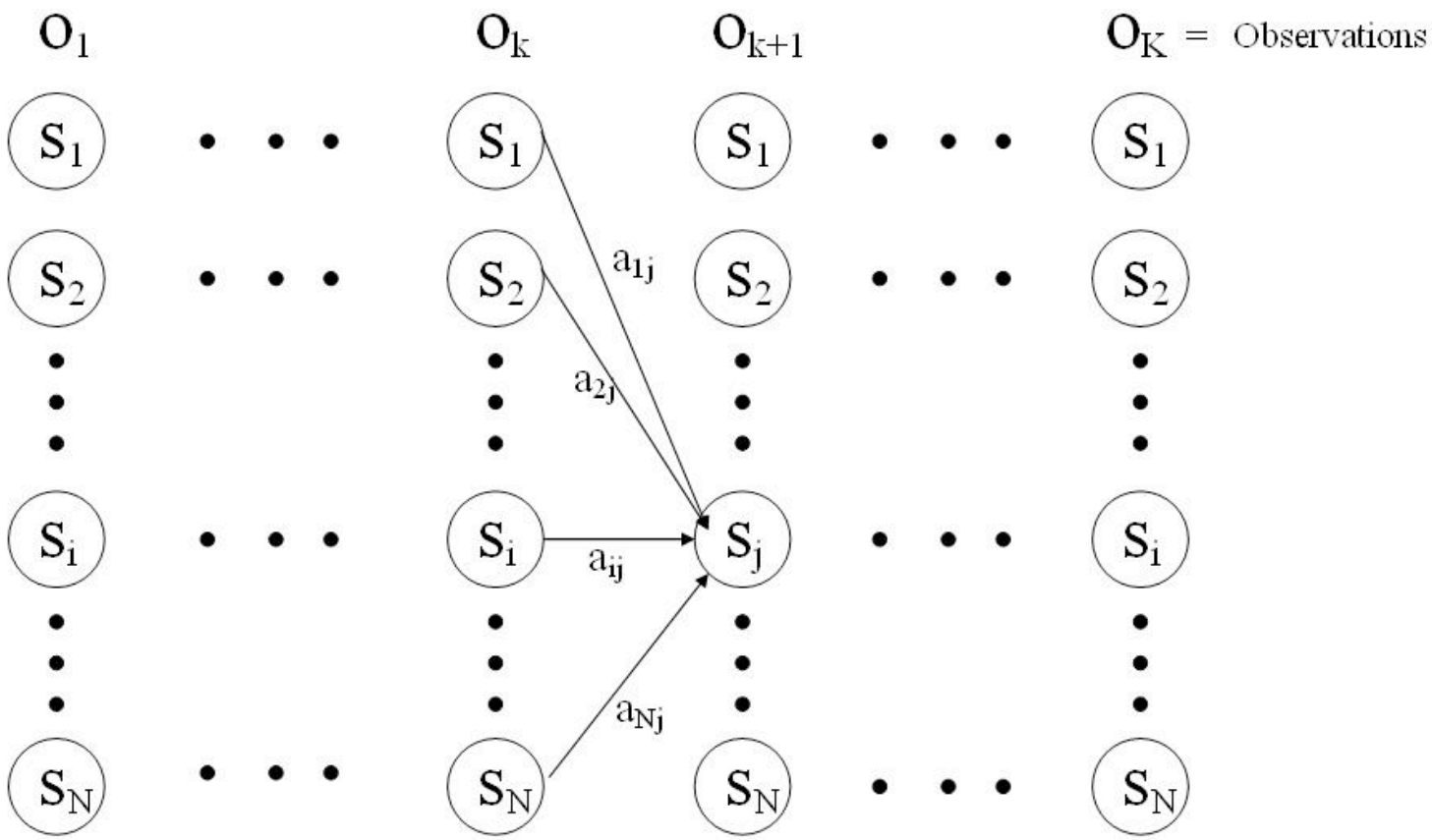
1 given a set of observations **o** = $\{o_1, o_2 \ldots o_T\}$ and a model λ, compute $P(\mathbf{o}|\lambda)$

$$P(\mathbf{o} \mid \lambda) = \sum_{\text{all Q}} P(\mathbf{o} \mid \mathbf{q}, \lambda) P(\mathbf{q} \mid \lambda) =$$

… cool. so we're done?

not quite. this will require $O(N^T)$ calculations

dynamic programming to the rescue!



time T

dynamic programming to the rescue!

1) Initialization:

$$\alpha_1(i) = \pi_i b_i(O_1), \qquad 1 \le i \le N.$$

2) Induction:

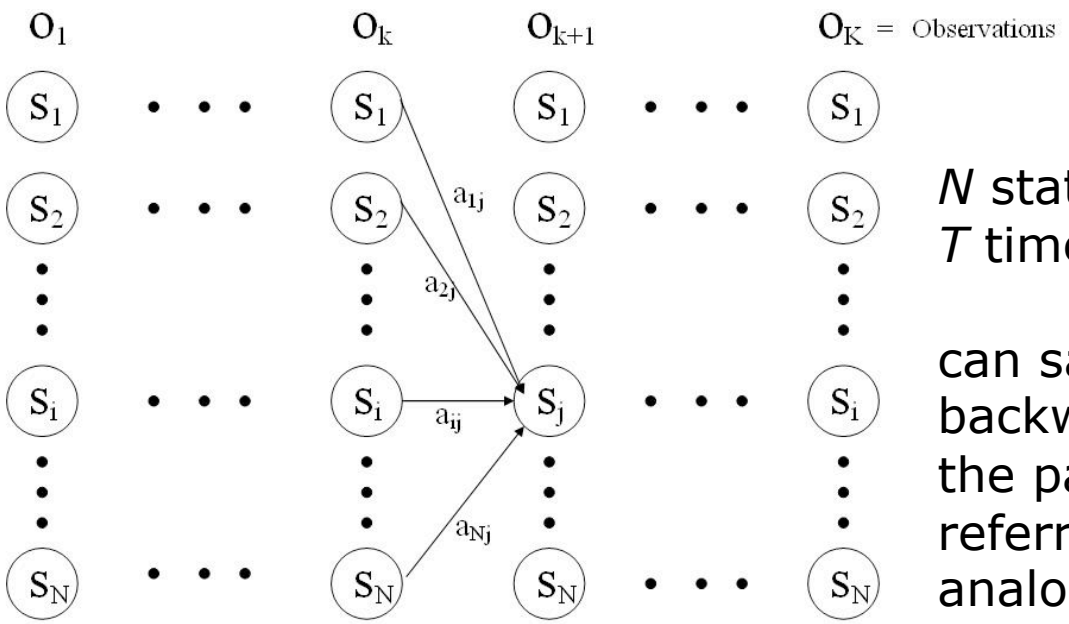
$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(O_{t+1}), \qquad 1 \le t \le T-1$$

$$1 \le j \le N.$$

3) Termination:

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i).$$

about that runtime



N states to (*N*-1) other states *T* times

can save a bit using the backward direction, too (see the paper); this will be referred to with a β and is analogous to α

hidden markov models: the three problems

1 given a set of observations **o** = $\{o_1, o_2 \ldots o_T\}$ and a model λ, compute $P(\mathbf{o}|\lambda)$

2 given **o**, λ, calculate most likely latent states (usually thought of as labels) **q** = $\{q_1, q_2 \ldots q_T\}$

3 given **o**, calculate λ

hidden markov models: problem 2

2 given a set of observations **o** = $\{o_1, o_2 \ldots o_T\}$ and a model λ, find most likely latent states (urns, say) **q*** = $\{q_1, q_2 \ldots q_T\}$

- a unique solution need not exist! what are we even optimizing here?
  - individual most likely states q_i?
  - joint probability **q**?

hidden markov models: problem 2

problem optimizing for the most likely state at *any given time* can lead to *impossible* sequences under λ

so let's consider the joint instead

solving problem 2; the joint solution

want to solve for **q***:

$$q^* \leftarrow \underset{\mathbf{q}}{\operatorname{argmax}} P(q, 0 \mid \lambda)$$

- obviously enumerating all possible **q** sequences is infeasible
- dynamic programming to the rescue, again!

the viterbi algorithm

define:

$$\delta_t(i) = \max_{q_1, q_2, \cdots, q_{t-1}} P[q_1\, q_2 \cdots q_t = i,\, O_1\, O_2 \cdots O_t | \lambda]$$

(most likely sequence up to time t-1). the inductive ste

$$\delta_{t+1}(j) = [\max_i \delta_t(i)\, a_{ij}] \cdot b_j(O$$

i.e., from the best path so far, we find the mo
transition/emission probability

that's it! we just need to keep track of the states we visit!

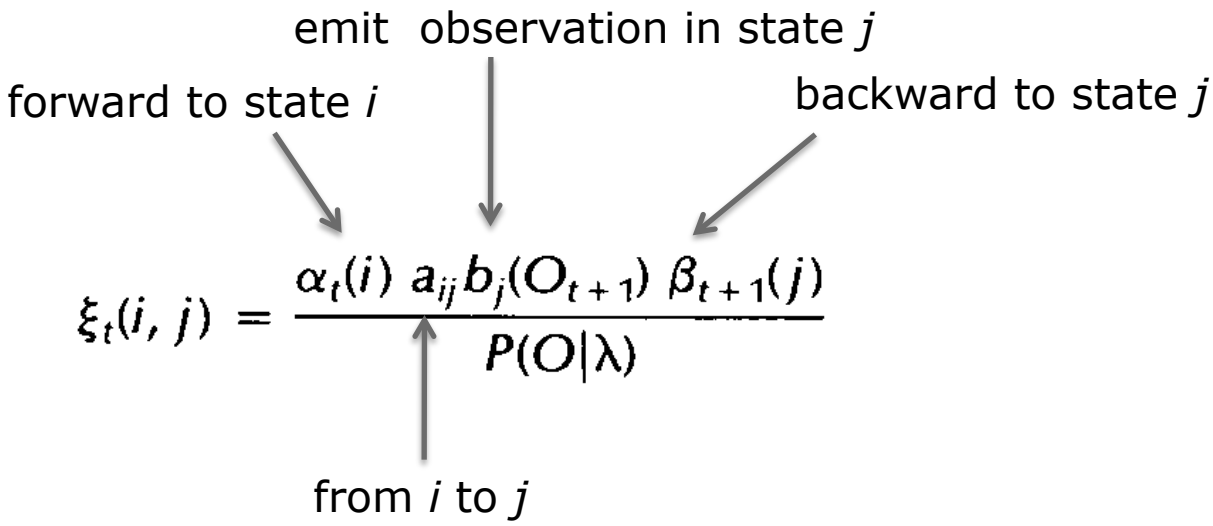
hidden markov models: the three problems

1 given a set of observations **o** = $\{o_1, o_2 \ldots o_T\}$ and a model λ, compute $P(\mathbf{o}|\lambda)$

2 given **o**, λ, calculate most likely latent states (usually thought of as labels) **q** = $\{q_1, q_2 \ldots q_T\}$

3 given **o**, calculate λ

EM for λ



define $$\xi_t(i, j) = \frac{\alpha_t(i)\, a_{ij} b_j(O_{t+1})\, \beta_{t+1}(j)}{P(O|\lambda)}$$

(the probability of being in state *i* at time *t* and state *j* at time *t*+1)

EM for λ

probability of being in state *i* is:

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i, j).$$

(we have to transition *somewhere*)

$$\sum_{t=1}^{T-1} \gamma_t(i) = \text{expected number of transitions from } S_i$$

$$\sum_{t=1}^{T-1} \xi_t(i, j) = \text{expected number of transitions from } S_i \text{ to } S_j.$$

EM for λ

$$\bar{\pi}_i = \text{expected frequency (number of times) in state } S_i \text{ at time } (t = 1) = \gamma_1(i)$$

$$\bar{a}_{ij} = \frac{\text{expected number of transitions from state } S_i \text{ to state } S_j}{\text{expected number of transitions from state } S_i}$$

$$= \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

$$\bar{b}_j(k) = \frac{\text{expected number of times in state } j \text{ and o}}{\text{expected number of times}}$$

$$= \frac{\sum_{\substack{t=1 \\ \text{s.t. } O_t = v_k}}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}.$$

all of these estimates can be computed from the observed data! (by counting!)

EM for λ

at a given time *t* we have

$$\overline{\lambda} = (\overline{A}, \overline{B}, \overline{\pi})$$

E-step calculate the likelihood of our observations **o** using the current estimates

M-step re-estimate the parameters (left-hand sides of equations on previous slide) using current estimates

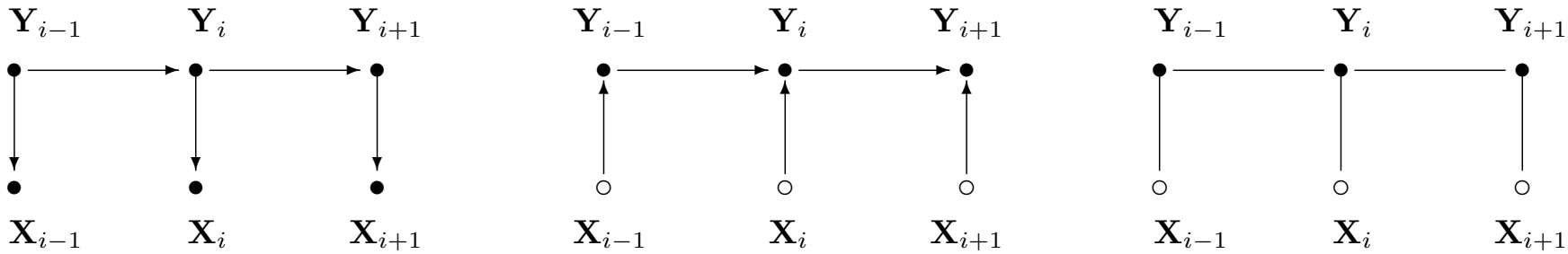
shortcomings of HMMs

- HMMs model the *joint* probability of the observations (**x**) and (**y**)
  - ie., it's a generative model

- but what we *really* care about is the *conditional probability* of a label sequence **y** given **x**

- enter *conditional markov models*

conditional models



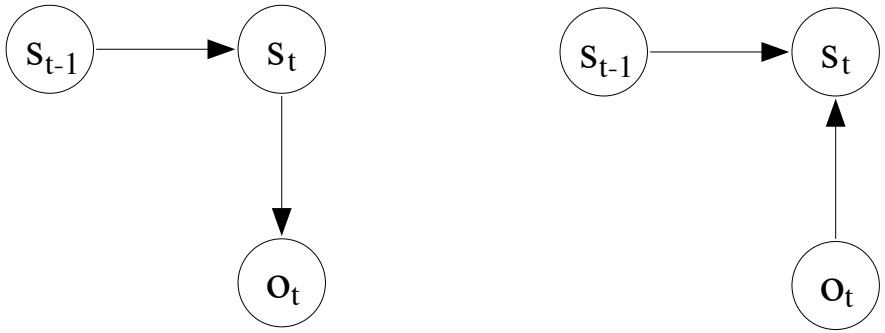
HMM MEMM CRF

conditional models don't waste time modeling the observed data

MEMMs (2000, McCallum et al)

- MEMM: Maximum Entropy Markov Model

- output probability vector of transitioning from one state to all other states, given an input observation **x**
  - *conditional* in that we estimate p(state) at time *t* given an observation and the preceding state



HMM

MEMM

MEMMs v. HMM

- HMM prob. of emitting **o** and being in state *s* at time *t*

$$\alpha_{t+1}(s) = \sum_{s' \in S} \alpha_t(s') P(s|s') P(o_{t+1}|s)$$

- MEMM

$$\alpha_{t+1}(s) = \sum_{s' \in S} \alpha_t(s') P_{s'}(s|o_{t+1})$$

CRFs (2001, Lafferty et al)

- CRF: Conditional Random Field

- single model for the joint probability of the sequence of labels given the observations

- mitigates the *label bias* problem in which states with low-entropy (almost certain) transition probability vectors effectively ignore the observation