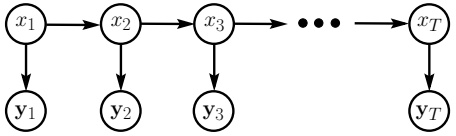
Lecture 17:

Profile HMMs & Hidden Markov Model Learning

March 13, 2006

Reminder: HMM Graphical Model 1



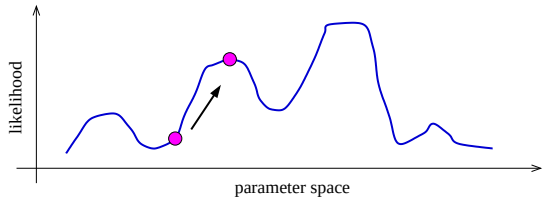
- Hidden states $\{x_t\}$, outputs $\{\mathbf{y}_t\}$
  Joint probability factorizes:

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

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

Baum-Welch Algorithm: EM Training 2

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.



Parameter Estimation using EM 3

- 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 $[x_t = i]$ equals 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 4

- 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
  $\gamma_i(t) = p(x_t = i|\{y\})$ and $\xi_{ij}(t) = p(x_t = i, x_{t+1} = j|\{y\})$.
- 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):

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

Recall: y_r^s is a shorthand for the subsequence $y_r, \ldots, y_s$ and $\{y\}$ is a shorthand for the entire sequence $y_1, \ldots, y_T$.

M-step: New Parameters are just Ratios of Frequency Counts 5

- 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:

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

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

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

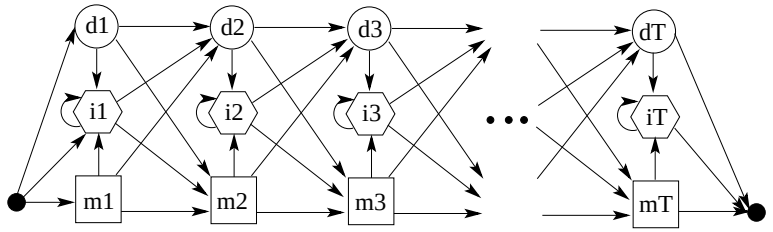
HMM Practicalities 6

- 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`

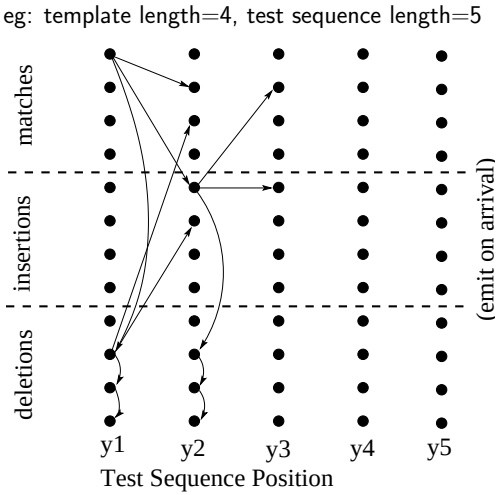
Profile (String-Edit) HMMs 7



i = insert d = delete m = match (state transition diagram)

- A "profile HMM" or "string-edit" HMM is used for probabilistically matching an observed input string to a stored template pattern with possible insertions and deletions.
- Three kinds of states: match, insert, delete.
  m_n – use position n in the template to match an observed symbol
  i_n – insert extra symbol(s) observations after template position n
  d_n – delete (skip) template position n

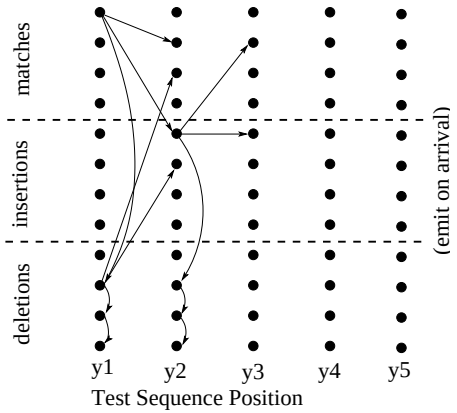
- How do we fill in the costs for a DP grid using a string-edit HMM?
- Almost the same as normal except:
  - Now the grid is 3 times its normal height.
  - It is possible to move down without moving right if you move into a deletion state.



$C_{x \to x'} = -\log T_{x,x'} - \log A_{x'}(\mathbf{y}_t)$ if x' is match or insert

$C_{x \to x'} = -\log T_{x,x'}$ if x' is a delete state

State $x \in \{m_n, i_n, d_n\}$ has nonzero transition probabilities only to states $x' \in \{m_{n+1}, i_n, d_{n+1}\}$.



- The equations for the delete states in profile HMMs need to be modified slightly, since they don't emit any symbols.
- For delete states k, the forward equations become:

$$\alpha_k(t) = \sum_j \alpha_j(t) S_{jk}$$

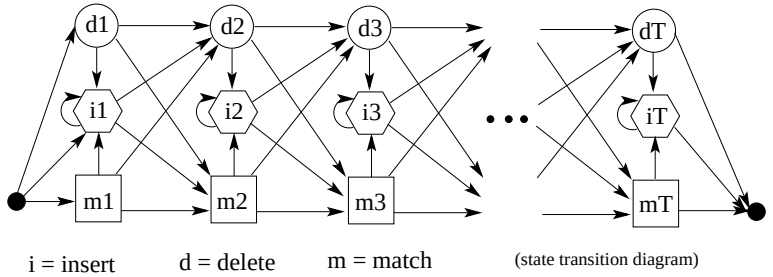
  which should be evaluated after the insert and match state updates.
- For all states, the backward equations become:

$$\beta_k(t) = \sum_{i \in \text{match,ins}} S_{ki} \beta_i(t+1) A_i(\mathbf{y}_{t+1}) + \sum_{j \in \text{del}} S_{kj} \beta_j(t)$$

  which should be evaluated first for delete states k; then for the rest.
- The gamma equations remain the same:

$$\gamma_i(t) = p(x_t = i \mid \mathbf{y}_1^T) = \alpha_i(t)\beta_i(t)/L$$

- Notice that each summation above contains only three terms, regardless of the number of states!



- number of states = 3(length_template)
- Only insert and match states can generate output symbols.
- Once you visit or skip a match state you can never return to it.
- At most 3 destination states from any state, so S_{ij} very sparse.
- Storage/Time cost *linear* in #states, not quadratic.
- State variables and observations no longer in sync. (e.g. y1:m1 ; d2 ; y2:i2 ; y3:i2 ; y4:m3 ; ...)

Initializing Forward-Backward for Profile HMMs 12

- The initialization equations for Profile HMMs also need to be fixed up, to reflect the fact that the model can only begin in states m_1, i_1, d_1 and can only finish in states m_N, i_N, d_N.
- In particular, $\pi_j = 0$ if j is not one of m_1, i_1, d_1.
- When initializing $\alpha_k(1)$, delete states k have zeros, and all other states have the product of the transition probabilities through only delete states up to them, plus the final emission probability.
- When initializing $\beta_k(T)$, the same kind of adjustment must be made.

M-step for Profile HMMs 13

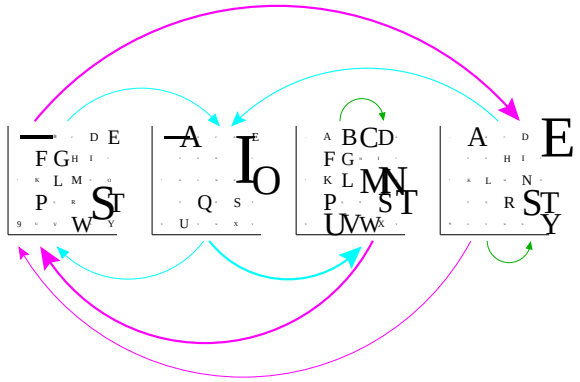
- The emission probabilities $A_j()$ for match and insert states and the initial state distribution π (for m_1, i_1, d_1) are updated exactly as in the regular M-step.
- The expected #transitions from state i to j which begin at time t are different when j is a delete state:

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

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

Symbol HMM Example 14

- Character sequences (discrete outputs)



Mixture HMM Example 15

- Geyser data (continuous outputs)