Types of data: incomplete

\[ p(x|\theta) = \sum_{j=1}^{m} p_j p(x|\mu_j, \Sigma_j) \]

In incomplete data for a mixture model typically contain incomplete data involving only \( x \) samples; the assignment to components has to be inferred.

We have to estimate these models from incomplete data involving only \( x \) samples; the assignment to components has to be inferred.

• We can infer the values for the missing data based on the current setting of the parameters.

\begin{array}{c|c}
 x & y \\
 \hline
 x_1 & P(y = 1|x_1, \theta) \\
 x_2 & P(y = 2|x_2, \theta) \\
 \vdots & \vdots \\
 x_n & P(y = m|x_n, \theta) \\
\end{array}

The parameter estimation problem is again easy if we treat the inferred data as complete data. The solution has to be iterative, however.
The EM-algorithm

**Step 0:** specify the initial setting of the parameters \( \theta = \theta^{(0)} \)

\[
p(x | \theta) = \sum_{j=1}^{m} p_j p(x | \mu_j, \Sigma_j)
\]

For example, we could
- set each \( \mu_j \) to \( x \) sampled at random from the training set
- set each \( \Sigma_j \) to be the sample covariance of the whole data
- set mixing proportions \( p_j \) to be uniform \( p_j = 1/m \).

**The EM-algorithm**

**Step 0:** specify the initial setting of the parameters \( \theta = \theta^{(0)} \)

**E-step:** complete the incomplete data with the posterior probabilities

\[
P(y = j | x_i, \theta^{(k)}), \quad j = 1, \ldots, m, \quad i = 1, \ldots, n
\]

**M-step:** find the new setting of the parameters \( \theta^{(k+1)} \) by maximizing the log-likelihood of the completed (inferred) data

\[
\theta^{(k+1)} = \arg \max_{\theta} \sum_{i=1}^{n} \sum_{j=1}^{m} P(y = j | x_i, \theta^{(k)}) \log \left( p_j p(x_i | \mu_j, \Sigma_j) \right)
\]

**EM-algorithm: convergence**

\[
p(x | \theta) = \sum_{j=1}^{m} p_j p(x | \mu_j, \Sigma_j)
\]

- The EM-algorithm monotonically increases the log-likelihood of the training data. In other words,
  \[ l(\theta^{(0)}) < l(\theta^{(1)}) < l(\theta^{(2)}) < \ldots \] until convergence
  \[ l(\theta^{(k)}) = \sum_{i=1}^{n} \log p(x_i | \theta^{(k)}) \]
**EM-algorithm: auxiliary objective**

- We first introduce possible posterior assignments \(\{Q(j|i)\}\) and the corresponding auxiliary likelihood objective:

\[
l(\theta^{(k)}) = \sum_{i=1}^{n} \log p(x_i|\theta^{(k)}) = \sum_{i=1}^{n} \sum_{j=1}^{m} p_{j|i}(x_i|\mu_{j}^{(k)}, \Sigma_{j}^{(k)}) = \sum_{i=1}^{n} \sum_{j=1}^{m} Q(j|i) \frac{p_{j|i}(x_i|\mu_{j}^{(k)}, \Sigma_{j}^{(k)})}{Q(j|i)} \geq n \sum_{i=1}^{n} \sum_{j=1}^{m} Q(j|i) \log \frac{p_{j|i}(x_i|\mu_{j}^{(k)}, \Sigma_{j}^{(k)})}{Q(j|i)} = l(Q; \theta^{(k)})
\]

**Topics**

- Gaussian mixtures and the EM-algorithm
  - complete, incomplete, and inferred data
  - EM for mixtures
  - demo
  - selecting the number of mixture components
  - Gaussian mixtures for classification

**EM-algorithm: max-max and monotonicity**

- We can now rewrite the EM-algorithm in terms of two maximization steps involving the auxiliary objective:

  **E-step**: \(Q^{(k)} = \arg\max_{Q} l(Q; \theta^{(k)})\)

  **M-step**: \(\theta^{(k+1)} = \arg\max_{\theta} l(Q^{(k)}; \theta)\)

  The monotonic increase of the log-likelihood now follows from the facts that 1) the auxiliary objective is monotonically increasing, and 2) it equals the log-likelihood after each E-step

\[
l(\theta^{(k)}) = l(Q^{(k)}; \theta^{(k)}) \leq l(Q^{(k)}; \theta^{(k+1)}) \leq l(Q^{(k+1)}; \theta^{(k+1)}) = l(\theta^{(k+1)})
\]

**Regularized EM**

- Even a single covariance matrix in the Gaussian mixture model involves a number of parameters and can easily lead to over-fitting.

\[
p(x|\theta) = \sum_{j=1}^{m} p_{j} p(x|\mu_{j}, \Sigma_{j})
\]

- We can regularize the model by assigning a prior distribution over the parameters, especially the covariance matrices

**Regularized EM: prior**

- A Wishart prior over each covariance matrix is given by

\[
P(\Sigma|S, n') \propto \frac{1}{|S|^{n'/2}} \exp \left( -\frac{n'}{2} \text{Trace}(\Sigma^{-1} S) \right)
\]

  (written here in a bit non-standard way)

  \(S = \) “prior” covariance matrix

  \(n' = \) equivalent sample size

  The equivalent sample size represents the number of training samples we would have to see in order for the prior and the data to have equal effect on the solution
Regularized EM

- The E-step is unaffected (though the resulting values for the soft assignments will change)
- In the M-step we now maximize a penalized log-likelihood of the weighted training set:

\[
\sum_{i=1}^{n} \sum_{j=1}^{m} \frac{p(j|i)}{n_j + n'} \left( \sum_{i=1}^{n} \hat{p}(j|i) (x_i - \hat{\mu}_j)(x_i - \hat{\mu}_j)^T + n' S \right)
\]

Formally the regularization penalty changes the resulting covariance estimates only slightly:

\[
\Sigma_j^{(k+1)} \leftarrow \frac{1}{n_j + n'} \left( \sum_{i=1}^{n} \hat{p}(j|i) (x_i - \hat{\mu}_j)(x_i - \hat{\mu}_j)^T + n' S \right)
\]

Topics

- Gaussian mixtures and the EM-algorithm
  - complete, incomplete, and inferred data
  - EM for mixtures
  - demo
  - EM and convergence
  - regularized mixtures
  - selecting the number of mixture components
  - Gaussian mixtures for classification

Model selection and mixtures

- As a simple strategy for selecting the appropriate number of mixture components, we can find \( m \) that minimizes the overall description length (cf. BIC):

\[
DL \approx -\log p(\text{data}|\theta_m) + \frac{d_m n}{2} \log(n)
\]

- \( n \) is the number of training points,
- \( \theta_m \) are the maximum likelihood parameters for the \( m \)-component mixture, and
- \( d_m \) is the (effective) number of parameters in the \( m \)-component mixture.

Model selection: example

- Typical cases

\[
\begin{align*}
\text{m=1}, & \quad -\log P(\text{data})=2017.38, \text{penalty}=14.98, \ DL=2032.36 \\
\text{m=2}, & \quad -\log P(\text{data})=1712.69, \text{penalty}=32.95, \ DL=1745.65 \\
\text{m=3}, & \quad -\log P(\text{data})=1711.40, \text{penalty}=50.93, \ DL=1762.32 \\
\text{m=4}, & \quad -\log P(\text{data})=1682.06, \text{penalty}=68.90, \ DL=1750.97 \\
\end{align*}
\]

- Best cases (out of several runs):

\[
\begin{align*}
\text{m=1}, & \quad -\log P(\text{data})=2017.38, \text{penalty}=14.98, \ DL=2032.36 \\
\text{m=2}, & \quad -\log P(\text{data})=1712.69, \text{penalty}=32.95, \ DL=1745.65 \\
\text{m=3}, & \quad -\log P(\text{data})=1678.56, \text{penalty}=50.93, \ DL=1729.49 \\
\text{m=4}, & \quad -\log P(\text{data})=1649.08, \text{penalty}=68.90, \ DL=1717.98 \\
\end{align*}
\]
Classification example
- A digit recognition problem (8x8 binary digits)
  Training set $n = 100$ (50 examples of each digit).
  Test set $n = 400$ (200 examples of each digit).
- We’d like to estimate class conditional mixture models (and prior class frequencies) to solve the classification problem.

For example:
- Class 1: $P(y = 1)$, $p(x|\theta_1)$, (e.g., a 3-component mixture)
- Class 0: $P(y = 0)$, $p(x|\theta_0)$, (e.g., a 3-component mixture)

A new test example $x$ would be classified according to

$$\text{Class} = 1 \text{ if } \log \frac{P(y = 1)p(x|\theta_1)}{P(y = 0)p(x|\theta_0)} > 0$$

and Class $= 0$ otherwise.

Class labels $y = 0$ $y = 1$

Class conditional mixtures

$$p(x|\theta_j) = \sum_{j=1}^{20} p_j p(x|\mu_{j2}, \Sigma_{j2})$$

(a hierarchical mixture model)