Lecture 18: Hidden Markov Model Learning

Sam Roweis

March 10, 2004

Reminder: HMM Graphical Model

\[ P(\{x\}, \{y\}) = \prod_{t=1}^{T} P(x_t|x_{t-1})P(y_t|x_t) \]

\[ = \pi_{x_1} \prod_{t=1}^{T-1} S_{x_t,x_{t+1}} \prod_{t=1}^{T} A_{x_t}(y_t) \]

• Hidden states \( \{x_t\} \), outputs \( \{y_t\} \)
  Joint probability factorizes:

\[ P(\{x\}, \{y\}) = \sum_x P(\{x\}, \{y\}) \] and \( \gamma_i(t) = P(x_t = i|\{y\}) \).

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.

Parameter Estimation using EM

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

\[ P(x_{t+1} = j|x_t = i) = S_{ij} \quad P(x_1 = j) = \pi_j \]

\[ P(y_t = y|x_t = j) = A_j(y) \]

• Complete log likelihood:

\[ \log p(x, y) = \log \{ \pi_{x_1} \prod_{t=1}^{T-1} S_{x_t,x_{t+1}} \prod_{t=1}^{T} A_{x_t}(y_t) \} \]

\[ = \log \{ \prod_i \pi_i \prod_{t=1}^{T-1} S_{ij}^{x_t,x_{t+1}} \prod_{t=1}^{T} A_k(y_t)[x_t] \} \]

\[ = \sum_i \log \pi_i + \sum_{t=1}^{T-1} \sum_{ij} S_{ij}^{x_t,x_{t+1}} \log S_{ij} + \sum_{t=1}^{T} \sum_{k} [x_t] \log A_k(y_t) \]

where the indicator \( [x_t] = 1 \) 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
  \[ \gamma_i(t) = < [x_i^t] > \quad \text{and} \quad \xi_{ij}(t) = < [x_i^t, x_{i+1}^t] > \]
- So in the E-step we need to compute both
  \[ \gamma_i(t) = p(x_t = i | \{y\}) \quad \text{and} \quad \xi_{ij}(t) = p(x_t = i, x_{t+1} = j | \{y\}) \]
- We already know how to compute \( \gamma_i(t) \) using \( \alpha \) and \( \beta \) recursions. We can compute \( \xi_{ij}(t) \) the same way (recall BP):
  \[
  \xi_{ij}(t) = \frac{p(x_i^t, x_{j+1}^t | \{y\})}{p(x_i^t, x_{j+1}^t | \{y\})} = \frac{p(x_i^t, y_i^t T_{i+1}^1 p(x_{j+1}^t | x_i^t, y_i^t) p(x_{j+1}^t | x_i^t, y_i^t) p(x_{j+1}^t | x_i^t)}{p(y_i^t T_{i+1}^1 p(y_{i+1}^t | x_i^t)} = \frac{p(x_i^t, y_i^t) p(y_{i+1}^t | x_i^t)}{p(y_{i+1}^t | x_i^t)} = \alpha_i(t) A_j(y_{i+1}) S_{ij} \beta_j(t + 1) / L
  \]

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 \( t \):
  \[ \hat{\xi}_{ij}(t) = \alpha_i(t) S_{ij} A_j(y_{t+1}) \beta_j(t + 1) / L \]
  so the estimated transition probabilities are:
  \[ \hat{S}_{ij} = \sum_{t=1}^{T-1} \hat{\xi}_{ij}(t) / \sum_{t=1}^{T-1} \gamma_i(t) \]
- The output distributions are the expected number of times we observe a particular symbol in a particular state:
  \[ \hat{A}_j(y_0) = \sum_{t | y_t = y_0} \gamma_j(t) / \sum_{t=1}^{T} \gamma_j(t) \]

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:
  \[ \rho_t = p(y_t | y_t^{t-1}) \quad \alpha(t) = \bar{\alpha}(t) \prod_{\epsilon = 1}^{t} \rho_\epsilon \]
  or represent all probabilities as logs and use logsum

M-step for Profile HMMs

- The emission probabilities \( A_j() \) for match and insert states and the initial state distribution \( \pi \) (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:
  \[ \hat{\xi}_{ij}(t) = \alpha_i(t) S_{ij} \beta_j(t + 1) / L \]
- 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)