2413, Machine Learning, Tutorial 8
Universität Bern

Simon Jenni (jenni@inf.unibe.ch)

K-means and Mixtures of Gaussians

1. Consider the following data points:
   $x^{(1)} = (1, 1)^T$,
   $x^{(2)} = (1, 3)^T$,
   $x^{(3)} = (7, 1)^T$,
   $x^{(4)} = (7, 3)^T$.

   - Apply the k-means clustering algorithm, when $k = 2$, and the initial centres are $c_1 = (10, 4)^T$ and $c_2 = (0, 2)^T$.

     Solution. Let us first compute the squared distances between the data points and cluster centers.

$dist^2$	$x^{(1)}$	$x^{(2)}$	$x^{(3)}$	$x^{(4)}$
c_1	90	82	18	10
c_2	2	2	50	50

     You can see that $x^{(1)}$ and $x^{(2)}$ are closer to c_2 and $x^{(3)}$ and $x^{(4)}$ are closer to c_1. The cluster assignment is therefore:

	$x^{(1)}$	$x^{(2)}$	$x^{(3)}$	$x^{(4)}$
$assign$	2	2	1	1

     The new cluster centres are $c_1 = (x^{(3)} + x^{(4)})/2 = (7, 2)^T$ and $c_2 = (x^{(1)} + x^{(2)})/2 = (1, 2)^T$. Let us iterate this one more time with the new cluster centres.

$dist^2$	$x^{(1)}$	$x^{(2)}$	$x^{(3)}$	$x^{(4)}$
c_1	37	37	1	1
c_2	1	1	37	37

	$x^{(1)}$	$x^{(2)}$	$x^{(3)}$	$x^{(4)}$
$assign$	2	2	1	1

     The cluster assignments of the data points are the same as in the previous iteration, therefore the algorithm has converged.

- Apply the k-means clustering algorithm with a different initialisation. The number of clusters is $k = 2$, and the initial centres are $c_1 = (4, 4)^T$ and $c_2 = (4, 0)^T$.

  Solution. The results of the first iteration:

$dist^2$	$x^{(1)}$	$x^{(2)}$	$x^{(3)}$	$x^{(4)}$
c_1	18	10	18	10
c_2	10	18	10	18

	$x^{(1)}$	$x^{(2)}$	$x^{(3)}$	$x^{(4)}$
$assign$	2	1	2	1

  $c_1 = (x^{(2)} + x^{(4)})/2 = (4, 3)^T$ and $c_2 = (x^{(1)} + x^{(3)})/2 = (4, 1)^T$

  The next iteration:

$dist^2$	$x^{(1)}$	$x^{(2)}$	$x^{(3)}$	$x^{(4)}$
c_1	13	9	13	9
c_2	9	13	9	13

	$x^{(1)}$	$x^{(2)}$	$x^{(3)}$	$x^{(4)}$
$assign$	2	1	2	1

  The algorithm has converged since the assignments did not change.

- Compare the results of the two runs of the k-means algorithm above.

  Solution. We started from two different initializations. In both cases the k-means algorithm converged. The two solutions correspond to two different local optima. The first solution has better cost, since $J_1 = \sum_{i=1}^{n} \|x^{(i)} - c(assign(x^{(i)}))\|^2 = (1+1+1+1)/2 = 2$ and $J_2 = (9+9+9+9)/2 = 18$.

2. Consider the following data points:
   $x^{(1)} = (1, 1)^T$,
   $x^{(2)} = (1, 3)^T$,
   $x^{(3)} = (2, 1)^T$,
   $x^{(4)} = (2, 3)^T$,
   $x^{(1)} = (7, 1)^T$,
   $x^{(2)} = (7, 3)^T$,
   $x^{(3)} = (8, 1)^T$,
   $x^{(4)} = (8, 3)^T$.

   We start with a hard cluster assignment of the data points, where $p(z^{(i)} = 1 \mid x^{(i)}; \phi, \mu, \Sigma) = 1$ for $i \in \{1, 2, 3, 4\}$ and $p(z^{(i)} = 2 \mid x^{(i)}; \phi, \mu, \Sigma) = 1$ for $i \in \{5, 6, 7, 8\}$. Apply the Expectation Maximisation (EM) algorithm to estimate the parameters of the Mixtures of Gaussians model.

   Solution. The initial cluster assignment matrix w is given by

   $w = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$, where $w_j^{(i)}$ represents the probability that $x^{(i)}$ belongs to the cluster j.

   Let us compute ϕ, μ and Σ (the maximisation step):

   $\phi_1 = \frac{1}{m} \sum_{i=1}^{m} w_1^{(i)} = (1+1+1+1+0+0+0+0)/8 = 0.5$, similarly $\phi_2 = 0.5$.

$$\mu_1 = \frac{\sum_{i=1}^{m} w_1^{(i)} \cdot x^{(i)}}{\sum_{i=1}^{m} w_1^{(i)}} = (x^{(1)} + x^{(2)} + x^{(3)} + x^{(4)})/4 = (1.5, 2)^T,$$
similarly $\mu_2 = (7.5, 2)^T$.

$$\Sigma_1 = \frac{\sum_{i=1}^{m} w_1^{(i)} \cdot (x^{(i)} - \mu_1)(x^{(i)} - \mu_1)^T}{\sum_{i=1}^{m} w_1^{(i)}} =$$
$$\frac{(-0.5, -1)^T(-0.5, -1) + \cdots + (0.5, 1)^T(0.5, 1)}{4} = \begin{pmatrix} 0.25 & 0 \\ 0 & 1 \end{pmatrix},$$
similarly $\Sigma_2 = \begin{pmatrix} 0.25 & 0 \\ 0 & 1 \end{pmatrix}$.

Now let us compute the expectation step, i.e. compute the updated w.

Using Bayes rule, we get:

$$w_1^{(i)} = p(z^{(i)} = j | x^{(i)}; \phi, \mu, \Sigma) = \frac{p(x^{(i)} | z^{(i)} = j; \mu, \Sigma) p(z^{(i)} = j; \phi)}{\sum_{j=1}^{k} p(x^{(i)} | z^{(i)} = j; \mu, \Sigma) p(z^{(i)} = j; \phi)}$$

According to the Mixtures of Gaussians model,

$p(z^{(i)} = j; \phi) = \phi_j$ and $p(x^{(i)} | z^{(i)} = j; \mu, \Sigma) = \frac{1}{(2\pi)^{n/2} |\Sigma_j|^{1/2}} \cdot exp(-\frac{1}{2}(x^{(i)} - \mu_j)^T \Sigma_j^{-1} (x^{(i)} - \mu_j))$.

$$(x^{(1)} - \mu_1)^T \Sigma_1^{-1} (x^{(1)} - \mu_1) = \begin{pmatrix} -0.5 & -1 \end{pmatrix} \begin{pmatrix} 4 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} -0.5 \\ -1 \end{pmatrix} = 2$$

$$(x^{(1)} - \mu_2)^T \Sigma_2^{-1} (x^{(1)} - \mu_2) = \begin{pmatrix} -6.5 & -1 \end{pmatrix} \begin{pmatrix} 4 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} -6.5 \\ -1 \end{pmatrix} = 170$$

From the above equations it follows that

$$w_1^{(1)} = \frac{0.5 \cdot exp(-1)}{0.5 \cdot exp(-1) + 0.5 \cdot exp(-85)} \approx 1, \text{ and}$$
$$w_2^{(1)} = \frac{0.5 \cdot exp(-85)}{0.5 \cdot exp(-1) + 0.5 \cdot exp(-85)} \approx 0,$$

because $exp(-85)$ is a very small number. We can calculate all the entries of w similarly,

$$w = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

We can see, that the new cluster label assignment w remained the same (within machine precision), therefore the algorithm converged.

3. There is a connection between K-means and the Mixtures of Gaussians model. You can modify the Maximization step of the later by setting $\Sigma = \epsilon \cdot I$, where I is the identity matrix. Prove that when $\epsilon \to 0$, the Expectation Maximization algorithm reduces to the K-means algorithm.

   Solution. Let us look at the Expectation step. By fixing $\Sigma_j = \epsilon \cdot I$, we get the following formula for $w_j^{(i)}$ below.

$$w_j^{(i)} = \frac{exp(-\frac{\|x^{(i)} - \mu_j\|^2}{2\epsilon})\phi_j}{\sum_{l=1}^{k} exp(-\frac{\|x^{(i)} - \mu_l\|^2}{2\epsilon})\phi_l} = \tag{1}$$

$$\frac{\phi_j}{\phi_j + \sum_{l \neq j} exp(-\frac{(\|x^{(i)} - \mu_l\|^2 - \|x^{(i)} - \mu_j\|^2)}{2\epsilon})\phi_l} \tag{2}$$

   As $\epsilon \to 0$, it is easy to see, that $w_j^{(i)} \to 1$ when μ_j is the closest center to $x^{(i)}$, and $w_j^{(i)} \to 0$ otherwise. This is exactly the k-means update rule for cluster label assignments. Similarly, when W is a hard cluster assignment (as in our case, when $\epsilon \to 0$), the formula for computing μ_j is exactly the same as the k-means update rule.

4. Let us denote the dimension of the training data with m, and let n be the number of data points. What happens to the Expectation Maximisation algorithm, when $m > n$?

   Solution. The maximum possible rank of the estimated covariance matrix is n. Because $m > n$ the covariance matrix becomes singular. The expectation step will fail because we can not compute the inverse of a singular covariance matrix.