Expectation-Maximization (EM) Algorithm

Adopted from slides by Alexander Ihler
Probabilistic models in unsupervised learning

• **K-means algorithm**
  • Assigned each example to exactly one cluster
  • What if clusters are overlapping?
    • Hard to tell which cluster is right
    • Maybe we should try to remain uncertain
  • Used Euclidean distance
  • What if cluster has a non-circular shape?

• **EM algorithm**
  • Assign data to cluster with some probability
  • Gives probability model of \( x \)! ("generative")
Expectation-Maximization (EM) Algorithm

• Learning algorithm for latent variable models
• Observed features \( x: x^{(1)}, x^{(2)}, \ldots, x^{(m)} \)
• Latent features \( z: z^{(1)}, z^{(2)}, \ldots, z^{(m)} \)

• Assume a probabilistic model over \( x, z \)
  \[
P_\theta(x, z) = P_\theta(x|z)P(z)
\]
• Learning most likely parameters \( \theta \) and \( z \) based on the observed data
  \[
  \arg \max_{\theta, z} P_\theta(x) = \sum_z P_\theta(x|z)P(z)
  \]
Expectation-Maximization (EM) Algorithm

• Iteratively update $\theta$ and $z$

• Initially assume random parameters $\theta$

• Iterate following two steps until convergence:
  • **Expectation (E-step):** Compute $P_{\theta}(z^{(i)}|x^{(i)})$ for each example $i$ based on the current parameters $\theta$
  • **Maximization (M-step):** Re-estimate the most likely parameters $\theta$ based on the current data $x, z$
Coin tossing example

- Two coins $A$ and $B$ with unknown biases $\theta_A$ and $\theta_B$
- Repeat following procedure 5 times:
  - Randomly choose one of the two coins
  - Perform 10 independent coin tosses with selected coin

\[ \hat{\theta}_A = \frac{\text{# of heads using coin } A}{\text{total # of flips using coin } A} \]
\[ \hat{\theta}_B = \frac{\text{# of heads using coin } B}{\text{total # of flips using coin } B} \]

<table>
<thead>
<tr>
<th>Coin A</th>
<th>Coin B</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 H, 5 T</td>
<td></td>
</tr>
<tr>
<td>9 H, 1 T</td>
<td>5 H, 5 T</td>
</tr>
<tr>
<td>8 H, 2 T</td>
<td>9 H, 11 T</td>
</tr>
<tr>
<td>4 H, 6 T</td>
<td></td>
</tr>
<tr>
<td>24 H, 6 T</td>
<td></td>
</tr>
</tbody>
</table>

5 sets, 10 tosses per set
EM Algorithm

• Observed feature $x \in \{0, 1, \cdots, 10\}$: # of heads
  • Observed data:
    $x^{(1)} = 5, x^{(2)} = 9, x^{(3)} = 8, x^{(4)} = 4, x^{(5)} = 7$

• Latent feature $z \in \{A, B\}$: identity of the coin
  • $z^{(1)}, z^{(2)}, z^{(3)}, z^{(4)}, z^{(5)}$
  • Assume $P(z) = 0.5$

• Model
  • Parameters $\theta_A, \theta_B$
  
  $P_\theta(x|z) = \begin{cases} 
  C_{10}^x \cdot (\theta_A)^x \cdot (1 - \theta_A)^{10-x} & \text{if } z = 'A' \\
  C_{10}^x \cdot (\theta_B)^x \cdot (1 - \theta_B)^{10-x} & \text{if } z = 'B' 
  \end{cases}$
EM Algorithm

$$P_\theta(z^{(i)}|x^{(i)}) = \frac{P(z^{(i)})P_\theta(x^{(i)}|z^{(i)})}{\sum_z P(z) P_\theta(x^{(i)}|z)}$$
EM Algorithm

\[ \hat{\theta}_A = \frac{\text{# of heads using coin } A}{\text{total # of flips using coin } A} \]

\[ \hat{\theta}_B = \frac{\text{# of heads using coin } B}{\text{total # of flips using coin } B} \]

<table>
<thead>
<tr>
<th>Coin A</th>
<th>Coin B</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \approx 2.2 ) H, 2.2 T</td>
<td>( \approx 2.8 ) H, 2.8 T</td>
</tr>
<tr>
<td>( \approx 7.2 ) H, 0.8 T</td>
<td>( \approx 1.8 ) H, 0.2 T</td>
</tr>
<tr>
<td>( \approx 5.9 ) H, 1.5 T</td>
<td>( \approx 2.1 ) H, 0.5 T</td>
</tr>
<tr>
<td>( \approx 1.4 ) H, 2.1 T</td>
<td>( \approx 2.6 ) H, 3.9 T</td>
</tr>
<tr>
<td>( \approx 4.5 ) H, 1.9 T</td>
<td>( \approx 2.5 ) H, 1.1 T</td>
</tr>
<tr>
<td>( \approx 21.3 ) H, 8.6 T</td>
<td>( \approx 11.7 ) H, 8.4 T</td>
</tr>
</tbody>
</table>

\[ \hat{\theta}_A^{(0)} = 0.60 \]
\[ \hat{\theta}_B^{(0)} = 0.50 \]

\[ \hat{\theta}_A^{(1)} \approx \frac{21.3}{21.3 + 8.6} \approx 0.71 \]
\[ \hat{\theta}_B^{(1)} \approx \frac{11.7}{11.7 + 8.4} \approx 0.58 \]

\[ \hat{\theta}_A^{(10)} \approx 0.80 \]
\[ \hat{\theta}_B^{(10)} \approx 0.52 \]
EM for Clustering: Mixtures of Gaussians

• Start with parameters describing each cluster
• Mean $\mu_c$, variance $\sigma_c$, “size” $\pi_c$
• Probability distribution:

$$p(x) = \sum_c \pi_c \mathcal{N}(x ; \mu_c, \sigma_c)$$
Mixtures of Gaussians

• Start with parameters describing each cluster
• Mean $\mu_c$, variance $\sigma_c$, “size” $\pi_c$
• Probability distribution:

$$p(x) = \sum_c \pi_c \mathcal{N}(x ; \mu_c, \sigma_c)$$

• Equivalent “latent variable” form:

$$p(z = c) = \pi_c$$

$$p(x|z = c) = \mathcal{N}(x ; \mu_c, \sigma_c)$$

Select a mixture component with probability $\pi$
Sample from that component’s Gaussian

“Latent assignment” $z$:
we observe $x$, but $z$ is hidden

$p(x) = \text{marginal over } x$
Multivariate Gaussian models

\[ \mathcal{N}(\mathbf{x} ; \mu, \Sigma) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{-1/2}} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu) \right\} \]

Maximum Likelihood estimates

\[ \hat{\mu} = \frac{1}{m} \sum_i x^{(i)} \]
\[ \hat{\Sigma} = \frac{1}{m} \sum_i (x^{(i)} - \hat{\mu})^T (x^{(i)} - \hat{\mu}) \]

We’ll model each cluster using one of these Gaussian “bells”…
EM Algorithm

• Observed feature \( x \in \mathbb{R}^d \)
• Latent feature \( z \in \{c_1, c_2, c_3\} \)
• Model
  • Parameters: Mean \( \mu_c \), variance \( \Sigma_c \), “size” \( \pi_c \) for each \( c \)

\[
P(x|z = c) = \mathcal{N}(x; \mu_c, \Sigma_c)
\]

\[
P(z = c) = \pi_c
\]

\[
P(x, z = c) = P(x|z = c)P(z = c) = \pi_c \cdot \mathcal{N}(x; \mu_c, \Sigma_c)
\]
EM Algorithm: E-step

- Start with clusters: Mean $\mu_c$, Covariance $\Sigma_c$, “size” $\pi_c$
- E-step (“Expectation”)
  - For each datum (example) $x_i$,
  - Compute “$r_{ic}$”, the probability that it belongs to cluster $c$
    - Compute its probability under model $c$
    - Normalize to sum to one (over clusters $c$)

\[
P(z^{(i)} \mid x^{(i)}) = r_{ic} = \frac{\pi_c N(x_i \mid \mu_c, \Sigma_c)}{\sum_{c'} \pi_{c'} N(x_i \mid \mu_{c'}, \Sigma_{c'})}
\]
EM Algorithm: E-step

- Start with clusters: Mean $\mu_c$, Covariance $\Sigma_c$, “size” $\pi_c$
- E-step (“Expectation”)
  - For each datum (example) $x_i$,
  - Compute “$r_{ic}$”, the probability that it belongs to cluster $c$
    - Compute its probability under model $c$
    - Normalize to sum to one (over clusters $c$)

  $$P(z^{(i)}|x^{(i)}) = r_{ic} = \frac{\pi_c N(x_i; \mu_c, \Sigma_c)}{\sum_{c'} \pi_{c'} N(x_i; \mu_{c'}, \Sigma_{c'})}$$

- If $x_i$ is very likely under the $c^{th}$ Gaussian, it gets high weight
- Denominator just makes r’s sum to one

$$r_1 \approx .33; \quad r_2 \approx .66$$
EM Algorithm: M-step

- Start with assignment probabilities $r_{ic}$
- Update parameters: Mean $\mu_c$, Covariance $\Sigma_c$, “size” $\pi_c$
- M-step (“Maximization”)
  - For each cluster (Gaussian) $z = c$,
  - Update its parameters using the (weighted) data points

$$m_c = \sum_i r_{ic} \quad \text{Total responsibility allocated to cluster c}$$

$$\pi_c = \frac{m_c}{m} \quad \text{Fraction of total assigned to cluster c}$$

$$\mu_c = \frac{1}{m_c} \sum_i r_{ic} x^{(i)} \quad \text{Weighted mean of assigned data}$$

$$\Sigma_c = \frac{1}{m_c} \sum_i r_{ic} (x^{(i)} - \mu_c)^T (x^{(i)} - \mu_c) \quad \text{Weighted covariance of assigned data (use new weighted means here)}$$
ANEMIA PATIENTS AND CONTROLS

Red Blood Cell Volume

Red Blood Cell Hemoglobin Concentration

From P. Smyth
ICML 2001
From P. Smyth
ICML 2001
EM ITERATION 5

From P. Smyth
ICML 2001
EM ITERATION 10

From P. Smyth
ICML 2001
EM ITERATION 15

From P. Smyth
ICML 2001
EM and missing data

• EM is a general framework for partially observed data
  • “Complete data” \( x_i, z_i \) – features and assignments
  • Assignments \( z_i \) are missing (unobserved)

• EM corresponds to
  • Computing the distribution over all \( z_i \) given the parameters
  • Maximizing the “expected complete” log likelihood
  • GMMs = plug in “soft assignments”, but not always so easy

• Alternatives: Stochastic EM, Hard EM
  • Instead of expectations, just sample the \( z_i \) or choose best (often easier)
  • Called “imputing” the values of \( z \)
  • Hard EM: similar to EM, but less “smooth”, more local minima
  • Stochastic EM: similar to EM, but with extra randomness
    • Not obvious when it has converged