Chapter 12
Mixture Models

supplementary slides to

Machine Learning Fundamentals

published by Cambridge University Press

August 2020

Outline

Mixture Models

- mixture models: a mixture of some component distributions

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = \sum_{m=1}^{M} w_m \cdot f_{\boldsymbol{\theta}_m}(\mathbf{x})$$

  where $\boldsymbol{\theta} = \{w_m, \boldsymbol{\theta}_m \,|\, m = 1, 2, \cdots, M\}$ denotes all model parameters

- mixture weights satisfy $\sum_{m=1}^{M} w_m = 1$
- each component distribution $f_{\boldsymbol{\theta}_m}(\mathbf{x})$ is normally a simpler unimodal distribution, e.g. Gaussian, multinomial,...
- more generally, $f_{\boldsymbol{\theta}}(\mathbf{x})$ is chosen from the **exponential family (e-family)**

Exponential Family (e-family)

- exponential family (e-family) includes all probabilistic models that can be reparameterized as:

$$f_{\boldsymbol{\theta}}(\mathbf{x}) = \exp\Big(A(\bar{\mathbf{x}}) + \bar{\mathbf{x}}^\intercal \boldsymbol{\lambda} - K(\boldsymbol{\lambda})\Big)$$

  - $\boldsymbol{\lambda} = g(\boldsymbol{\theta})$ is called *natural parameters*
  - $\bar{\mathbf{x}} = h(\mathbf{x})$ is called *sufficient statistics*
  - $K(\boldsymbol{\lambda})$ is a normalization term:

$$\int_{\mathbf{x}} f_{\boldsymbol{\theta}}(\mathbf{x}) d\mathbf{x} = 1 \implies K(\boldsymbol{\lambda}) = \ln\Big[\int_{\mathbf{x}} \big(A(h(\mathbf{x})) + (h(\mathbf{x}))^\intercal \boldsymbol{\lambda}\big) d\mathbf{x}\Big]$$

- take logarithm: $\ln f_{\boldsymbol{\theta}}(\mathbf{x}) = A(\bar{\mathbf{x}}) + \bar{\mathbf{x}}^\intercal \boldsymbol{\lambda} - K(\boldsymbol{\lambda})$
- e.g. Gaussian, binomial, multinomial,beta, Dirichlet ...
- products of e-family distributions still belong to e-family

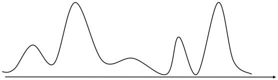
Exponential Family (e-family): Some Examples

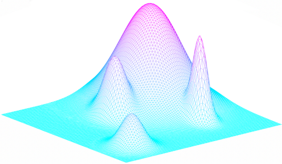
$f_{\boldsymbol{\theta}}(\mathbf{x})$	$\boldsymbol{\lambda} = g(\boldsymbol{\theta})$	$\bar{\mathbf{x}} = h(\mathbf{x})$	$K(\boldsymbol{\lambda})$	$A(\bar{\mathbf{x}})$
univariate Gaussian $\mathcal{N}(x \mid \mu, \sigma^2)$	$[\overbrace{\mu/\sigma^2}^{\lambda_1}, \overbrace{1/\sigma^2}^{\lambda_2}]$	$[x, -x^2/2]$	$-\frac{1}{2}\lambda_1^2/\lambda_2 + \frac{1}{2}\ln(\lambda_2)$	$-\frac{1}{2}\ln(2\pi)$
multivariate Gaussian $\mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$	$\left[\overbrace{\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}}^{\boldsymbol{\lambda}_1}, \overbrace{\boldsymbol{\Sigma}^{-1}}^{\boldsymbol{\lambda}_2}\right]$	$[\mathbf{x}, -\frac{1}{2}\mathbf{x}\mathbf{x}^\intercal]$	$-\frac{1}{2}\boldsymbol{\lambda}_1^\intercal\boldsymbol{\lambda}_2^{-1}\boldsymbol{\lambda}_1 + \frac{1}{2}\ln\lvert\boldsymbol{\lambda}_2\rvert$	$-\frac{d}{2}\ln(2\pi)$
Gaussian (mean only) $\mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}_0)$	$\boldsymbol{\mu}$	$\boldsymbol{\Sigma}_0^{-1}\mathbf{x}$	$-\frac{1}{2}\boldsymbol{\lambda}^\intercal\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\lambda}$	$-\frac{d}{2}\ln(2\pi) - \frac{1}{2}\ln\lvert\boldsymbol{\Sigma}_0\rvert - \frac{1}{2}\mathbf{x}^\intercal\boldsymbol{\Sigma}_0^{-1}\mathbf{x}$
Multinomial $C \cdot \prod_{d=1}^{D} p_d^{x_d}$	$\left[\ln p_1, \cdots, \ln p_D\right]$	$\mathbf{x}$	0	$\ln(C)$

Gaussian mixture model (GMM)

in order to model **multi-modal** distributions of $\mathbf{x} \in \mathbb{R}^d$, we may consider a group of Gaussians:

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = \sum_{m=1}^{M} w_m \cdot \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)$$





- mixture weights w_m satisfy $\sum_{m=1}^{M} w_m = 1$
- mean vector and covariance matrix of m-th Gaussian component: $\boldsymbol{\mu}_m$ and $\boldsymbol{\Sigma}_m$ for all $m = 1, 2, \cdots, M$
- if M is large enough, a GMM can approximate any arbitrary distribution in $\mathbb{R}^d$

Maximum Likelihood Estimation of Mixture Models

- it is not trivial to estimate mixture models
- given some training data $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$
- the log-likelihood function of a mixture model contains *log-sum*
- e.g. the log-likelihood function of GMMs

$$l\Big(\{w_m, \boldsymbol{\mu}_m, \Sigma_m\}\Big) = \sum_{i=1}^{N} \ln \left(\sum_{m=1}^{M} w_m \cdot \mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}_m, \Sigma_m) \right)$$

- can we switch *log-sum* into *sum-log*?

Expectation-Maximization (EM) Method

- log-likelihood function of mixture models:

$$l(\boldsymbol{\theta}) = \sum_{i=1}^{N} \ln p_{\boldsymbol{\theta}}(\mathbf{x}_i) = \sum_{i=1}^{N} \ln \left(\sum_{m=1}^{M} w_m \cdot f_{\boldsymbol{\theta}_m}(\mathbf{x}_i) \right)$$

- treat index m as a **latent variable**: an unobserved random variable taking values in $\{1, 2, \cdots, M\}$
- given any model $\boldsymbol{\theta}^{(n)}$, compute a conditional probability distribution of m based on data $\mathbf{x}_i$:

$$\Pr(m \,|\, \mathbf{x}_i, \boldsymbol{\theta}^{(n)}) = \frac{w_m^{(n)} \cdot f_{\boldsymbol{\theta}_m^{(n)}}(\mathbf{x}_i)}{\sum_{m=1}^{M} w_m^{(n)} \cdot f_{\boldsymbol{\theta}_m^{(n)}}(\mathbf{x}_i)} \qquad (\forall m = 1, 2, \cdots, M)$$

where we have $\sum_{m=1}^{M} \Pr(m \,|\, \mathbf{x}_i, \boldsymbol{\theta}^{(n)}) = 1$ for any $\mathbf{x}_i$

Auxiliary Function $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n)})$ (I)

define an auxiliary function of $\boldsymbol{\theta}$ as follows:

$$
\begin{aligned}
Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n)}) &= \sum_{i=1}^{N} \mathbb{E}_m \Big[\overbrace{\ln\big(w_m \cdot f_{\boldsymbol{\theta}_m}(\mathbf{x}_i)\big)}^{\text{use } \boldsymbol{\theta} \text{ here}} \;\Big|\; \mathbf{x}_i, \boldsymbol{\theta}^{(n)} \Big] + C \\
&= \sum_{i=1}^{N} \sum_{m=1}^{M} \ln\big[w_m \cdot f_{\boldsymbol{\theta}_m}(\mathbf{x}_i)\big] \cdot \Pr(m \mid \mathbf{x}_i, \boldsymbol{\theta}^{(n)}) + C
\end{aligned}
$$

where C is a constant defined as the sum of the entropy of the above conditional probability distributions:

$$
C \triangleq H(\boldsymbol{\theta}^{(n)}|\boldsymbol{\theta}^{(n)}) = -\sum_{i=1}^{N} \sum_{m=1}^{M} \ln \Pr(m \,|\, \mathbf{x}_i, \boldsymbol{\theta}^{(n)}) \Pr(m \,|\, \mathbf{x}_i, \boldsymbol{\theta}^{(n)})
$$

Auxiliary Function $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n)})$ (II)

Theorem 1

the auxiliary function $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n)})$ satisfies the following three properties:

1. $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n)})$ *and* $l(\boldsymbol{\theta})$ *achieve the same value at* $\boldsymbol{\theta}^{(n)}$*:*

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n)})\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(n)}} = l(\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(n)}}$$

2. $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n)})$ *is tangent to* $l(\boldsymbol{\theta})$ *at* $\boldsymbol{\theta}^{(n)}$*:*

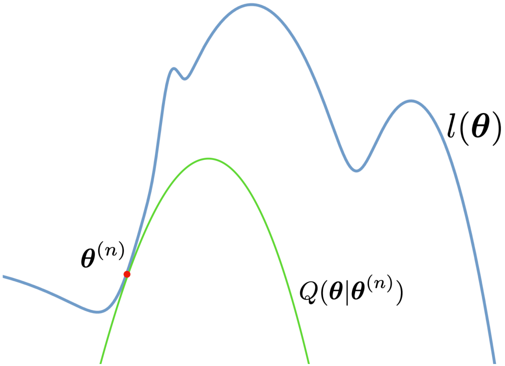
$$\frac{\partial Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n)})}{\partial \boldsymbol{\theta}}\Bigg|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(n)}} = \frac{\partial l(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\Bigg|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(n)}}$$

3. *For all* $\boldsymbol{\theta} \neq \boldsymbol{\theta}^{(n)}$, $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n)})$ *is located strictly below* $l(\boldsymbol{\theta})$*:*

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n)}) < l(\boldsymbol{\theta}) \quad (\forall \boldsymbol{\theta} \neq \boldsymbol{\theta}^{(n)})$$

Auxiliary Function $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n)})$ (III)

the auxiliary function $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n)})$ is related to $l(\boldsymbol{\theta})$ like this:



Auxiliary Function $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n)})$ (IV)

Proof:

- Bayes theorem $Pr(y|x) = \frac{p(x,y)}{p(x)} \implies p(x) = \frac{p(x,y)}{Pr(y|x)}$
- apply to the model $p_{\boldsymbol{\theta}}(m, \mathbf{x})$, we have
  $p_{\boldsymbol{\theta}}(\mathbf{x}) = \frac{p_{\boldsymbol{\theta}}(m,\mathbf{x})}{\Pr(m|\mathbf{x},\boldsymbol{\theta})} \implies \ln p_{\boldsymbol{\theta}}(\mathbf{x}) = \ln p_{\boldsymbol{\theta}}(m,\mathbf{x}) - \ln \Pr(m|\mathbf{x},\boldsymbol{\theta})$
- multiply $\Pr(m|\mathbf{x},\boldsymbol{\theta}^{(n)})$ to both sides, and sum over all $m = \{1, 2, \cdots, M\}$:

$$\begin{aligned}\sum_{m=1}^{M} \ln p_{\boldsymbol{\theta}}(\mathbf{x}) \cdot \Pr(m|\mathbf{x},\boldsymbol{\theta}^{(n)}) &= \sum_{m=1}^{M} \ln p_{\boldsymbol{\theta}}(m,\mathbf{x}) \cdot \Pr(m|\mathbf{x},\boldsymbol{\theta}^{(n)}) \\ &- \sum_{m=1}^{M} \ln \Pr(m|\mathbf{x},\boldsymbol{\theta}) \cdot \Pr(m|\mathbf{x},\boldsymbol{\theta}^{(n)})\end{aligned}$$

Auxiliary Function $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n)})$ (V)

Proof (continued):

- substitute $\mathbf{x}$ with every training sample $\mathbf{x}_i$ in $\mathcal{D}$ and sum over all N samples, so we have

$$\begin{aligned}\sum_{i=1}^{N} \ln p_{\boldsymbol{\theta}}(\mathbf{x}_i) \quad &= \quad \sum_{i=1}^{N}\sum_{m=1}^{M} \ln p_{\boldsymbol{\theta}}(m, \mathbf{x}_i) \cdot \Pr(m|\mathbf{x}_i, \boldsymbol{\theta}^{(n)}) \\ &- \quad \sum_{i=1}^{N}\sum_{m=1}^{M} \ln \Pr(m|\mathbf{x}_i, \boldsymbol{\theta}) \cdot \Pr(m|\mathbf{x}_i, \boldsymbol{\theta}^{(n)})\end{aligned}$$

- we have $\sum_{m=1}^{M} \Pr(m|\mathbf{x}, \boldsymbol{\theta}^{(n)}) = 1$
- $p_{\boldsymbol{\theta}}(m, \mathbf{x}_i) = \Pr(m|\boldsymbol{\theta}) p_{\boldsymbol{\theta}}(\mathbf{x}_i|m) = w_m \cdot f_{\boldsymbol{\theta}_m}(\mathbf{x}_i)$

Auxiliary Function $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n)})$ (VI)

Proof (continued):

- substituting $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n)})$ into the above

$$\begin{aligned}
l(\boldsymbol{\theta}) &= Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n)}) + \left[\sum_{i=1}^{N}\sum_{m=1}^{M} \ln \Pr(m|\mathbf{x}_i, \boldsymbol{\theta}^{(n)}) \Pr(m|\mathbf{x}_i, \boldsymbol{\theta}^{(n)}) \right. \\
&- \left. \sum_{i=1}^{N}\sum_{m=1}^{M} \ln \Pr(m|\mathbf{x}_i, \boldsymbol{\theta}) \Pr(m|\mathbf{x}_i, \boldsymbol{\theta}^{(n)}) \right] \\
&= Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n)}) + \sum_{i=1}^{N} \underbrace{\left[\sum_{m=1}^{M} \ln\left(\frac{\Pr(m|\mathbf{x}_i, \boldsymbol{\theta}^{(n)})}{\Pr(m|\mathbf{x}_i, \boldsymbol{\theta})}\right) \Pr(m|\mathbf{x}_i, \boldsymbol{\theta}^{(n)})\right]}_{\mathsf{KL}\left(\Pr(m|\mathbf{x}_i,\boldsymbol{\theta}^{(n)}) \,||\, \Pr(m|\mathbf{x}_i,\boldsymbol{\theta})\right) \geq 0} \\
&\geq Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n)})
\end{aligned}$$

- properties 1 and 3 are proved

Auxiliary Function $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n)})$ (VII)

Proof (continued):

- from above, we have

$$\frac{\partial l(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \frac{\partial Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n)})}{\partial \boldsymbol{\theta}} - \frac{\partial H(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n)})}{\partial \boldsymbol{\theta}}$$

with

$$\begin{aligned}
\left.\frac{\partial H(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n)})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(n)}} &= \sum_{i=1}^{N} \left[\sum_{m=1}^{M} \frac{\Pr(m|\mathbf{x}_i, \boldsymbol{\theta}^{(n)})}{\Pr(m|\mathbf{x}_i, \boldsymbol{\theta})} \frac{\partial \Pr(m|\mathbf{x}_i, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right] \Bigg|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(n)}} \\
&= \sum_{i=1}^{N} \left[\sum_{m=1}^{M} \frac{\partial \Pr(m|\mathbf{x}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right] \Bigg|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(n)}} \\
&= \sum_{i=1}^{N} \frac{\partial}{\partial \boldsymbol{\theta}} \left[\sum_{m=1}^{M} \Pr(m|\mathbf{x}, \boldsymbol{\theta}) \right] \Bigg|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(n)}} \\
&= \sum_{i=1}^{N} \frac{\partial}{\partial \boldsymbol{\theta}} \Big[\ 1\ \Big] \Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(n)}} = 0 \qquad \blacksquare
\end{aligned}$$

Expectation-Maximization (EM) Algorithm

EM algorithm

initialize $\boldsymbol{\theta}^{(0)}$, set $n = 0$

while not converged **do**

E-step:

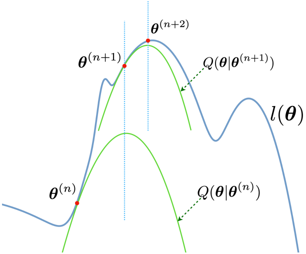
$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n)}) = \sum_{i=1}^{N} \mathbb{E}_m \Big[\ln \big(w_m \cdot f_{\boldsymbol{\theta}_m}(\mathbf{x}_i) \big) \Big| \mathbf{x}_i, \boldsymbol{\theta}^{(n)} \Big]$$

M-step:

$$\boldsymbol{\theta}^{(n+1)} = \arg\max_{\boldsymbol{\theta}} \; Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n)})$$

$n = n + 1$

end while



if $f_{\boldsymbol{\theta}_m}(\mathbf{x})$ belongs to e-family, $Q(\cdot)$ is concave and M-step can be solved in closed-form.

Convergence Analysis of EM algorithm (I)

Theorem 2

Each EM iteration guarantees to improve $l(\boldsymbol{\theta})$:

$$l(\boldsymbol{\theta}^{(n+1)}) \geq l(\boldsymbol{\theta}^{(n)})$$

Furthermore, the improvement of the log-likelihood function is not less than the improvement of the auxiliary function:

$$l(\boldsymbol{\theta}^{(n+1)}) - l(\boldsymbol{\theta}^{(n)}) \geq Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n)})\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(n+1)}} - Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n)})\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(n)}}$$

Convergence Analysis of EM algorithm (II)

Proof:

- property 1 $\implies l(\boldsymbol{\theta}^{(n)}) = Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n)})\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(n)}}$
- M-step $\implies Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n)})\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(n+1)}} \geq Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n)})\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(n)}}$
- property 3 $\implies l(\boldsymbol{\theta}^{(n+1)}) > Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n)})\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(n+1)}}$

$$l(\boldsymbol{\theta}^{(n+1)}) > Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n)})\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(n+1)}} \geq Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n)})\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(n)}} = l(\boldsymbol{\theta}^{(n)})$$

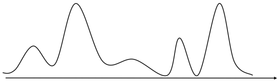
- therefore, we have $l(\boldsymbol{\theta}^{(n+1)}) \geq l(\boldsymbol{\theta}^{(n)})$ and $l(\boldsymbol{\theta}^{(n+1)}) - l(\boldsymbol{\theta}^{(n)}) > Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n)})\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(n+1)}} - Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n)})\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(n)}}$

$\blacksquare$

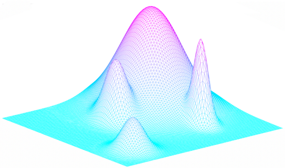
Gaussian mixture model (GMM)

Gaussian mixtures models (GMMs):

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = \sum_{m=1}^{M} w_m \cdot \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)$$



- mixture weights w_m satisfy $\sum_{m=1}^{M} w_m = 1$
- mean vector and covariance matrix of m-th Gaussian component: $\boldsymbol{\mu}_m$ and $\boldsymbol{\Sigma}_m$ for all $m = 1, 2, \cdots, M$
- if M is large enough, a GMM can approximate any arbitrary distribution in $\mathbb{R}^d$



EM algorithm for GMMs (I)

- denote

$$\xi_m^{(n)}(\mathbf{x}) = \Pr(m|\mathbf{x}, \boldsymbol{\theta}^{(n)}) = \frac{w_m^{(n)}\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_m^{(n)}, \Sigma_m^{(n)})}{\sum_{m=1}^{M} w_m^{(n)}\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_m^{(n)}, \Sigma_m^{(n)})}$$

- given a set of training data $\{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$
- E-Step: construct the auxiliary function

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n)}) =$$

$$\sum_{i=1}^{N}\sum_{m=1}^{M}\left[\ln w_m - \frac{\ln|\Sigma_m|}{2} - \frac{(\mathbf{x}_i - \boldsymbol{\mu}_m)^\intercal \Sigma_m^{-1}(\mathbf{x}_i - \boldsymbol{\mu}_m)}{2}\right]\xi_m^{(n)}(\mathbf{x}_i)$$

EM algorithm for GMMs (II)

- M-step: for all $m = 1, 2, \cdots M$

$$\frac{\partial Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n)})}{\partial \boldsymbol{\mu}_m} = 0 \implies \boldsymbol{\mu}_m^{(n+1)} = \frac{\sum_{i=1}^{N} \xi_m^{(n)}(\mathbf{x}_i)\, \mathbf{x}_i}{\sum_{i=1}^{N} \xi_m^{(n)}(\mathbf{x}_i)}$$

$$\frac{\partial Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n)})}{\partial \boldsymbol{\Sigma}_m} = 0 \implies$$

$$\boldsymbol{\Sigma}_m^{(n+1)} = \frac{\sum_{i=1}^{N} \xi_m^{(n)}(\mathbf{x}_i)\,(\mathbf{x}_i - \boldsymbol{\mu}_m^{(n+1)})(\mathbf{x}_i - \boldsymbol{\mu}_m^{(n+1)})^\intercal}{\sum_{i=1}^{N} \xi_m^{(n)}(\mathbf{x}_i)}$$

$$\frac{\partial}{w_m}\left[Q(\cdot) - \lambda\Big(\sum_{m=1}^{M} w_m - 1\Big)\right] = 0 \implies w_m^{(n+1)} = \frac{\sum_{i=1}^{N} \xi_m^{(n)}(\mathbf{x}_i)}{N}$$

EM Algorithm for GMMs

given a training set as $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$

EM algorithm for GMMs

initialize $\left\{w_m^{(0)}, \boldsymbol{\mu}_m^{(0)}, \boldsymbol{\Sigma}_m^{(0)}\right\}$, set $n = 0$

while not converged **do**

E-step: for all $m = 1, \cdots, M$ and $i = 1, \cdots, N$:

$$\left\{w_m^{(n)}, \boldsymbol{\mu}_m^{(n)}, \boldsymbol{\Sigma}_m^{(n)}\right\} \cup \left\{\mathbf{x}_i\right\} \longrightarrow \left\{\xi_m^{(n)}(\mathbf{x}_i)\right\}$$

M-step: for all $m = 1, \cdots, M$:

$$\left\{\xi_m^{(n)}(\mathbf{x}_i)\right\} \cup \left\{\mathbf{x}_i\right\} \longrightarrow \left\{w_m^{(n+1)}, \boldsymbol{\mu}_m^{(n+1)}, \boldsymbol{\Sigma}_m^{(n+1)}\right\}$$

$n = n + 1$

end while

K-means Clustering

use k-means clustering to initialize GMMs:

$$\mathcal{D} \longmapsto M \text{ disjoint clusters: } C_1 \cup C_2 \cdots \cup C_M$$

Top-down K-means Clustering

$k = 1$
initialize the centroid of C_1
while $k \leq M$ **do**
  repeat
    assign each $\mathbf{x}_i \in \mathcal{D}$ to the nearest cluster among $C_1, \cdots, C_k$
    update the centroids for the first k clusters: $C_1, \cdots, C_k$
  until assignments no longer change
  split: split any cluster into two clusters
  $k = k + 1$
end while

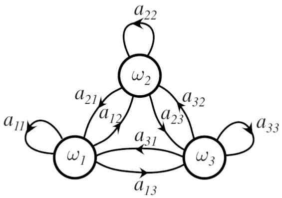
Hidden Markov Models

1. HMMs: mixture models for sequences
2. evaluation problem: Forward-Backward algorithm
3. decoding problem: Viterbi algorithm
4. training problem: Baum-Welch algorithm

Markov Chain Models: Revisit

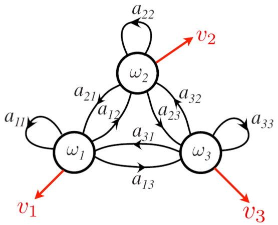
- Markov chain models are unimodal models for sequences
  - given a state sequence $\mathbf{s} = \{\omega_2\omega_1\omega_1\omega_3\}$,

$$\Pr(\mathbf{s}) = \Pr(\omega_2\omega_1\omega_1\omega_3) = \pi_2 \times a_{21} \times a_{11} \times a_{13}$$



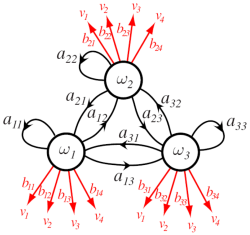
- Markov chain models belong to e-family
- assume each state *deterministically* omits a unique observation symbol
  - for an observation sequence $\mathbf{o} = \{v_2v_1v_1v_3\}$

$$\begin{aligned}\Pr(\mathbf{o}) &= \Pr(v_2v_1v_1v_3) = \Pr(\omega_2\omega_1\omega_1\omega_3) \\ &= \pi_2 \times a_{21} \times a_{11} \times a_{13}\end{aligned}$$



Hidden Markov Models (I)

- hidden Markov models (HMM): mixture models for sequences
- each HMM state can generate all possible symbols based on a unique probability distribution
- HMMs are a doubly-embedded stochastic process to generate symbols
  - *Markov assumption*: state transition is a 1st-order Markov chain
  - *output independence assumption*: the probability of generating an observation only depends on the current state



$$\mathbf{o} = \{v_2 v_1 v_1 v_3\}$$

is generated from

$$\mathbf{s} = \{\omega_2 \omega_1 \omega_1 \omega_3\}$$

$$\Pr(\mathbf{o}, \mathbf{s}) = \pi_2 \times b_{22} \times a_{21} \times b_{11} \times a_{11} \times b_{11} \times a_{13} \times b_{33}$$

Hidden Markov Models (II)

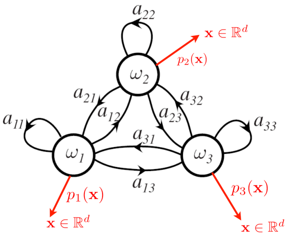
- extend to deal with sequences of continuous observations
- what about the underlying state sequence $\mathbf{s}$ is hidden?
- an HMM has to sum over all possible state sequences:

$$\Pr(\mathbf{o}) = \sum_{\mathbf{s}\in\mathcal{S}} \Pr(\mathbf{o},\mathbf{s})$$

- HMMs are mixture models for sequences:

$$\Pr(\mathbf{o}) = \sum_{\mathbf{s}\in\mathcal{S}} \Pr(\mathbf{s})\cdot p(\mathbf{o}|\mathbf{s})$$

where $\sum_{\mathbf{s}\in\mathcal{S}} \Pr(\mathbf{s}) = 1$



$$\mathbf{o} = \{\mathbf{x}_1\mathbf{x}_2\mathbf{x}_3\mathbf{x}_4\}$$

$$\mathbf{s} = \{\omega_2\omega_1\omega_1\omega_3\}$$

$$\Pr(\mathbf{o},\mathbf{s}) = \pi_2 \times p_2(\mathbf{x}_1)$$

$$\times a_{21} \times p_1(\mathbf{x}_2) \times a_{11} \times p_1(\mathbf{x}_3) \times$$

$$a_{13} \times p_3(\mathbf{x}_4)$$

Hidden Markov Models (III)

- an HMM, denoted as $\mathbf{\Lambda}$, includes:
  - N Markov states: $\Omega = \{\omega_1, \omega_2, \cdots \omega_N\}$
  - initial state probabilities:
    $\boldsymbol{\pi} = \{\pi_i \,|\, i = 1, 2, \cdots N\}$, where $\pi_i = \pi(\omega_i)$
  - state transition probabilities:
    $\mathbf{A} = \{a_{ij} \,|\, 1 \leq i, j \leq N\}$, where $a_{ij} = a(\omega_i, \omega_j)$
  - state-dependent probability distributions:
    $\mathbb{B} = \{b_i(\mathbf{x}) \,|\, i = 1, 2, \cdots N\}$, where $b_i(\mathbf{x}) = b(\mathbf{x}|\omega_i)$
- an HMM can compute the probability of observing any sequence of T observations: $\mathbf{o} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T\}$

$$p_{\mathbf{\Lambda}}(\mathbf{o}) = \sum_{\mathbf{s}} p_{\mathbf{\Lambda}}(\mathbf{o}, \mathbf{s}) = \sum_{s_1 \cdots s_T} \pi(s_1) b(\mathbf{x}_1|s_1) \prod_{t=2}^{T} a(s_{t-1}, s_t) b(\mathbf{x}_t|s_t)$$

$$= \sum_{s_1 \cdots s_T} \pi(s_1) b(\mathbf{x}_1|s_1) a(s_1, s_2) b(\mathbf{x}_2|s_2) \cdots a(s_{T-1}, s_T) b(\mathbf{x}_T|s_T)$$

Evaluation Problem

- how to compute $p_{\mathbf{\Lambda}}(\mathbf{o})$?
- a brute-force method requires to sum $O(N^T)$ terms
- forward algorithm: use dynamic programming method to compute this summation recursively from left to right

$$\sum_{s_1 \cdots s_T} \underbrace{\pi(s_1) b(\mathbf{x}_1|s_1)}_{\alpha_1(s_1)} a(s_1, s_2) b(\mathbf{x}_2|s_2) \cdots a(s_{T-1}, s_T) b(\mathbf{x}_T|s_T)$$

$$= \sum_{s_2 \cdots s_T} \underbrace{\left(\sum_{s_1=1}^{N} \alpha_1(s_1) a(s_1, s_2) b(\mathbf{x}_2|s_2) \right)}_{\alpha_2(s_2)} a(s_2, s_3) \cdots a(s_{T-1}, s_T) b(\mathbf{x}_T|s_T)$$

$$= \sum_{s_3 \cdots s_T} \underbrace{\left(\sum_{s_2=1}^{N} \alpha_2(s_2) a(s_2, s_3) b(\mathbf{x}_3|s_3) \right)}_{\alpha_3(s_3)} a(s_3, s_4) \cdots a(s_{T-1}, s_T) b(\mathbf{x}_T|s_T)$$

Evaluation Problem: Forward Algorithm

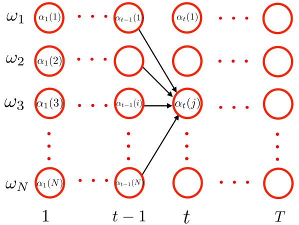
$$\vdots$$

$$= \underbrace{\sum_{s_T} \left(\sum_{s_{T-1}=1}^{N} \alpha_{T-1}(s_{T-1}) a(s_{T-1}, s_T) b(\mathbf{x}_T | s_T) \right)}_{\alpha_T(s_T)} = \sum_{s_T=1}^{N} \alpha_T(s_T)$$

- the above forward procedure requires $O(T \times N^2)$ operations
- denote forward probabilities:

$$\alpha_t(i) \triangleq \alpha_t(s_t) \Big|_{s_t=\omega_i}$$

- run the forward algorithm in a 2-D lattice



Evaluation Problem: Backward Algorithm

backward algorithm: use dynamic programming method to compute recursively from right to left

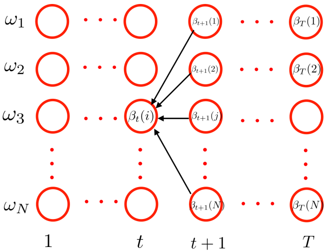
$$\sum_{s_1\cdots s_T} \pi(s_1)b(\mathbf{x}_1|s_1)\cdots a(s_{T-1},s_T)b(\mathbf{x}_T|s_T)$$

$$= \sum_{s_1\cdots s_{T-1}} \pi(s_1)\cdots \underbrace{\left(\sum_{s_T} a(s_{T-1},s_T)b(\mathbf{x}_T|s_T)\right)}_{\beta_{T-1}(s_{T-1})}$$

$$\vdots$$

$$= \sum_{s_1} \pi(s_1)b(\mathbf{x}_1|s_1) \underbrace{\left(\sum_{s_2} a(s_1,s_2)b(\mathbf{x}_2|s_2)\beta_2(s_2)\right)}_{\beta_1(s_1)}$$

$$= \sum_{s_1} \pi(s_1)b(\mathbf{x}_1|s_1)\beta_1(s_1)$$



$$\beta_t(i) \triangleq \beta_t(s_t)\Big|_{s_t=\omega_i}$$

$$\forall t = 1,\cdots,T; i = 1,\cdots,N$$

Evaluation Problem: Forward & Backward Algorithm

HMM forward-backward algorithm

input: an HMM $\mathbf{\Lambda}$ and a sequence $\mathbf{o} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots \mathbf{x}_T\}$
output: $\{\alpha_t(i), \beta_t(i) \mid t = 1, \cdots, T,\ i = 1, \cdots, N\}$

initiate $\alpha_1(j) = \pi_j b_j(\mathbf{x}_1)$ for all $j = 1, 2 \cdots, N$
for $t = 2, 3, \cdots, T$ **do**
  for $j = 1, 2, \cdots, N$ **do**
    $\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} b_j(\mathbf{x}_t)$
  end for
end for
initiate $\beta_T(j) = 1$ for all $j = 1, 2 \cdots, N$
for $t = T - 1, \cdots, 1$ **do**
  for $i = 1, 2, \cdots, N$ **do**
    $\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(\mathbf{x}_{t+1}) \beta_{t+1}(j)$
  end for
end for

$\forall t = 1, 2, \cdots, T$

$$p_{\mathbf{\Lambda}}(\mathbf{o}) = \sum_{i=1}^{N} \alpha_t(i)\beta_t(i)$$

e.g.

$$p_{\mathbf{\Lambda}}(\mathbf{o}) = \sum_{i=1}^{N} \alpha_T(i)$$

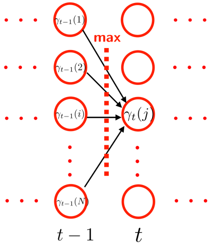
$$p_{\mathbf{\Lambda}}(\mathbf{o}) = \sum_{i=1}^{N} \alpha_1(i)\beta_1(i)$$

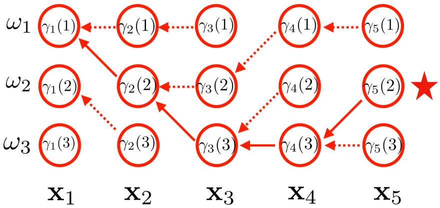
Decoding Problem

- recover the most probable state sequence $\mathbf{s}^*$ for any $\mathbf{o}$

$$\mathbf{s}^* = \arg\max_{\mathbf{s}\in\mathcal{S}} \; p_{\mathbf{\Lambda}}(\mathbf{o},\mathbf{s})$$

- Viterbi algorithm: dynamic programming to find $\mathbf{s}^*$ recursively





Decoding Problem: Viterbi Algorithm

Viterbi algorithm for HMMs

input: an HMM $\mathbf{\Lambda} = \{\Omega, \boldsymbol{\pi}, \mathbf{A}, \mathbb{B}\}$ and a sequence $\mathbf{o} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots \mathbf{x}_T\}$
output: Viterbi path $\mathbf{s}^*$ and $p_{\mathbf{\Lambda}}(\mathbf{o}, \mathbf{s}^*)$

initiate $\gamma_1(j) = \pi_j b_j(\mathbf{x}_1)$ for all $j = 1, 2 \cdots, N$
for $t = 2, 3, \cdots, T$ **do**
  for $j = 1, 2, \cdots, N$ **do**
    $\gamma_t(j) = \Big(\max_{i=1}^N \ \gamma_{t-1}(i) a_{ij} \Big) b_j(\mathbf{x}_t)$
    $\delta_t(j) = \arg\max_{i=1}^N \ \gamma_{t-1}(i) a_{ij}$
  end for
end for
termination: $p_{\mathbf{\Lambda}}(\mathbf{o}, \mathbf{s}^*) = \max_{i=1}^N \gamma_T(i)$
path backtracking: $\mathbf{s}^* = \{s_1^* \, s_2^* \cdots s_T^*\}$ with $s_T^* = \arg\max_{i=1}^N \gamma_T(i)$ and $s_{t-1}^* = \delta_t(s_t^*)$ for $t = T, \cdots, 2$

Training Problem

- how to estimate HMM parameters $\mathbf{\Lambda} = \{\boldsymbol{\pi}, \mathbf{A}, \mathbb{B}\}$
- collect a training set of variable-length sequences:

$$\mathcal{D} = \left\{\mathbf{o}^{(1)}, \mathbf{o}^{(2)}, \cdots, \mathbf{o}^{(R)}\right\}$$

where each $\mathbf{o}^{(r)} = \{\mathbf{x}_1^{(r)}, \mathbf{x}_2^{(r)}, \cdots \mathbf{x}_{T_r}^{(r)}\}$ denotes a sequence of T_r observations $(r = 1, 2 \cdots R)$

- maximum likelihood estimation:

$$\begin{aligned} \mathbf{\Lambda}^*_{\text{MLE}} &= \arg\max_{\mathbf{\Lambda}} \sum_{r=1}^{R} \ln p_{\mathbf{\Lambda}}\big(\mathbf{o}^{(r)}\big) \\ &= \arg\max_{\mathbf{\Lambda}} \sum_{r=1}^{R} \ln \sum_{\mathbf{s}^{(r)}} p_{\mathbf{\Lambda}}\big(\mathbf{o}^{(r)}, \mathbf{s}^{(r)}\big) \end{aligned}$$

- use EM algorithm: leading to the *Baum-Welch* algorithm

E-Step: Auxiliary Function $Q(\mathbf{\Lambda}|\mathbf{\Lambda}^{(n)})$ (I)

$$
\begin{aligned}
Q(\mathbf{\Lambda}|\mathbf{\Lambda}^{(n)}) &= \sum_{r=1}^{R} \mathbb{E}_{\mathbf{s}^{(r)}} \left[\ln p_{\mathbf{\Lambda}}(\mathbf{o}^{(r)}, \mathbf{s}^{(r)}) \,\middle|\, \mathbf{o}^{(r)}, \mathbf{\Lambda}^{(n)} \right] \\
&= \sum_{r=1}^{R} \sum_{\mathbf{s}^{(r)}} \ln p_{\mathbf{\Lambda}}(\mathbf{o}^{(r)}, \mathbf{s}^{(r)}) \Pr\left(\mathbf{s}^{(r)} \,\middle|\, \mathbf{o}^{(r)}, \mathbf{\Lambda}^{(n)}\right)
\end{aligned}
$$

where

$$
p_{\mathbf{\Lambda}}(\mathbf{o}^{(r)}, \mathbf{s}^{(r)}) = \pi(s_1^{(r)}) b(\mathbf{x}_1^{(r)}|s_1^{(r)}) \prod_{t=1}^{T_r-1} a(s_t^{(r)}, s_{t+1}^{(r)}) b(\mathbf{x}_{t+1}^{(r)}|s_{t+1}^{(r)})
$$

$$
\Pr\left(\mathbf{s}^{(r)} \,\middle|\, \mathbf{o}^{(r)}, \mathbf{\Lambda}^{(n)}\right) = \frac{p_{\mathbf{\Lambda}^{(n)}}\left(\mathbf{o}^{(r)}, \mathbf{s}^{(r)}\right)}{p_{\mathbf{\Lambda}^{(n)}}\left(\mathbf{o}^{(r)}\right)} = \frac{p_{\mathbf{\Lambda}^{(n)}}\left(\mathbf{o}^{(r)}, \mathbf{s}^{(r)}\right)}{\sum_{\mathbf{s}^{(r)}} p_{\mathbf{\Lambda}^{(n)}}\left(\mathbf{o}^{(r)}, \mathbf{s}^{(r)}\right)}
$$

E-Step: Auxiliary Function $Q(\mathbf{\Lambda}|\mathbf{\Lambda}^{(n)})$ (II)

$$Q(\mathbf{\Lambda}|\mathbf{\Lambda}^{(n)}) = \underbrace{\sum_{r=1}^{R}\sum_{i=1}^{N} \ln \pi_i \Pr(s_1^{(r)} = \omega_i \,|\, \mathbf{o}^{(r)}, \mathbf{\Lambda}^{(n)})}_{Q(\boldsymbol{\pi}|\boldsymbol{\pi}^{(n)})}$$

$$+ \underbrace{\sum_{r=1}^{R}\sum_{t=1}^{T_r-1}\sum_{i=1}^{N}\sum_{j=1}^{N} \ln a_{ij} \Pr(s_t^{(r)} = \omega_i, s_{t+1}^{(r)} = \omega_j \,|\, \mathbf{o}^{(r)}, \mathbf{\Lambda}^{(n)})}_{Q(\mathbf{A}|\mathbf{A}^{(n)})}$$

$$+ \underbrace{\sum_{r=1}^{R}\sum_{t=1}^{T_r}\sum_{i=1}^{N} \ln b_i(\mathbf{x}_t^{(r)}) \Pr(s_t^{(r)} = \omega_i \,|\, \mathbf{o}^{(r)}, \mathbf{\Lambda}^{(n)})}_{Q(\mathbb{B}|\mathbb{B}^{(n)})}$$

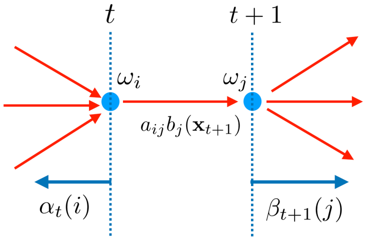
E-Step: Auxiliary Function $Q(\boldsymbol{\Lambda}|\boldsymbol{\Lambda}^{(n)})$ (III)

$$\eta_t^{(r)}(i,j) \triangleq \Pr\left(s_t^{(r)} = \omega_i, s_{t+1}^{(r)} = \omega_j \,\middle|\, \mathbf{o}^{(r)}, \boldsymbol{\Lambda}^{(n)}\right)$$

$$= \frac{\sum_{s_1^{(r)} \cdots s_{t-1}^{(r)} s_{t+2}^{(r)} \cdots s_{T_r}^{(r)}} p_{\boldsymbol{\Lambda}^{(n)}}\left(\mathbf{o}^{(r)}, s_1^{(r)}, \cdots s_{t-1}^{(r)}, \omega_i, \omega_j, s_{t+2}^{(r)} \cdots s_{T_r}^{(r)}\right)}{\sum_{s_1^{(r)} \cdots s_{T_r}^{(r)}} p_{\boldsymbol{\Lambda}^{(n)}}\left(\mathbf{o}^{(r)}, s_1^{(r)}, s_2^{(r)} \cdots s_{T_r}^{(r)}\right)}$$

$$\eta_t^{(r)}(i,j) = \frac{\alpha_t^{(r)}(i) a_{ij} b_j(\mathbf{x}_{t+1}) \beta_{t+1}^{(r)}(j)}{\sum_{i=1}^{N} \alpha_{T_r}^{(r)}(i)}$$

for all $1 \le t \le T_r$ and $1 \le i, j \le N$



E-Step: Auxiliary Function $Q(\mathbf{\Lambda}|\mathbf{\Lambda}^{(n)})$ (IV)

use $\eta_t^{(r)}(i,j)$ to re-write all auxiliary functions as:

$$Q\big(\boldsymbol{\pi}|\boldsymbol{\pi}^{(n)}\big) = \sum_{r=1}^{R}\sum_{i=1}^{N}\sum_{j=1}^{N} \ln \pi_i \cdot \eta_1^{(r)}(i,j)$$

$$Q\big(\mathbf{A}|\mathbf{A}^{(n)}\big) = \sum_{r=1}^{R}\sum_{t=1}^{T_r-1}\sum_{i=1}^{N}\sum_{j=1}^{N} \ln a_{ij} \cdot \eta_t^{(r)}(i,j)$$

$$Q\big(\mathbb{B}|\mathbb{B}^{(n)}\big) = \sum_{r=1}^{R}\sum_{t=1}^{T_r}\sum_{i=1}^{N}\sum_{j=1}^{N} \ln b_i(\mathbf{x}_t^{(r)}) \cdot \eta_t^{(r)}(i,j)$$

M-Step: $\boldsymbol{\pi}$ and $\mathbf{A}$

- initial probabilities $\boldsymbol{\pi}$:

$$\frac{\partial}{\partial \boldsymbol{\pi}}\Big(Q(\boldsymbol{\pi}|\boldsymbol{\pi}^{(n)}) + \lambda\big(\sum_{i=1}^{N}\pi_i - 1\big)\Big) = 0 \implies$$

$$\pi_i^{(n+1)} = \frac{\sum_{r=1}^{R}\sum_{j=1}^{N} \ \eta_1^{(r)}(i,j)}{\sum_{r=1}^{R}\sum_{i=1}^{N}\sum_{j=1}^{N} \ \eta_1^{(r)}(i,j)}$$

- transition probabilities $\mathbf{A}$: considering $\sum_j a_{ij} = 1$ for all i

$$a_{ij}^{(n+1)} = \frac{\sum_{r=1}^{R}\sum_{t=1}^{T_r-1} \ \eta_t^{(r)}(i,j)}{\sum_{r=1}^{R}\sum_{t=1}^{T_r-1}\sum_{j=1}^{N} \ \eta_t^{(r)}(i,j)}$$

M-Step: $\mathbb{B}$ for discrete HMMs

- $\mathbb{B}$ consists of all multinomial models in all HMM states $i = 1, 2, \cdots, N$:

$$\mathbb{B} = \left\{ b_{ik} \,\middle|\, 1 \le i \le N, 1 \le k \le K \right\}$$

- auxiliary function:

$$Q\left(\mathbb{B}|\mathbb{B}^{(n)}\right) = \sum_{r=1}^{R} \sum_{t=1}^{T_r} \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{K} \ln b_{ik} \cdot \delta(\mathbf{x}_t^{(r)} - v_k) \cdot \eta_t^{(r)}(i,j)$$

- updating formula:

$$b_{ik}^{(n+1)} = \frac{\sum_{r=1}^{R} \sum_{t=1}^{T_r} \sum_{j=1}^{N} \eta_t^{(r)}(i,j) \cdot \delta(\mathbf{x}_t^{(r)} - v_k)}{\sum_{r=1}^{R} \sum_{t=1}^{T_r} \sum_{j=1}^{N} \eta_t^{(r)}(i,j)}$$

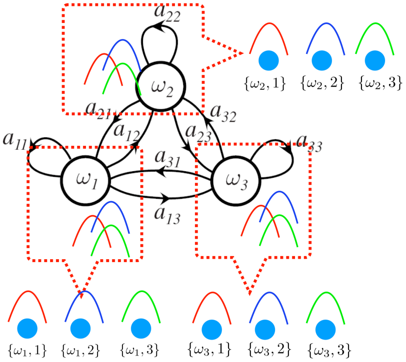
Gaussian Mixture Continuous Density HMMs (I)

- continuous HMMs: each state is associated with a p.d.f. of continuous observations
- use a GMM for each state

$$b_i(\mathbf{x}) = \sum_{m=1}^{M} w_{im} \cdot \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_{im}, \boldsymbol{\Sigma}_{im})$$

- $\mathbb{B}$ is composed of all GMM parameters:

$$\mathbb{B} = \Big\{ \boldsymbol{\mu}_{im}, \boldsymbol{\Sigma}_{im}, w_{im} \,\Big|\, 1 \le i \le N, \\ 1 \le m \le M \Big\}$$



Gaussian Mixture Continuous Density HMMs (II)

$$(1) \qquad Q(\mathbb{B}|\mathbb{B}^{(n)}) = \sum_{r=1}^{R}\sum_{t=1}^{T_r}\sum_{i=1}^{N}\sum_{m=1}^{M}\Big[\ln w_{im} + \ln \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_{im}, \boldsymbol{\Sigma}_{im})\Big] \Pr\left(s_t^{(r)} = \omega_i, l_t^{(r)} = m \,\middle|\, \mathbf{o}^{(r)}, \boldsymbol{\Lambda}^{(n)}\right)$$

$$(2) \qquad \Pr\left(s_t^{(r)} = \omega_i, l_t^{(r)} = m \,\middle|\, \mathbf{o}^{(r)}, \boldsymbol{\Lambda}^{(n)}\right) = \underbrace{\Pr\left(s_t^{(r)} = \omega_i \,\middle|\, \mathbf{o}^{(r)}, \boldsymbol{\Lambda}^{(n)}\right)}_{=\sum_{j=1}^{N}\eta_t^{(r)}(i,j)} \underbrace{\Pr\left(l_t^{(r)} = m \,\middle|\, s_t^{(r)} = \omega_i, \mathbf{o}^{(r)}, \boldsymbol{\Lambda}^{(n)}\right)}_{\triangleq\ \xi_t^{(r)}(i,m)}$$

$$(3) \qquad \xi_t^{(r)}(i,m) = \Pr(l_t^{(r)} = m \mid s_t^{(r)} = \omega_i, \mathbf{x}_t^{(r)}, \boldsymbol{\Lambda}^{(n)}) = \frac{w_{im}^{(n)}\mathcal{N}(\mathbf{x}_t^{(r)} \mid \boldsymbol{\mu}_{im}^{(n)}, \boldsymbol{\Sigma}_{im}^{(n)})}{\sum_{m=1}^{M} w_{m}^{(n)}\mathcal{N}(\mathbf{x}_t^{(r)} \mid \boldsymbol{\mu}_{im}^{(n)}, \boldsymbol{\Sigma}_{im}^{(n)})}$$

Gaussian Mixture Continuous Density HMMs (III)

After M-Step, we derive the updating formulas for all Gaussian mixture HMMs:

$$w_{im}^{(n+1)} = \frac{\sum_{r=1}^{R} \sum_{t=1}^{T_r} \sum_{j=1}^{N} \ \eta_t^{(r)}(i,j)\xi_t^{(r)}(i,m)}{\sum_{r=1}^{R} \sum_{t=1}^{T_r} \sum_{j=1}^{N} \sum_{m=1}^{M} \ \eta_t^{(r)}(i,j)\xi_t^{(r)}(i,m)}$$

$$\boldsymbol{\mu}_{im}^{(n+1)} = \frac{\sum_{r=1}^{R} \sum_{t=1}^{T_r} \sum_{j=1}^{N} \ \eta_t^{(r)}(i,j)\xi_t^{(r)}(i,m) \cdot \mathbf{x}_t^{(r)}}{\sum_{r=1}^{R} \sum_{t=1}^{T_r} \sum_{j=1}^{N} \ \eta_t^{(r)}(i,j)\xi_t^{(r)}(i,m)}$$

$$\boldsymbol{\Sigma}_{im}^{(n+1)} = \frac{\sum_{r=1}^{R} \sum_{t=1}^{T_r} \sum_{j=1}^{N} \ \eta_t^{(r)}(i,j)\xi_t^{(r)}(i,m)\left(\mathbf{x}_t^{(r)} - \boldsymbol{\mu}_{im}^{(n+1)}\right)\left(\mathbf{x}_t^{(r)} - \boldsymbol{\mu}_{im}^{(n+1)}\right)^{\intercal}}{\sum_{r=1}^{R} \sum_{t=1}^{T_r} \sum_{j=1}^{N} \ \eta_t^{(r)}(i,j)\xi_t^{(r)}(i,m)}$$

Training Problem: Baum-Welch Algorithm

Baum-Welch algorithm for HMMs

input: a training set $\{\mathbf{o}^{(r)} \mid r = 1, 2, \cdots, R\}$
output: HMM parameters $\boldsymbol{\Lambda} = \{\boldsymbol{\pi}, \mathbf{A}, \mathbb{B}\}$

initialize $\boldsymbol{\Lambda}^{(0)} = \{\boldsymbol{\pi}^{(0)}, \mathbf{A}^{(0)}, \mathbb{B}^{(0)}\}$; set $n = 0$
while not converged **do**
zero numerator/denominator accumulators for all parameters
for $r = 1, 2, \cdots, R$ **do**
1. forward-backward algorithm: $\{\mathbf{o}^{(r)}, \boldsymbol{\Lambda}^{(n)}\} \longrightarrow \{\alpha_t^{(r)}(i), \beta_t^{(r)}(i)\}$
2. $\{\alpha_t^{(r)}(i), \beta_t^{(r)}(i)\} \longrightarrow \{\eta_t^{(r)}(i,j), \xi_t^{(r)}(i,m)\}$
3. accumulate all numerator/denominator statistics

end for
update all parameters as the ratios of statistics $\longrightarrow \boldsymbol{\Lambda}^{(n+1)}$
$n = n + 1$
end while