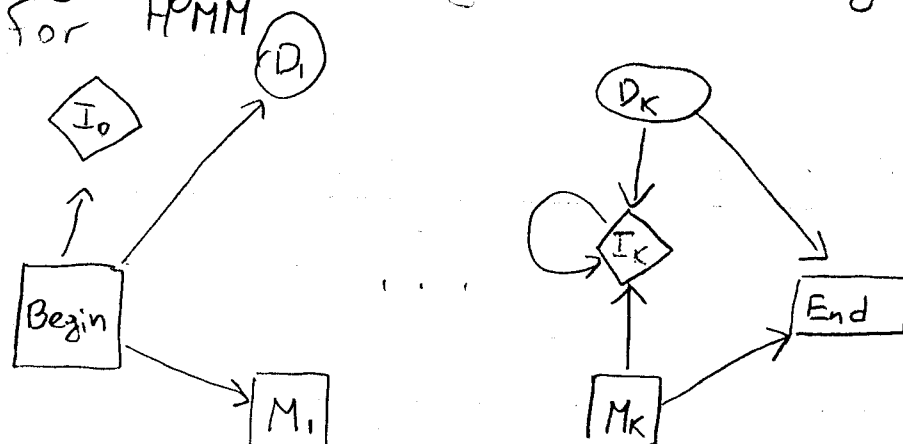


①

Lecture 12May 9, 2005Vadim Lyubashevsky  
59Beginning and Ending the Viterbi Algorithm  
for HMM $\pi$  is concentrated on  $I_0, M_1, D_1$ In addition, define  $a_{\ell, \text{end}}$ ,  $\ell = D_K, I_K, M_K$ .

$$V(O^*) = \max_{\ell = D_K, I_K, M_K} (a_{\ell, \text{end}} \cdot \delta_L(\ell))$$

Parameter estimation (No silent states)

Problem 1: Assume  $N$  is fixed,  $M$  is fixed  
(Training) Possibly assume some topology for HMM

Consider we have sample outputs  $\{\theta^i\}_{i=1}^K$   
from HMMNeed to find  $\lambda = (\pi, a, b)$ :

$$\arg \max_{\lambda} \sum_{i=1}^K S^{\lambda}(\theta^i)$$

$$\text{For example, } S^{\lambda}(\theta^i) = \log P^{\lambda}(\theta^i)$$

Problem 3:

$$\arg \max_{\lambda} P^{\lambda}(\theta) \text{ given } \theta$$

(Classification)

Focus for simplicity on Problem 3.

(Problem 1 is similar)

②

Difficulty:  $\lambda \rightarrow P^\lambda(\theta)$  typically has lots of local maxima

- $\exists$  Good algorithms for finding local maxima
- Can, in addition, do simulated annealing or use multistart to find best of a set of local maxima.

Two methods for local maxima:

(i) Baum-Welch algorithm (special case of EM algorithm)

expectation (maximization)

(1977) Dempster, Laird, Rubin

(ii) Gradient ascent (Baldi-Chauvin, 1994)

Baum-Welch (iterative algorithm)

Fix  $\lambda$ .

Seek  $\bar{\lambda} = (\bar{\pi}, \bar{a}, \bar{b})$  s.t.  $P^{\bar{\lambda}}(\theta) > P^\lambda(\theta)$

or  $P^{\bar{\lambda}}(\theta)$  is a local max of

$\lambda \rightarrow P^\lambda(\theta)$

Fix  $\theta^* = (\theta_0^*, \dots, \theta_{T-1}^*)$

③

Define updated estimates:

$$\bar{\pi}_i = P^\lambda(Q_0 = i | \theta^*)$$

$$\bar{a}_{ij} = \frac{E^\lambda[\# \text{ transitions from } i \text{ to } j \text{ in } [0, T-2] | \theta^*]}{E^\lambda[\# \text{ transitions from } i \text{ in } [0, T-2] | \theta^*]}$$

$$= \frac{\sum_{t=0}^{T-2} P^\lambda(Q_t = i, Q_{t+1} = j | \theta^*)}{\sum_{t=0}^{T-2} P^\lambda(Q_t = i | \theta^*)}$$

$$\sum_{t=0}^{T-2} 1_{\{Q_t = i, Q_{t+1} = j\}}$$

$$\bar{b}_{jk} = \frac{E^\lambda[\# \text{ times in state } j \text{ and output } k \text{ in } [0, T-1] | \theta^*]}{E^\lambda[\# \text{ times in state } j | \theta^*]}$$

$$= \frac{\sum_{t=0}^{T-1} P^\lambda(Q_t = j, \theta_t = k | \theta^*)}{\sum_{t=0}^{T-1} P^\lambda(Q_t = j | \theta^*)}$$

$$= \frac{\sum_{\substack{t=0 \\ \theta_t^* = k}}^{T-1} P^\lambda(Q_t = j, \theta_t = k | \theta^*)}{\sum_{t=0}^{T-1} P^\lambda(Q_t = j | \theta^*)}$$

4

# Efficient Computation

$$\xi_t^\lambda(i, j) = P^\lambda(Q_t = i, Q_{t+1} = j | \theta^*) \quad t = 0, \dots, T-2$$

In the forward algorithm,

$$\alpha_t(i) = P^\lambda(Q_t = i, \theta_t = \theta^* | \theta^*)$$

$$\rightarrow \frac{P^\lambda(Q_t = i, Q_{t+1} = j, \theta = \theta^*)}{P^\lambda(\theta^*)}$$

$$= \alpha_t(i) a_{ij} b_{j|\theta_{t+1}^*} \beta_{t+1}(j)$$

## Backward algorithm

$$\beta_{t+1}(j) = P^\lambda(Q_{t+1} = j, (\theta_{t+2}, \dots, \theta_{T-1}) = (\theta_{t+2}^*, \dots, \theta_{T-1}^*) | \theta^*)$$

$$\beta_{T-1}(j) = 1$$

$$\beta_t(i) = \sum_{j=1}^N a_{ij} b_{j|\theta_{t+1}^*} \beta_{t+1}(j)$$

$$\bar{\pi}_i = P^\lambda(Q_0 = i | \theta^*)$$

$$= \sum_{j=1}^N \xi_0^\lambda(i, j)$$

$$\bar{a}_{ij} = \frac{\sum_{t=0}^{T-2} P^\lambda(Q_t = i, Q_{t+1} = j | \theta^*)}{\sum_{t=0}^{T-2} P^\lambda(Q_t = i | \theta^*)}$$

$$= \frac{\sum_{t=0}^{T-2} \xi_t^\lambda(i, j)}{\sum_{j=1}^N \sum_{t=0}^{T-2} \xi_t^\lambda(i, j)}$$

⑤

$$\bar{b}_{j,k} = \frac{\sum_{t=0}^{T-1} P^\lambda(Q_t = j, O_t = k | \theta^*)}{\sum_{t=0}^{T-1} P^\lambda(Q_t = j | \theta^*)}$$

$$= \frac{\sum_{t=0}^{T-2} \sum_{\substack{q=1 \\ O_t^*=k}}^N \tau_t(j, l) + P^\lambda(Q_{T-1}=j, O_{T-1}=k | \theta^*) 1_{\{O_{T-1}^*=k\}}}{\sum_k \sum_{t=0}^{T-2} \sum_{\substack{q=1 \\ O_t^*=k}}^N \tau_t(j, l) + P^\lambda(Q_{T-1}=j | \theta^*)}$$

$$\sum_k \sum_{t=0}^{T-2} \sum_{\substack{q=1 \\ O_t^*=k}}^N \tau_t(j, l) + P^\lambda(Q_{T-1}=j | \theta^*)$$

Why the update procedure produces a better  $\bar{\lambda}$

$$\begin{aligned} \text{Define } M(\lambda, \bar{\lambda}) &= E^\lambda[\log P^{\bar{\lambda}}(Q, O^*) | \theta^*] \\ &= \sum_{\substack{q=(q_0, \dots, q_{T-1})}} P^\lambda(Q=q | \theta^*) \log P^{\bar{\lambda}}(q, \theta^*) \end{aligned}$$

$$\text{Claim: } M(\lambda, \bar{\lambda}) > M(\lambda, \lambda) \Rightarrow P^{\bar{\lambda}}(\theta^*) > P^\lambda(\theta^*)$$

Proof:

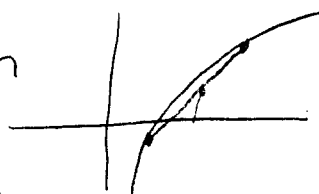
$$\begin{aligned} & \log \left( \frac{P^{\bar{\lambda}}(\theta^*)}{P^\lambda(\theta^*)} \right) \quad \leftarrow R^\lambda(q) \\ &= \log \left( \sum_q \frac{P^{\bar{\lambda}}(Q=q, \theta^*) / P^\lambda(Q=q, \theta^*)}{P^\lambda(Q=q, \theta^*)} \right) \end{aligned}$$

$$R^\lambda(q) = P^\lambda(Q=q | \theta^*)$$

$$\sum_q R^\lambda(q) = 1$$

⑤

$\log$  is a concave function



$$\geq \sum_q \log \left( \frac{p^{\bar{\lambda}}(Q=q, \theta^*)}{p^{\lambda}(Q=q, \theta^*)} \right) p^{\lambda}(Q=q | \theta^*)$$

$$= M(\lambda, \bar{\lambda}) - M(\lambda, \lambda)$$

Follows, if  $M(\lambda, \bar{\lambda}) > M(\lambda, \lambda)$ , then  $p^{\bar{\lambda}}(\theta^*) > p^{\lambda}(\theta^*)$

Using constraint optimization (Lagrange multipliers), can find  $\arg \max_{\bar{\lambda}} M(\lambda, \bar{\lambda})$

This  $\arg \max$  yields the estimates  $\bar{\pi}, \bar{a}, \bar{b}$  in Baum-Welch.

With iteration of  $\lambda$ , keep increasing  $p^{\lambda}(\theta^*)$  until reach a local max  $\lambda$ 's need not converge (may oscillate near local maxs), but can pass to a subsequence for convergence.