Automatic Speech Recognition (CS753)

Lecture 6: Hidden Markov Models (Part II)

Instructor: Preethi Jyothi

Lecture 6

Recall: Computing Likelihood

Problem 1 (Likelihood):	Given an HMM $\lambda = (A, B)$ and an observation sequence O, determine the likelihood $P(O\|\lambda)$.
Problem 2 (Decoding):	Given an observation sequence O and an HMM $\lambda = (A, B)$, discover the best hidden state sequence Q.
Problem 3 (Learning):	Given an observation sequence O and the set of states in the HMM, learn the HMM parameters A and B.

Computing Likelihood: Given an HMM $\lambda = (A, B)$ and an observation sequence O, determine the likelihood $P(O|\lambda)$.

Recall: Decoding best state sequence

Problem 1 (Likelihood):	Given an HMM $\lambda = (A, B)$ and an observation sequence O, determine the likelihood $P(O\|\lambda)$.
Problem 2 (Decoding):	Given an observation sequence O and an HMM $\lambda = (A, B)$, discover the best hidden state sequence Q.
Problem 3 (Learning):	Given an observation sequence O and the set of states in the HMM, learn the HMM parameters A and B.

Decoding: Given as input an HMM $\lambda = (A, B)$ and a sequence of observations $O = o_1, o_2, ..., o_T$, find the most probable sequence of states $Q = q_1 q_2 q_3 \dots q_T$.

Learning HMM Parameters

Problem 1 (Likelihood):	Given an HMM $\lambda = (A, B)$ and an observation sequence O, determine the likelihood $P(O\|\lambda)$.
Problem 2 (Decoding):	Given an observation sequence O and an HMM $\lambda = (A, B)$, discover the best hidden state sequence Q.
Problem 3 (Learning):	Given an observation sequence O and the set of states in the HMM, learn the HMM parameters A and B.

Learning: Given an observation sequence O and the set of possible states in the HMM, learn the HMM parameters A and B.

Standard algorithm for HMM training: **Forward-backward** or **Baum-Welch** algorithm

Fitting Parameters to Data

Given:

1. a probabilistic model with yet-to-be-determined parameters for generating data samples

2. a collection of (independent) data samples

Goal:

Determine the “best” values for the parameters: probability assigned to the observed data be made as large as possible (a.k.a. MLE parameters)

$$\arg\max_\theta L_{Data}(\theta), \text{ where } L_{Data}(\theta) = Pr_{model(\theta)}[Data]$$

How?

Fitting Parameters to Data: EM

High-level idea/structure of the Expectation-Maximization (EM) algorithm:

In many models, if the data included all the state variables (i.e., no hidden variables), can find MLE parameters analytically

When hidden variables involved, iteratively estimate the parameters as follows: roughly, use parameters from previous rounds to estimate hidden variables and then recompute optimal parameters

Actually, works with distributions over hidden variables

$$Q(\theta, \theta^{t-1}) = E_{model(\theta^{t-1})} \left[\log(L_{Data,Hidden}(\theta) \mid Data) \right]$$ **E step**

$$\theta^{t} = \arg\max_{\theta} Q(\theta, \theta^{t-1})$$ ***M* step**

EM is guaranteed to converge to a local optimum [Wu83, Jeff Wu, “On the Convergence Properties of the EM Algorithm”, Ann. Statist, 11(1), 1983].

Learning HMM Parameters

Problem 1 (Likelihood):	Given an HMM $\lambda = (A, B)$ and an observation sequence O, determine the likelihood $P(O\|\lambda)$.
Problem 2 (Decoding):	Given an observation sequence O and an HMM $\lambda = (A, B)$, discover the best hidden state sequence Q.
Problem 3 (Learning):	Given an observation sequence O and the set of states in the HMM, learn the HMM parameters A and B.

Learning: Given an observation sequence O and the set of possible states in the HMM, learn the HMM parameters A and B.

Standard algorithm for HMM training: **Forward-backward** or **Baum-Welch** algorithm (special case of EM)

Forward/Backward Probabilities

Require two probabilities to compute estimates for the transition and observation probabilities

1. **Forward** probability: Recall $\alpha_t(j) = P(o_1, o_2 \ldots o_t, q_t = j|\lambda)$
2. **Backward** probability: $\beta_t(i) = P(o_{t+1}, o_{t+2} \ldots o_T|q_t = i, \lambda)$

Backward probability

1. **Initialization:**

$$\beta_T(i) \;=\; a_{iF}, \quad 1 \leq i \leq N$$

2. **Recursion** (again since states 0 and q_F are non-emitting):

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij}\, b_j(o_{t+1})\, \beta_{t+1}(j), \quad 1 \leq i \leq N, 1 \leq t < T$$

3. **Termination:**

$$P(O|\lambda) = \alpha_T(q_F) = \beta_1(q_0) = \sum_{j=1}^{N} a_{0j}\, b_j(o_1)\, \beta_1(j)$$

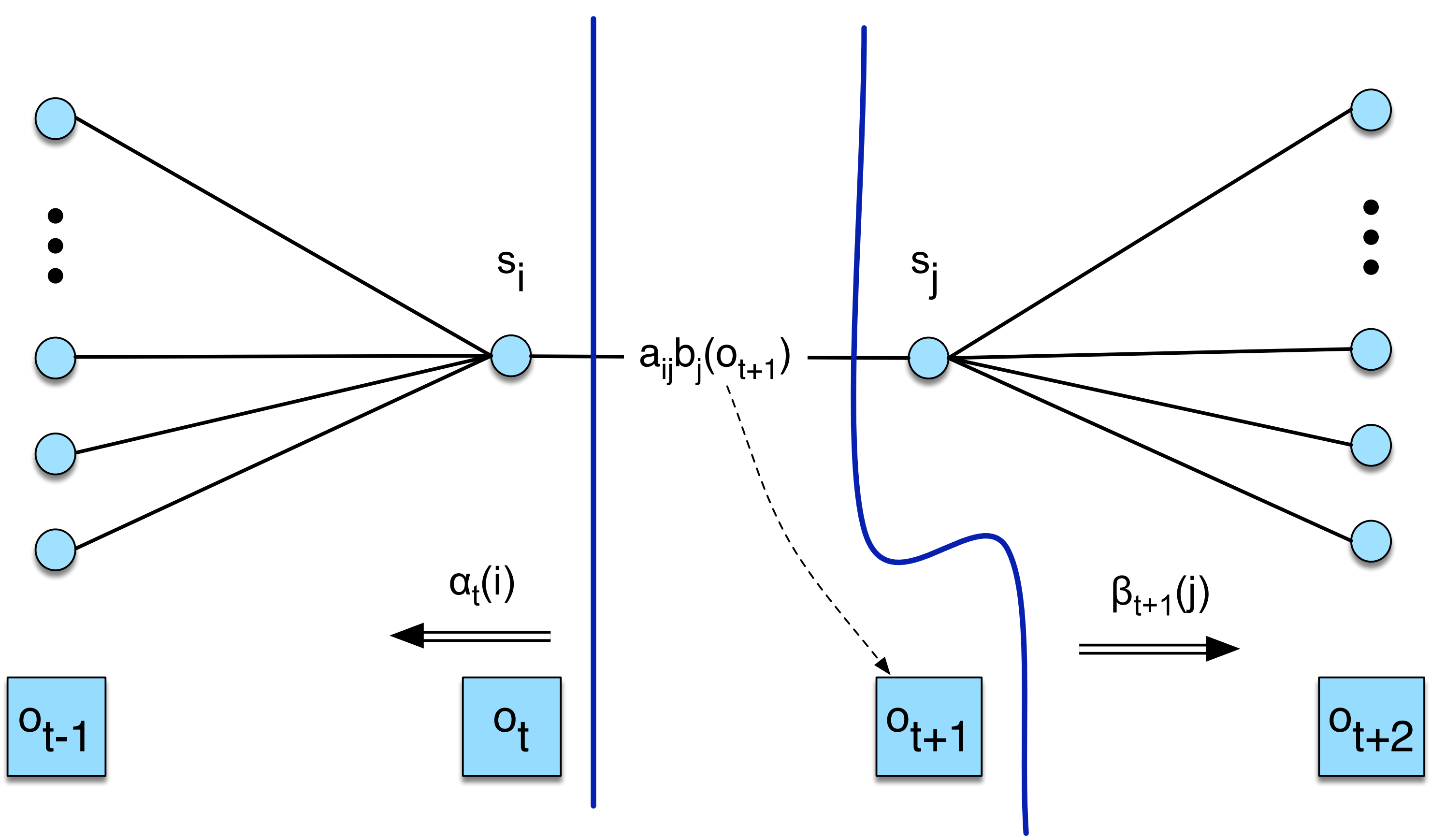
1. Baum-Welch: Estimating a_{ij}

Define a new quantity $\xi_t(i,j)$ to estimate a_{ij}

where $\xi_t(i,j) = P(q_t = i, q_{t+1} = j | O, \lambda)$

which works out as $\xi_t(i,j) = \dfrac{\alpha_t(i)\, a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\alpha_T(q_F)}$

Then, $\hat{a}_{ij} = \dfrac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \sum_{k=1}^{N} \xi_t(i,k)}$



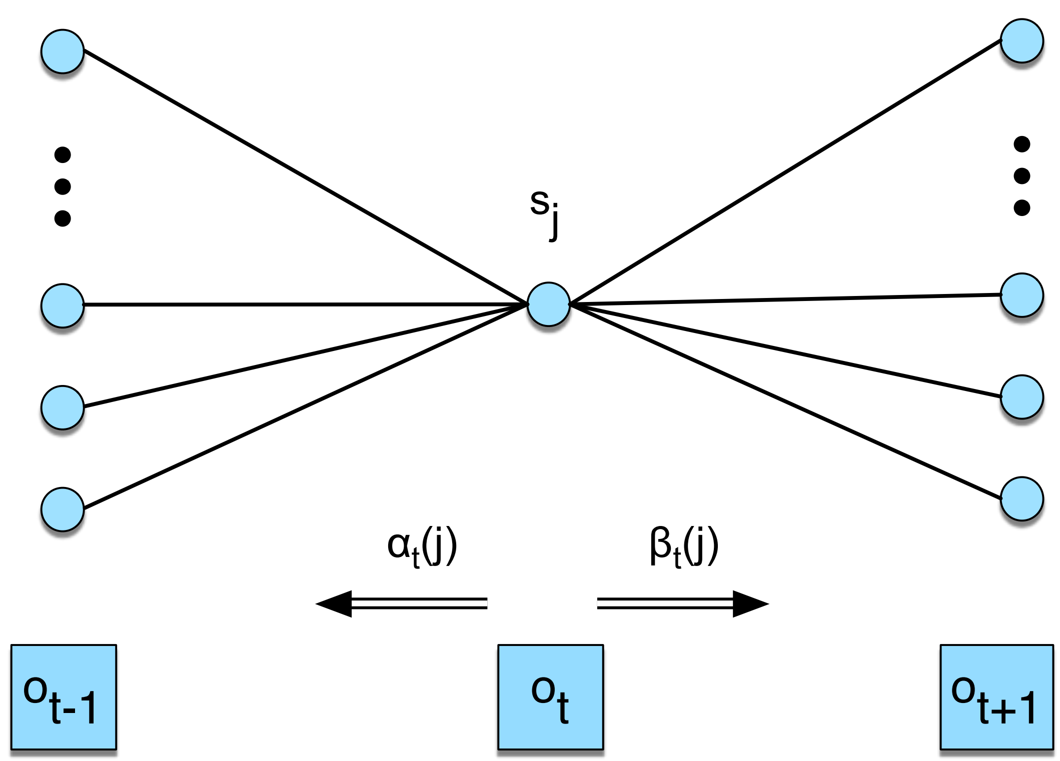
2. Baum-Welch: Estimating $b_i(o_t)$

Define a new quantity $\gamma_t(j)$ to estimate $b_i(o_t)$

where $\gamma_t(j) = P(q_t = j | O, \lambda)$

which works out as $\gamma_t(j) = \frac{\alpha_t(j)\beta_t(j)}{P(O|\lambda)}$

Then, $\hat{b}_j(v_k) = \frac{\sum_{t=1 s.t. O_t=v_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}$



Bringing it all together: Baum-Welch

- Estimating HMM parameters iteratively using the EM algorithm. For each iteration, do:

  E step For all time-state pairs, compute the state occupation probabilities $\gamma_t(j)$ and $\xi_t(i, j)$

  M step Reestimate the HMM parameters based on the estimates derived in the E step: transition probabilities, observation probabilities

Baum-Welch algorithm (pseudocode)

function FORWARD-BACKWARD(*observations* of len T, *output vocabulary* V, *hidden state set* Q) **returns** *HMM=(A,B)*

initialize A and B
iterate until convergence
E-step

$$\gamma_t(j) = \frac{\alpha_t(j)\beta_t(j)}{\alpha_T(q_F)} \quad \forall\, t \text{ and } j$$

$$\xi_t(i,j) = \frac{\alpha_t(i)\,a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}{\alpha_T(q_F)} \quad \forall\, t,\ i, \text{ and } j$$

M-step

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\sum_{k=1}^{N}\xi_t(i,k)}$$

$$\hat{b}_j(v_k) = \frac{\sum_{t=1 s.t.\ O_t=v_k}^{T}\gamma_t(j)}{\sum_{t=1}^{T}\gamma_t(j)}$$

return A, B

Gaussian (normal) distribution

A common probability distribution that can be used for HMM observation probabilities $b_i(o_t)$

$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(x-\mu)^2}$$

μ	$\mathbb{E}[X]$ is the mean
σ^2	$\mathrm{var}[X]$ is the variance
$X \sim \mathcal{N}(x\|\mu,\sigma^2)$	$p(X=x) = \mathcal{N}(x\|\mu,\sigma^2)$

Real data is not always Gaussian! More generally, use an arbitrary number of Gaussians a.k.a mixture of Gaussians

Next class: Gaussian Mixtures and EM