Lecture 18:

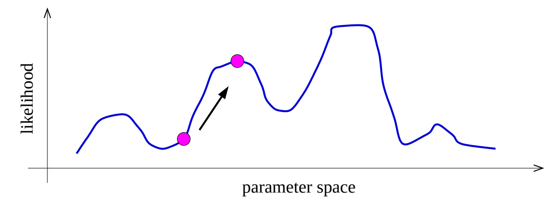
Hidden Markov Model Learning

Sam Roweis

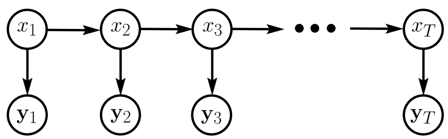
March 10, 2004

Baum-Welch Algorithm: EM Training

1. Intuition: if only we *knew* the true state path then ML parameter estimation would be trivial (MM1 on x, conditional on y).
2. But: can *estimate* state path using inference recursions.
3. *Baum-Welch algorithm* (special case of EM): estimate the states, then compute params, then re-estimate states, and so on ...
4. This works and we can *prove* that it always improves likelihood.
5. However: finding the ML parameters is NP complete, so initial conditions matter a lot and convergence is hard to tell.



Reminder: HMM Graphical Model



- Hidden states $\{x_t\}$, outputs $\{\mathbf{y}_t\}$
  Joint probability factorizes:

$$\mathsf{P}(\{x\},\{\mathbf{y}\}) = \prod_{t=1}^{T} \mathsf{P}(x_t|x_{t-1})\mathsf{P}(\mathbf{y}_t|x_t)$$
$$= \pi_{x_1} \prod_{t=1}^{T-1} S_{x_t,x_{t+1}} \prod_{t=1}^{T} A_{x_t}(\mathbf{y}_t)$$

- We saw efficient recursions for computing
  $L = \mathsf{P}(\{\mathbf{y}\}) = \sum_{\{x\}} \mathsf{P}(\{x\},\{\mathbf{y}\})$ and $\gamma_i(t) = \mathsf{P}(x_t = i|\{\mathbf{y}\})$.

Parameter Estimation using EM

- S_{ij} are transition probs; state j has output distribution $A_j(\mathbf{y})$

$$\mathsf{P}(x_{t+1} = j|x_t = i) = S_{ij} \qquad \mathsf{P}(x_1 = j) = \pi_j$$
$$\mathsf{P}(\mathbf{y}_t = y|x_t = j) = A_j(y)$$

- Complete log likelihood:

$$\log p(x,y) = \log\{\pi_{x_1} \prod_{t=1}^{T-1} S_{x_t,x_{t+1}} \prod_{t=1}^{T} A_{x_t}(\mathbf{y}_t)\}$$
$$= \log\{\prod_i \pi_i^{[x_1^i]} \prod_{t=1}^{T-1} \prod_{ij} S_{ij}^{[x_t^i,x_{t+1}^j]} \prod_{t=1}^{T} \prod_k A_k(\mathbf{y}_t)^{[x_t^k]}\}$$
$$= \sum_i [x_1^i] \log \pi_i + \sum_{t=1}^{T-1} \sum_{ij} [x_t^i, x_{t+1}^j] \log S_{ij} + \sum_{t=1}^{T} \sum_k [x_t^k] \log A_k(\mathbf{y}_t)$$

  where the indicator $[x_t^i] = 1$ if $x_t = i$ and 0 otherwise

- For EM, we need to compute the *expected complete log likelihood*.

State expectations required from the E-Step

- The expected complete log likelihood requires $\gamma_i(t) = < [x_t^i] >$ *and* $\xi_{ij}(t) = < [x_t^i, x_{t+1}^j] >$
- So in the E-step we need to compute both $\gamma_i(t) = p(x_t = i | \{\mathbf{y}\})$ and $\xi_{ij}(t) = p(x_t = i, x_{t+1} = j | \{\mathbf{y}\})$.
- We already know how to compute $\gamma_i(t)$ using α and β recursions. We can compute $\xi_{ij}(t)$ the same way (recall BP):

$$\begin{aligned}
\xi_{ij}(t) &= p(x_{it}, x_{jt+} | \{\mathbf{y}\}) = p(x_{it} | \{\mathbf{y}\}) p(x_{jt+} | x_{it}, \{\mathbf{y}\}) \\
&= p(x_{it}, y_1^t | y_{t+1}^T) p(x_{jt+} | x_{it}, y_{t+1}^T) / p(y_1^t | y_{t+1}^T) \\
&= \frac{p(x_{it}, y_1^t) p(y_{t+1}^T | x_{it}, y_1^t)}{p(y_1^t | y_{t+1}^T) p(y_{t+1}^T)} \frac{p(y_{t+1}^T | x_{jt+}, x_{it}) p(x_{jt+} | x_{it})}{p(y_{t+1}^T | x_i = t)} \\
&= \frac{p(x_{it}, y_1^t) p(y_{t+1}^T | x_{it})}{p(y_1^T)} \frac{p(y_{t+1} | x_{jt+}) p(y_{t+2}^T | x_{jt+}) p(x_{jt+} | x_{it})}{p(y_{t+1}^T | x_i = t)} \\
&= \alpha_i(t) A_j(y_{t+1}) S_{ij} \beta_j(t+1) / L
\end{aligned}$$

HMM Practicalities

- Multiple observation sequences: can be dealt with by averaging numerators and averaging denominators in the ratios given above.
- Initialization: mixtures of Naive Bayes or mixtures of Gaussians
- Numerical scaling: the probability values that the bugs carry get tiny for big times and so can easily underflow. Good rescaling trick:

$$\rho_t = \mathsf{P}(\mathbf{y}_t | \mathbf{y}_1^{t-1}) \qquad \alpha(t) = \tilde{\alpha}(t) \prod_{t'=1}^{t} \rho_{t'}$$

or represent all probabilities as logs and use `logsum`

M-step: New Parameters are just Ratios of Frequency Counts

- Initial state distribution: expected #times in state i at time 1:

$$\hat{\pi}_i = \gamma_i(1)$$

- Expected #transitions from state i to j which begin at time t:

$$\xi_{ij}(t) = \alpha_i(t) S_{ij} A_j(\mathbf{y}_{t+1}) \beta_j(t+1) / L$$

so the estimated transition probabilities are:

$$\hat{S}_{ij} = \sum_{t=1}^{T-1} \xi_{ij}(t) \bigg/ \sum_{t=1}^{T-1} \gamma_i(t)$$

- The output distributions are the expected number of times we observe a particular symbol in a particular state:

$$\hat{A}_j(y_0) = \sum_{t | \mathbf{y}_t = y_0} \gamma_j(t) \bigg/ \sum_{t=1}^{T} \gamma_j(t)$$

M-step for Profile HMMs

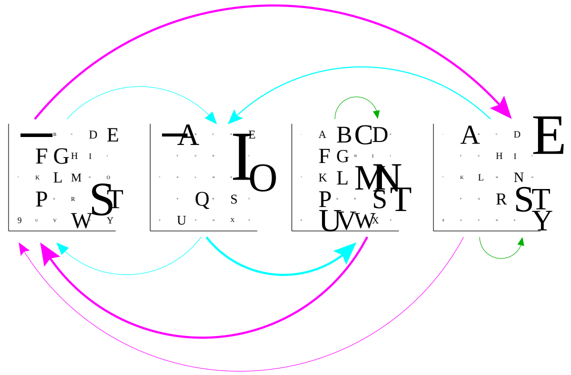
- The emission probabilities $A_j()$ for match and insert states and the initial state distribution π (for m_1, i_1, d_1) are updated exactly as in the regular M-step.
- The expected #transitions from state i to j which begin at time t are different when j is a delete state:

$$\xi_{ij}(t) = \alpha_i(t) S_{ij} \beta_j(t) / L$$

- Given this change, the updates to the transition parameters is the same as in the normal M-step.

Symbol HMM Example

- Character sequences (discrete outputs)



Mixture HMM Example

- Geyser data (continuous outputs)