The EM Algorithm

Observed data: \( O \) : “ball sequence”
Latent data: \( S \) : “bottle sequence”

Parameters to be estimated to maximize \( \log P(O | \lambda) \)
\( \lambda = \{ P(A), P(B), P(B|A), P(A|B), P(R|A), P(G|A), P(R|B), P(G|B) \} \)
The EM Algorithm

- Introduction of EM (Expectation Maximization):
  - Why EM?
    - Simple optimization algorithms for likelihood function relies on the intermediate variables, called latent (隱藏的)data
      In our case here, the state sequence is the latent data
    - Direct access to the data necessary to estimate the parameters is impossible or difficult
      In our case here, it is almost impossible to estimate \( \{A, B, \pi\} \) without consideration of the state sequence
  - Two Major Steps:
    - **E**: expectation with respect to the latent data using the current estimate of the parameters and conditioned on the observations \( E S|\lambda, o \)
    - **M**: provides a new estimation of the parameters according to Maximum likelihood (ML) or Maximum A Posterior (MAP) Criteria
The EM Algorithm

**ML and MAP**

- Estimation principle based on observations:
  \[
  x = (x_1, x_2, \ldots, x_n) \quad \leftrightarrow \quad X = \{X_1, X_2, \ldots, X_n\}
  \]

- **The Maximum Likelihood (ML) Principle**
  find the model parameter \( \Phi \) so that the likelihood \( p(x|\Phi) \) is maximum
  for example, if \( \Phi=\{\mu, \Sigma\} \) is the parameters of a multivariate normal distribution, and \( X \) is i.i.d. (independent, identically distributed), then the ML estimate of \( \Phi=\{\mu, \Sigma\} \) is
  \[
  \mu_{ML} = \frac{1}{n} \sum_{i=1}^{n} x_i, \quad \Sigma_{ML} = \frac{1}{n} \sum_{i=1}^{n} (x_i - \mu_{ML})(x_i - \mu_{ML})^t
  \]

- **The Maximum A Posteriori (MAP) Principle**
  find the model parameter \( \Phi \) so that the likelihood \( p(\Phi|x) \) is maximum
The EM Algorithm

- The EM Algorithm is important to HMMs and other learning techniques
  - Discover new model parameters to maximize the log-likelihood of incomplete data \( \log P(O|\lambda) \) by iteratively maximizing the expectation of log-likelihood from complete data \( \log P(O,S|\lambda) \)

- Using scalar random variables to introduce the EM algorithm
  - The observable training data \( O \)
    - We want to maximize \( P(O|\lambda) \), \( \lambda \) is a parameter vector
  - The hidden (unobservable) data \( S \)
    - E.g. the component densities of observable data \( O \), or the underlying state sequence in HMMs
The EM Algorithm

- Assume we have $\lambda$ and estimate the probability that each $S$ occurred in the generation of $O$.
- Pretend we had in fact observed a complete data pair $(O, S)$ with frequency proportional to the probability $P(O, S | \lambda)$, to compute a new $\tilde{\lambda}$, the maximum likelihood estimate of $\lambda$.
- Does the process converge?
- **Algorithm** unknown model setting

$$P(O, S | \tilde{\lambda}) = P(S | O, \tilde{\lambda})P(O | \tilde{\lambda}) \quad \text{(Bayes’ rule)}$$

**Log-likelihood expression** and expectation taken over $S$

$$\log P(O | \tilde{\lambda}) = \log P(O, S | \tilde{\lambda}) - \log P(S | O, \tilde{\lambda})$$

$$\log P(O | \tilde{\lambda}) = \sum_{S} \left[ P(S | O, \lambda) \log P(O | \tilde{\lambda}) \right]$$

$$= \sum_{S} \left[ P(S | O, \lambda) \log P(O, S | \tilde{\lambda}) \right] - \sum_{S} \left[ P(S | O, \lambda) \log P(S | O, \tilde{\lambda}) \right]$$
The EM Algorithm

- Algorithm (Cont.)
  - We can thus express \( \log P(O|\lambda) \) as follows
    
    \[
    \log P(O|\lambda) = \sum_s [P(S|O,\lambda)\log P(O,S|\lambda)] - \sum_s [P(S|O,\lambda)\log P(S|O,\lambda)]
    \]
    
    \[= Q(\lambda, \tilde{\lambda}) - H(\lambda, \tilde{\lambda}) \]
    
    where
    
    \[Q(\lambda, \tilde{\lambda}) = \sum_s [P(S|O,\lambda)\log P(O,S|\lambda)] \]
    
    \[H(\lambda, \tilde{\lambda}) = \sum_s [P(S|O,\lambda)\log P(S|O,\lambda)] \]
    
  - We want \( \log P(O|\lambda) \geq \log P(O|\lambda) \)
    
    \[
    \log P(O|\lambda) - \log P(O|\lambda) = [Q(\lambda, \tilde{\lambda}) - H(\lambda, \tilde{\lambda})] - [Q(\lambda, \lambda) - H(\lambda, \lambda)]
    \]
    
    \[= Q(\lambda, \tilde{\lambda}) - Q(\lambda, \lambda) - H(\lambda, \lambda) + H(\lambda, \lambda) \]
The EM Algorithm

• \(-H(\lambda, \bar{\lambda}) + H(\lambda, \bar{\lambda})\) has the following property

\[
-H(\lambda, \bar{\lambda}) + H(\lambda, \bar{\lambda}) = -\sum_s \left[ P(S|O, \lambda) \log \frac{P(S|O, \bar{\lambda})}{P(S|O, \lambda)} \right]
\]

\[
\geq \sum_s \left[ P(S|O, \lambda) \left( 1 - \frac{P(S|O, \bar{\lambda})}{P(S|O, \lambda)} \right) \right] \quad (\because \log x \leq x - 1)
\]

\[
= \sum_s \left[ P(S|O, \lambda) - P(S|O, \bar{\lambda}) \right]
\]

\[
= 0
\]

\[
\therefore -H(\lambda, \bar{\lambda}) + H(\lambda, \bar{\lambda}) \geq 0
\]

Therefore, for maximizing \( \log P(O|\bar{\lambda}) \), we only need to maximize the \( Q \)-function (auxiliary function)

\[
Q(\lambda, \bar{\lambda}) = \sum_s \left[ P(S|O, \lambda) \log P(O, S|\bar{\lambda}) \right]
\]

Jensen's inequality

Kullback-Leibler (KL) distance

Expectation of the complete data log likelihood with respect to the latent state sequences
EM Applied to Discrete HMM Training

- Apply EM algorithm to iteratively refine the HMM parameter vector $\lambda = (A, B, \pi)$
  - By maximizing the auxiliary function

\[
Q(\lambda, \bar{\lambda}) = \sum_{s} \left[ P(S|O, \lambda) \log P(O, S|\bar{\lambda}) \right]
\]

\[
= \sum_{s} \left[ \frac{P(O, S|\lambda)}{P(O|\lambda)} \log P(O, S|\bar{\lambda}) \right]
\]

- Where $P(O, S|\lambda)$ and $P(O, S|\bar{\lambda})$ can be expressed as

\[
P(O, S|\lambda) = \pi_{s_1} \left[ \prod_{t=1}^{T-1} a_{s_t s_{t+1}} \right] \left[ \prod_{t=1}^{T} b_{s_t} (o_t) \right]
\]

\[
\log P(O, S|\lambda) = \log \pi_{s_1} + \sum_{t=1}^{T-1} \log a_{s_t s_{t+1}} + \sum_{t=1}^{T} \log b_{s_t} (o_t)
\]

\[
\log P(O, S|\bar{\lambda}) = \log \pi_{\bar{s}_1} + \sum_{t=1}^{T-1} \log \bar{a}_{s_t s_{t+1}} + \sum_{t=1}^{T} \log \bar{b}_{s_t} (o_t)
\]
EM Applied to Discrete HMM Training

• Rewrite the auxiliary function as

\[ Q(\lambda, \bar{\lambda}) = Q_\pi(\lambda, \bar{\pi}) + Q_a(\lambda, \bar{a}) + Q_b(\lambda, \bar{b}) \]

\[ Q_a(\lambda, \bar{\pi}) = \sum_{S} \left[ \frac{P(O, S | \lambda)}{P(O | \lambda)} \log \bar{\pi}_i \right] = \sum_{i=1}^{N} \left[ \frac{P(O, s_1 = i | \lambda)}{P(O | \lambda)} \log \bar{\pi}_i \right] \]

\[ Q_a(\lambda, \bar{a}) = \sum_{S} \left[ \frac{P(O, S | \lambda)}{P(O | \lambda)} \log \bar{a}_{ij} \right] = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{t=1}^{T-1} \left[ \frac{P(O, s_t = i, s_{t+1} = j | \lambda)}{P(O | \lambda)} \log \bar{a}_{ij} \right] \]

\[ Q_b(\lambda, \bar{b}) = \sum_{S} \left[ \frac{P(O, S | \lambda)}{P(O | \lambda)} \log \bar{b}_j(k) \right] = \sum_{j=1}^{N} \sum_{k} \sum_{t \in a_t = v_k} \left[ \frac{P(O, s_t = j | \lambda)}{P(O | \lambda)} \log \bar{b}_j(k) \right] \]

**Figure 8.7** Illustration of the operations required for the computation of \( \gamma_t(i, j) \), which is the probability of taking the transition from state \( i \) to state \( j \) at time \( t \).
EM Applied to Discrete HMM Training

- The auxiliary function contains three independent terms, $\pi_i$, $a_{ij}$, and $b_j(k)$
  - Can be maximized individually
  - All of the same form

$$F(y) = g(y_1, y_2, ..., y_N) = \sum_{j=1}^{N} w_j \log y_j,$$

where $\sum_{j=1}^{N} y_j = 1$, and $y_j \geq 0$

$F(y)$ has maximum value when:

$$y_j = \frac{w_j}{\sum_{j=1}^{N} w_j}$$
EM Applied to Discrete HMM Training

- **Proof**: Apply Lagrange Multiplier

By applying Lagrange Multiplier $\ell$

Suppose that $F = \sum_{j=1}^{N} w_j \log y_j = \sum_{j=1}^{N} w_j \log y_j + \ell \left( \sum_{j=1}^{N} y_j - 1 \right)$

$$\frac{\partial F}{\partial y_j} = \frac{w_j}{y_j} + \ell = 0 \implies \ell = -\frac{w_j}{y_j} \forall \ j$$

$$\ell \sum_{j=1}^{N} y_j = -\sum_{j=1}^{N} w_j \implies \ell = -\sum_{j=1}^{N} w_j$$

$$\therefore y_j = \frac{w_j}{\sum_{j=1}^{N} w_j}$$
EM Applied to Discrete HMM Training

• The new model parameter set $\bar{\lambda} = (\bar{\pi}, \bar{A}, \bar{B})$ can be expressed as:

$$\bar{\pi}_i = \frac{P(O, s_1 = i | \lambda)}{P(O | \lambda)} = \gamma_1(i)$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} P(O, s_t = i, s_{t+1} = j | \lambda)}{\sum_{t=1}^{T-1} P(O, s_t = i | \lambda)} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

$$\bar{b}_i(k) = \frac{\sum_{t=1}^{T} P(O, s_t = i | \lambda) \text{ s.t. } o_t = v_k}{\sum_{t=1}^{T} P(O, s_t = i | \lambda)} = \frac{\sum_{t=1}^{T} \gamma_t(i) \text{ s.t. } o_t = v_k}{\sum_{t=1}^{T} \gamma_t(i)}$$
EM Applied to Continuous HMM Training

• Continuous HMM: the state observation does not come from a finite set, but from a continuous space
  – The difference between the discrete and continuous HMM lies in a different form of state output probability
  – Discrete HMM requires the quantization procedure to map observation vectors from the continuous space to the discrete space

• Continuous Mixture HMM
  – The state observation distribution of HMM is modeled by multivariate Gaussian mixture density functions ($M$ mixtures)

\[
b_j(o) = \sum_{k=1}^{M} c_{jk} b_{jk}(o) = \sum_{k=1}^{M} c_{jk} N(o; \mu_{jk}, \Sigma_{jk}) = \sum_{k=1}^{M} c_{jk} \left( \frac{1}{\sqrt{2\pi}} \left| \Sigma_{jk} \right|^{1/2} \exp \left( -\frac{1}{2} (o - \mu_{jk})^T \Sigma_{jk}^{-1} (o - \mu_{jk}) \right) \right) \]

\[
\sum_{k=1}^{M} c_{jk} = 1
\]
EM Applied to Continuous HMM Training

• Express $b_j(o)$ with respect to each single mixture component $b_{jk}(o)$

\[
P(O, S|\lambda) = \pi_{s_1} \left\{ \prod_{t=1}^{T-1} a_{s_t s_{t+1}} \right\} \left\{ \prod_{t=1}^{T} b_{s_t}(o_t) \right\} = \pi_{s_1} \left\{ \prod_{t=1}^{T-1} a_{s_t s_{t+1}} \right\} \left\{ \sum_{k_1=1}^{M} \sum_{k_2=1}^{M} \cdots \sum_{k_T=1}^{M} \left[ \prod_{t=1}^{T} c_{s_t k_t} b_{s_t k_t}(o_t) \right] \right\}
\]

\[
P(O, S, K|\lambda) = \pi_{s_1} \left\{ \prod_{t=1}^{T-1} a_{s_t s_{t+1}} \right\} \left\{ \prod_{t=1}^{T} \left[ c_{s_t k_t} b_{s_t k_t}(o_t) \right] \right\}
\]

$K$ : one of the possible mixture component sequence along with the state sequence $S$

\[
P(O|\lambda) = \sum_{S} \sum_{K} P(O, S, K|\lambda)
\]
EM Applied to Continuous HMM Training

Therefore, an auxiliary function for the EM algorithm can be written as:

\[
Q(\lambda, \bar{\lambda}) = \sum_S \sum_K \left[ P(S, K | O, \lambda) \log P(O, S, K | \bar{\lambda}) \right]
\]

\[
= \sum_S \sum_K \left[ \frac{P(O, S, K | \lambda)}{P(O | \lambda)} \log P(O, S, K | \bar{\lambda}) \right]
\]

\[
\log P(O, S, K | \bar{\lambda}) = \log \bar{\pi}_{s_1} + \sum_{t=1}^{T-1} \log \bar{a}_{s_ts_{t+1}} + \sum_{t=1}^{T} \log \bar{b}_{s_{t}k_{t}}(o_{t}) + \sum_{t=1}^{T} \log \bar{c}_{s_{t}k_{t}}
\]

\[
Q(\lambda, \bar{\lambda}) = Q_\pi(\lambda, \bar{\pi}) + Q_a(\lambda, \bar{a}) + Q_b(\lambda, \bar{b}) + Q_c(\lambda, \bar{c})
\]

- **Q_\pi** : initial probabilities
- **Q_a** : state transition probabilities
- **Q_b** : Gaussian density functions
- **Q_c** : mixture components
EM Applied to Continuous HMM Training

• The only difference we have when compared with Discrete HMM training

\[
Q_b(\lambda, \bar{b}) = \sum_{t=1}^{T} \left\{ \sum_{j=1}^{N} \sum_{k=1}^{M} P(s_t = j, k_t = k | O, \lambda) \log \bar{b}_{jk}(o_t) \right\}
\]

\[
Q_c(\lambda, \bar{c}) = \sum_{t=1}^{T} \left\{ \sum_{j=1}^{N} \sum_{k=1}^{M} P(s_t = j, k_t = k | O, \lambda) \log \bar{c}_{jk}(o_t) \right\}
\]
EM Applied to Continuous HMM Training

Let \( \gamma_t(j,k) = \sum_{k=1}^{M} P(s_t = j, k_t = k|\mathcal{O}, \lambda) \)

\[
\bar{b}_{jk}(o_t) = N(o_t; \mu_{jk}, \Sigma_{jk}) = \frac{1}{(2\pi)^{L/2} |\Sigma_{jk}|^{1/2}} \exp\left(-\frac{1}{2}(o_t - \mu_{jk})^{T} \Sigma_{jk}^{-1} (o_t - \mu_{jk})\right)
\]

\[
\log \bar{b}_{jk}(o_t) = -\frac{L}{2} \log (2\pi) + \frac{1}{2} \log |\Sigma_{jk}^{-1}| - \left(\frac{1}{2}(o_t - \mu_{jk})^{T} \Sigma_{jk}^{-1} (o_t - \mu_{jk})\right)
\]

\[
\frac{\partial \log \bar{b}_{jk}(o_t)}{\partial \mu_{jk}} = \Sigma_{jk}^{-1} (o_t - \mu_{jk})
\]

\[
\partial Q_b(\lambda, \bar{b}) = \frac{\partial}{\partial \mu_{jk}} \left\{ \sum_{t=1}^{T} \sum_{j=1}^{N} \sum_{k=1}^{M} \gamma_t(j,k) \log \bar{b}_{jk}(o_t) \right\}
\]

\[
\Rightarrow \sum_{t=1}^{T} \{ \gamma_t(j,k) \Sigma_{jk}^{-1} (o_t - \mu_{jk}) \} = 0
\]

\[
\Rightarrow \mu_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k) \cdot o_t}{\sum_{t=1}^{T} \gamma_t(j,k)}
\]

\[d \left( x^T C x \right) = (C + C^T) x \]

and \( \Sigma_{jk}^{-1} \) is symmetric here
EM Applied to Continuous HMM Training

\[
\log \overline{b}_{jk}(o_t) = -\frac{L}{2} \cdot \log (2\pi) + \frac{1}{2} \cdot \log |\overline{\Sigma}_{jk}^{-1}| - \left( \frac{1}{2} (o_t - \overline{\mu}_{jk}) \overline{\Sigma}_{jk}^{-1}(o_t - \overline{\mu}_{jk}) \right)
\]

\[
\log \frac{\partial \overline{b}_{jk}(o_t)}{\partial (\overline{\Sigma}_{jk}^{-1})} = \frac{1}{2} \cdot |\overline{\Sigma}_{jk}| \cdot |\overline{\Sigma}_{jk}^{-1}| \cdot \overline{\Sigma}_{jk} - \left( \frac{1}{2} (o_t - \overline{\mu}_{jk})(o_t - \overline{\mu}_{jk}) \right)
\]

\[
\frac{\partial Q_b(\lambda, \overline{b})}{\partial (\overline{\Sigma}_{jk}^{-1})} = \frac{\partial}{\partial (\overline{\Sigma}_{jk}^{-1})} \sum_{t=1}^{T} \left\{ \sum_{j=1}^{N} \sum_{k=1}^{M} \gamma_t(j,k) \log \overline{b}_{jk}(o_t) \right\}
\]

\[
\Rightarrow \sum_{t=1}^{T} \left\{ \gamma_t(j,k) \frac{1}{2} \left| \overline{\Sigma}_{jk} \right| - (o_t - \overline{\mu}_{jk})(o_t - \overline{\mu}_{jk}) \right\} = 0
\]

\[
\Rightarrow \overline{\Sigma}_{jk} = \frac{\sum_{t=1}^{T} \left[ \gamma_t(j,k)(o_t - \overline{\mu}_{jk})(o_t - \overline{\mu}_{jk}) \right]}{\sum_{t=1}^{T} \gamma_t(j,k)}
\]

\[
d \left( a^T X b \right) = a b^T
\]

\[
d [\det(X)] = \det(X) \cdot X^{-T}
\]
EM Applied to Continuous HMM Training

- The new model parameter set for each mixture component and mixture weight can be expressed as:

\[
\bar{\mu}_{jk} = \frac{\sum_{t=1}^{T} \left[ P(O, s_t = j, k_t = k | \lambda) \right] o_t}{\sum_{t=1}^{T} P(O, s_t = j, k_t = k | \lambda)} \quad = \quad \frac{\sum_{t=1}^{T} [\gamma_t (j, k) o_t]}{\sum_{t=1}^{T} \gamma_t (j, k)}
\]

\[
\bar{\Sigma}_{jk} = \frac{\sum_{t=1}^{T} \left[ P(O, s_t = j, k_t = k | \lambda) \right] (o_t - \bar{\mu}_{jk})(o_t - \bar{\mu}_{jk})}{\sum_{t=1}^{T} P(O, s_t = j, k_t = k | \lambda)} \quad = \quad \frac{\sum_{t=1}^{T} [\gamma_t (j, k)(o_t - \bar{\mu}_{jk})(o_t - \bar{\mu}_{jk})]}{\sum_{t=1}^{T} \gamma_t (j, k)}
\]

\[
\bar{c}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t (j, k)}{\sum_{t=1}^{T} \sum_{k=1}^{M} \gamma_t (j, k)}
\]