EECS 496: Sequential Decision Making

Soumya Ray

sray@case.edu

Office: Olin 516

Office hours : T 4-5:30 or by appointment

Recap

- We estimate model parameters from data using the technique of _____ _____ _____. This finds parameters that _____ .
- For HMMs, the parameters are _____.
- Suppose the observations are annotated with state sequences. Then the emission MLEs are? The transition MLEs?
- If the observations are NOT annotated with state sequences, we use a general technique, called ___ ____.
- In this technique we first ____. Then we ___ two steps. In the ___ step, we find the ____ ____ of the missing information (or sometimes the _____ ___ the missing information). In the ___ step, we find the ____ given the previous. Eventually this procedure ____ to a ___ _____.
- When applied to HMMs this is called the ___ ___ algorithm.

Today

- Sequential Probabilistic Models (Ch 15)

Baum-Welch Algorithm

- Start by randomly initializing all the emission and transition probabilities
- Proceed in two steps until (local) convergence
  - E step: Since we don't know the state sequences, we calculate the *expected* number of times each transition or emission is used (as if annotating observations with *possible* state sequences and averaging over them)
  - M step: Update the emission and transition probabilities using the expected counts (same as in the fully observed case above)

Baum-Welch Algorithm (E step)

Expected number of times b is emitted by state k according to our training sample: $E_{k;b}$

For each training sequence, for each time step where b is present, find the probability k was used at that point; add up all the probabilities

Pr(S=k emitted this symbol)?

$O^1 = o_1 \; o_2 \; o_3$ **b** ….. o_n

$O^2 = o_1 \; o_2 \; \mathbf{b}$ ….. o_m

$O^3 = o_1 \; o_2 \; o_3$ ….. o_k

$O^4 = \mathbf{b} \; o_2 \; o_3$ …. **b** o_p

$O^5 = o_1 \; o_2 \; o_3$ …. o_n

$O^6 = \mathbf{b} \; \mathbf{b} \; \mathbf{b}$ …. o_q

Baum-Welch Algorithm (E step)

Expected number of times b is emitted by state k according to our training sample: $E_{k;b}$

For each training sequence, for each time step where b is present, find the probability k was used at that point; add up all the probabilities

$$E_{k;b} = \sum_{j \in sequences} \sum_{i|o_i^j = b} \Pr(s_i = k \mid \mathbf{o}^j)$$

$$= \sum_{j \in sequences} \frac{1}{\Pr(\mathbf{o}^j)} \sum_{i|o_i^j = b} \Pr(s_i = k, \mathbf{o}^j)$$

The Backward Algorithm

- Let $\beta_k(i) = \Pr(o_{i+1}, \ldots, o_n \mid s_i = k)$
- Calculated similarly to $\alpha_k(i)$
- Then notice that:

Prob of emitting sequence AND *using state k at i^{th} time step*

$$\Pr(s_i = k, \{o_1, \ldots, o_n\})$$
$$= \Pr(\{o_1, \ldots, o_i\}, s_i = k) \Pr(\{o_{i+1}, \ldots, o_n\} \mid \{o_1, \ldots, o_i\}, s_i = k)$$
$$= \Pr(\{o_1, \ldots, o_i\}, s_i = k) \Pr(\{o_{i+1}, \ldots, o_n\} \mid s_i = k)$$
$$= \alpha_k(i)\beta_k(i)$$

Baum-Welch Algorithm (E step)

Expected number of times b is emitted by state k according to our training sample:
For each training sequence, for each time step where b is present, find the probability k was used at that point; add up all the probabilities

$$E_{k;b} = \sum_{j \in sequences} \sum_{i|o_i^j = b} \Pr(s_i = k \mid \mathbf{o}^j)$$

$$= \sum_{j \in sequences} \frac{1}{\Pr(\mathbf{o}^j)} \sum_{i|o_i^j = b} \Pr(s_i = k, \mathbf{o}^j) = \sum_{j \in sequences} \sum_{i|o_i^j = b} \frac{\alpha_{k,j}(i)\beta_{k,j}(i)}{\alpha_{END,j}(n)}$$

Baum-Welch Algorithm (E step)

Expected number of times a transition $k \rightarrow l$ is used:
For each training sequence, for each time step, find the probability $k \rightarrow l$ was used at that point; add up all the probabilities

$$\Pr(s_i = k, s_{i+1} = l \mid \mathbf{o})$$

$$= \frac{\alpha_k(i) \Pr(s_{i+1} = l \mid s_i = k) \Pr(o_{i+1} \mid s_{i+1}) \beta_l(i+1)}{\Pr(\mathbf{o})}$$

$$C_{kl} = \sum_{j \in sequences} \frac{1}{\Pr(\mathbf{o}^j)} \sum_i \alpha_{k,j}(i) \Pr(s_{i+1} = l \mid s_i = k) \Pr(o_{i+1} \mid s_{i+1}) \beta_{l,j}(i+1)$$

Baum-Welch Algorithm (M-Step)

- Emission distribution

$$\Pr(a \mid S = k) = \frac{E_{k;a}}{\sum_{b} E_{k;b}}$$

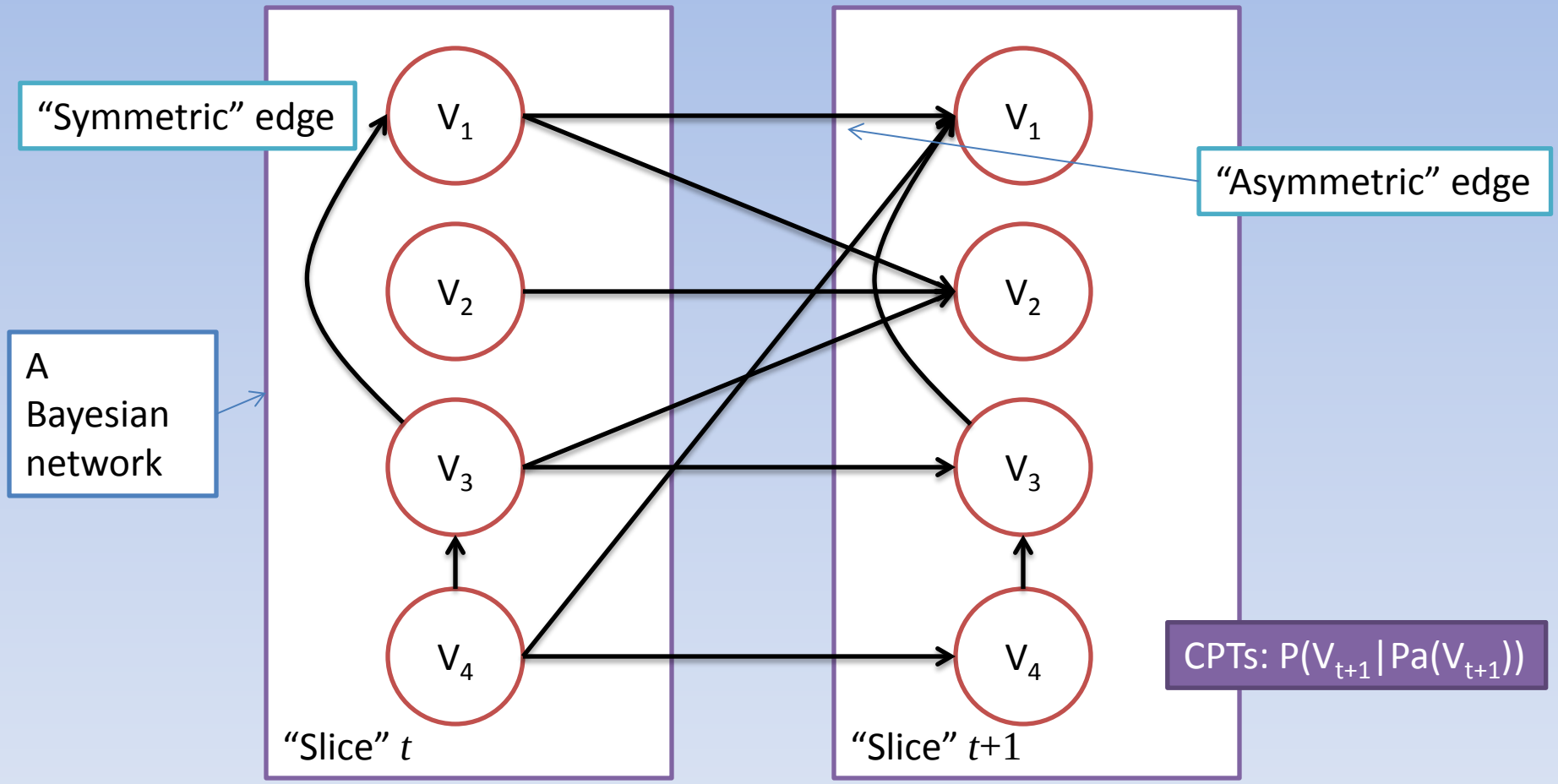
- Transition Distribution

$$\Pr(S = k \mid S = j) = \frac{C_{jk}}{\sum_{l} C_{jl}}$$

Dynamic Bayesian Networks

- HMMs have *atomic* states
- In most problems, states have structure and the observations are some aspect of this structure
- Sequential probabilistic models with *factored* state representations are called Dynamic Bayesian Networks (DBNs)

DBN Architecture



Inference in DBNs

- Since a DBN is a BN, we could technically answer queries by “unrolling” it and applying variable elimination
- However this is generally computationally expensive and becomes increasingly infeasible over time
- Therefore we must use approximate inference algorithms

Particle Filtering/ Sequential Monte Carlo

- An approach that extends likelihood weighting to sequential probabilistic models
- Problem: Must avoid “unrolling” probabilistic models
- So at each time step, need to represent the joint distribution over the variables, then propagate to the next step

Particle Filtering/ Sequential Monte Carlo

- In PF, we adopt a nonparamteric representation: *use a set of weighted samples* ("particles") to approximate the underlying distribution

- Parameter: Sample size set N

Particle Filtering

1. Pick an initial set of N samples from a prior over the initial state distribution S_0
2. Each time step:
    1. Propagate each sample *i* forward by sampling the next state from $P(V^i_{t+1}|Pa(V^i_{t+1}))$.
    2. Weight the sample by the likelihood it assigns to any evidence at t+1: $w^i = P(e_{t+1}|\mathbf{V}_{t+1})$
    3. Resample a set of particles from the weighted sample in step 2. This set is unweighted.

Particle Filtering

- It can be shown that the procedure is consistent (see book): the fraction of samples in any state is proportional to the true probabilities for being in that state (for large enough *N*)