1. Consider the following unlabeled training data:

	y_1	y_2
x_1	0	0
x_2	1	0
x_3	0	2
x_4	2	2

Consider also the following initialization centroids:

$$\mu_1 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \quad \mu_2 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

(a) Apply k-means until convergence. What are the final centroids?

(b) Plot the data points and draw the clusters (with their respective centroids).

(c) Compute the silhouette score for sample x_1, cluster c_1 and for the overall solution.

(d) As the ground truth, take x_3 as a "negative" sample and the rest as "positive". Compute the error classification rate (ECR) of k-means here against the ground truth.

For starters, it's probably worth it to write a general explanation of how k-means works. Basically, k-means is a clustering algorithm which, given a set of n data points $\{x_1, \dots, x_n\}$, tries to assign each data point to one of k clusters. Each cluster is centered around a centroid μ_i; while other clustering algorithms may create clusters of different shapes, k-means clusters are always circular/spherical. The algorithm works as follows:

(a) Initialize the centroids $\mu_1, \dots, \mu_n$.

(b) Assign each data point to the closest centroid.

(c) Recompute the centroids based on the assigned data points.

(d) Repeat steps b) and c) until convergence - that is, until the centroids don't change anymore (or until the change is below a certain threshold).

In k-means solution implementations, we usually perform a series of runs with different centroid initializations and pick the best one (where the *best* here is defined by a combination of metrics, which we'll discuss later). In the following exercises, though, we'll just use a single run. We'll also be using the Euclidian distance here, the most common distance metric used in this algorithm.

For starters, let's assign each sample to the closest centroid. We can do this by computing the distance between each sample and each centroid, and then picking the centroid with the smallest distance:

$$\|x_1 - \mu_1\| = \left\| \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 2 \\ 0 \end{bmatrix} \right\|^2 = 4, \quad \|x_1 - \mu_2\| = \left\| \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 2 \\ 1 \end{bmatrix} \right\|^2 = 5$$

$c = \text{argmin}_{k \in \{1,2\}} \|x_1 - \mu_n\|$ is, therefore, $k = 1$, and x_1 is assigned to cluster c_1.

Performing similar computations for the other samples, we get the following:

$$\begin{aligned}\operatorname{argmin}_{k\in\{1,2\}} \|x_2 - \mu_n\| &= \operatorname{argmin}_{k\in\{1,2\}}\{1, 2\} = c_1\\ \operatorname{argmin}_{k\in\{1,2\}} \|x_3 - \mu_n\| &= \operatorname{argmin}_{k\in\{1,2\}}\{8, 5\} = c_2\\ \operatorname{argmin}_{k\in\{1,2\}} \|x_4 - \mu_n\| &= \operatorname{argmin}_{k\in\{1,2\}}\{4, 1\} = c_2\end{aligned}$$

Now, we'll want to adjust our centroids: for each cluster, we'll compute the mean of all the samples assigned to it, and use that as the new centroid. k-means differs from other clustering algorithms here: EM, for example, utilizes every single sample in the dataset to compute the new centroids' parameters. k-means, on the other hand, by nature of working with hard assignments ends up using only a subset of the samples in the dataset to compute the new centroids.

$$\mu_1 = \frac{1}{2}(x_1 + x_2) = \frac{1}{2}\left(\begin{bmatrix}1\\0\\0\end{bmatrix} + \begin{bmatrix}8\\8\\4\end{bmatrix}\right) = \begin{bmatrix}0.5\\0\end{bmatrix}, \quad \mu_2 = \frac{1}{2}(x_3 + x_4) = \frac{1}{2}\left(\begin{bmatrix}3\\3\\0\end{bmatrix} + \begin{bmatrix}0\\0\\1\end{bmatrix}\right) = \begin{bmatrix}1\\2\end{bmatrix}$$

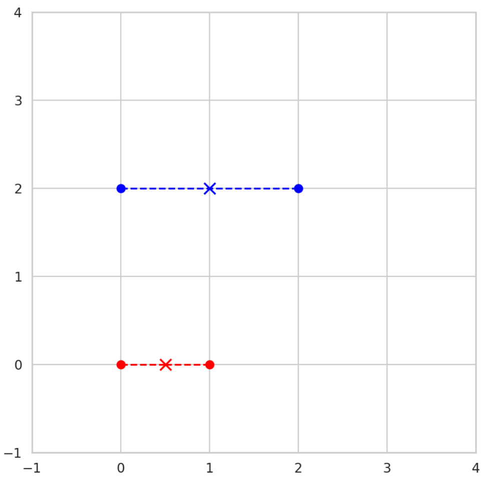
The centroids have moved, so we'll have to repeat steps b) and c).

$$\begin{aligned}\operatorname{argmin}_{k\in\{1,2\}} \|x_1 - \mu_n\| &= \operatorname{argmin}_{k\in\{1,2\}}\{0.25, 5\} = c_1\\ \operatorname{argmin}_{k\in\{1,2\}} \|x_2 - \mu_n\| &= \operatorname{argmin}_{k\in\{1,2\}}\{0.25, 4\} = c_1\\ \operatorname{argmin}_{k\in\{1,2\}} \|x_3 - \mu_n\| &= \operatorname{argmin}_{k\in\{1,2\}}\{4.25, 1\} = c_2\\ \operatorname{argmin}_{k\in\{1,2\}} \|x_4 - \mu_n\| &= \operatorname{argmin}_{k\in\{1,2\}}\{6.25, 1\} = c_2\end{aligned}$$

$$\mu_1 = \frac{1}{2}(x_1 + x_2) = \frac{1}{2}\left(\begin{bmatrix}1\\0\\0\end{bmatrix} + \begin{bmatrix}8\\8\\4\end{bmatrix}\right) = \begin{bmatrix}0.5\\0\end{bmatrix}, \quad \mu_2 = \frac{1}{2}(x_3 + x_4) = \frac{1}{2}\left(\begin{bmatrix}3\\3\\0\end{bmatrix} + \begin{bmatrix}0\\0\\1\end{bmatrix}\right) = \begin{bmatrix}1\\2\end{bmatrix}$$

The centroids haven't moved, so we're done. At the end of the algorithm's run, the cluster assignments are as follows: $c_1 = \{x_1, x_2\}$, $c_2 = \{x_3, x_4\}$.

Afterward, we can plot the clusters and the centroids (we can pretend that the cluster's circle is actually drawn and such), with c_1 in red and c_2 in blue:



Finally, regarding the **silhouette scores** of observations, we can compute them as follows (considering s_n to be the silhouette of sample x_n and c_n to be its assigned cluster):

$$
\begin{aligned}
s_n &= \frac{b_n - a_n}{\max\{a_n, b_n\}} \\
a_n &= \frac{1}{|c_n| - 1} \sum_{x_i \in c_n, x_i \neq x_n} \|x_i - x_n\| \\
b_n &= \min_{n' \in \{1, \cdots, k\}, n' \neq n} \frac{1}{|c_{n'}|} \sum_{x_i \in c_{k'}} \|x_i - x_n\|
\end{aligned}
$$

Essentially, the silhouette score of a sample will take into account how close it is to the other samples in its cluster (in average), and how far it is from the samples in its most neighboring cluster (in average).

$$
\begin{aligned}
a_1 &= \frac{1}{1} (\|x_2 - x_1\|) = 1, \quad & a_2 &= \frac{1}{1} (\|x_1 - x_2\|) = 1 \\
a_3 &= \frac{1}{1} (\|x_4 - x_3\|) = 2, \quad & a_4 &= \frac{1}{1} (\|x_3 - x_4\|) = 2
\end{aligned}
$$

$$
\begin{aligned}
b_1 &= \min_{n' \in \{1,2\}, n' \neq 1} \frac{1}{2} (\|x_3 - x_1\| + \|x_4 - x_1\|) = 2.414, \quad & b_2 &= \min_{n' \in \{1,2\}, n' \neq 1} \frac{1}{2} (\|x_3 - x_2\| + \|x_4 - x_2\|) = 2.236 \\
b_3 &= \min_{n' \in \{1,2\}, n' \neq 2} \frac{1}{2} (\|x_1 - x_3\| + \|x_2 - x_3\|) = 2.118, \quad & b_4 &= \min_{n' \in \{1,2\}, n' \neq 2} \frac{1}{2} (\|x_1 - x_4\| + \|x_2 - x_4\|) = 2.532
\end{aligned}
$$

$$
\begin{aligned}
s_1 &= \frac{2.414 - 1}{\max\{1, 2.414\}} = 0.5858, \quad & s_2 &= \frac{2.236 - 1}{\max\{1, 2.236\}} = 0.5528 \\
s_3 &= \frac{2.118 - 2}{\max\{2, 2.118\}} = 0.0557, \quad & s_4 &= \frac{2.532 - 2}{\max\{2, 2.532\}} = 0.2102
\end{aligned}
$$

The silhouette scores for each cluster are given by the average of the silhouette scores of its samples:

$$
s_1 = \frac{1}{2} (0.5858 + 0.5528) = 0.5693, \quad s_2 = \frac{1}{2} (0.0557 + 0.2102) = 0.1329
$$

The overall silhouette score is given by the average of the silhouette scores of each cluster:

$$
s = \frac{1}{2} (0.5693 + 0.1329) = 0.3511
$$

Finally, the **error classification rate** measures the proportion of misclassified samples in the dataset. If we think about the confusion matrix of the problem, we can write down the expression as:

$$
ECR = \frac{FP + FN}{FP + FN + TP + TN} = \frac{1 + 1}{1 + 1 + 2 + 2} = 0.5
$$

2. Consider the following unlabeled training data:

	y_1	y_2	y_3
x_1	1	0	0
x_2	8	8	4
x_3	3	3	0
x_4	0	0	1
x_5	0	1	0
x_6	3	2	1

and let the initial k centroids be the first k samples.

(a) Apply k-means until convergence, for $k \in \{2, 3\}$. What are the final centroids?

(b) Which k provides a better clustering regarding **cohesion** (i.e the intra-cluster distance - the sum of the distances from every point to their centroid)?

(c) Which k provides a better clustering regarding **separation** (i.e the inter-cluster distance - the average distance of every centroid to every other centroid)?

Won't be doing the first question since it's basically repeating the last question's exercise (twice), and it's also in the teacher's solutions. We'll need the post-convergence centroids, though, so I'll write them down here (plus each cluster's assigned samples):

$k = 2$:

$$\mu_1 = \begin{bmatrix} 1.4 \\ 1.2 \\ 0.4 \end{bmatrix}, \quad \mu_2 = \begin{bmatrix} 8 \\ 8 \\ 4 \end{bmatrix}$$

$$k_1 = \{x_1, x_3, x_4, x_5, x_6\}, \quad k_2 = \{x_2\}$$

$k = 3$:

$$\mu_1 = \begin{bmatrix} 0.333333 \\ 0.333333 \\ 0.333333 \end{bmatrix}, \quad \mu_2 = \begin{bmatrix} 8 \\ 8 \\ 4 \end{bmatrix}, \quad \mu_3 = \begin{bmatrix} 3 \\ 2.5 \\ 0.5 \end{bmatrix}$$

$$k_1 = \{x_1, x_4, x_5\}, \quad k_2 = \{x_2\}, \quad k_3 = \{x_3, x_6\}$$

Regarding **cohesion**, we measure it as the sum of the distances from every point to their centroid. We can write down the expression as:

$$\text{Cohesion}(k) = \sum_{i=1}^{k} \sum_{x \in k_i} \|x - \mu_i\|^2$$

$$\text{Cohesion}(2) = \|x_1 - \mu_1\|^2 + \|x_2 - \mu_2\|^2 + \|x_3 - \mu_1\|^2 + \|x_4 - \mu_1\|^2 + \|x_5 - \mu_1\|^2 + \|x_6 - \mu_1\|^2 \quad \approx 17.2$$

$$\text{Cohesion}(3) = \|x_1 - \mu_1\|^2 + \|x_2 - \mu_2\|^2 + \|x_3 - \mu_3\|^2 + \|x_4 - \mu_1\|^2 + \|x_5 - \mu_1\|^2 + \|x_6 - \mu_3\|^2 \quad \approx 3.0$$

Our goal is, ideally, to minimize this cohesion value: the closer the points are to their centroids, the better the clustering. We can see that $k = 3$ provides a better clustering in this regard, as expected: if there are more clusters, the points (generally speaking) should be able to better fit into them.

Separation, on the other hand, is the average distance of every centroid to every other centroid. We can write down the expression as:

$$\text{Separation}(k) = \frac{1}{k^2} \sum_{i=1}^{k} \sum_{j=1}^{k} \left\| \mu_i - \mu_j \right\|^2$$

$$\text{Separation}(2) = \frac{1}{4}(\|\mu_1 - \mu_1\|^2 + \|\mu_1 - \mu_2\|^2 + \|\mu_2 - \mu_1\|^2 + \|\mu_2 - \mu_2\|^2) = \frac{1}{2} \|\mu_1 - \mu_2\|^2 \approx 51.38$$
$$\text{Separation}(3) = \frac{1}{9}(\|\mu_1 - \mu_1\|^2 + \|\mu_1 - \mu_2\|^2 + \cdots \|\mu_3 - \mu_2\|^2 + \|\mu_3 - \mu_3\|^2) \approx 46.74$$

As we've seen above, our previous goal was to minimize the cohesion value (for a better clustering solution); intuitively, we can say that growing separation values should lead to a better clustering solution, as well: the more separated the centroids are, (ideally) the better the clustering. We can see that $k = 3$ provides a worse clustering solution in this regard: the centroids are closer to each other, which means that the points could be more likely to be assigned to the wrong cluster.

In the end, choosing the best k value is a matter of balancing these (and other) metrics.

3. Consider the following data points:

$$x_1 = (4), \quad x_2 = (0), \quad x_3 = (1)$$

Moreover, consider a mixture of two normal distributions
with the following initialization of likelihoods and priors:

$$\pi_1 = P(c = k_1) = 0.5, \qquad P(x|c = k_1) = \mathcal{N}(x; \mu_1, \sigma_1^2) = \mathcal{N}(x; 1, 1)$$
$$\pi_2 = P(c = k_2) = 0.5, \qquad P(x|c = k_2) = \mathcal{N}(x; \mu_2, \sigma_2^2) = \mathcal{N}(x; 0, 1)$$

Plot the clusters after a single iteration of the EM algorithm.

The Expectation-Maximization (EM) algorithm is a powerful tool for cluster creation, learning the parameters of a distribution/mixture of distributions. It is a two-step iterative algorithm that alternates between two steps: the **Expectation** step and the **Maximization** step.

Here, our centroids are precisely the means of each distribution; if in k-means we directly assigned the points to a cluster (and centroid updates would only take into account the samples assigned to them), in EM points aren't "assigned" to a given cluster, having rather a probability of belonging to each one of them. This way, parameter updates will always take into account **all** the points in the dataset (which should, ideally, lead us to a better clustering solution). This way, with each centroid being associated with a given distribution, we're not only able to know where the cluster's centroid is, but also its shape.

In the **E-step**, we calculate the **posteriors** (i.e the probability/*expectation* of a point belonging to each cluster), given the current parameters of the distributions. As we know, these probabilities can be written in function of the likelihoods and priors:

$$\gamma_{ni} = P(c = k_i|x_n) = \frac{P(x_n|c = k_i)\pi_i}{\sum_{j=1}^{k} P(x|c = k_j)\pi_j}$$

Since we know the priors in advance, we'll only need to calculate the likelihoods here. From the question's statement, we know that we can write down the likelihoods as:

$$P(x|c = k_i) = \mathcal{N}(x; \mu_i, \sigma_i^2)$$

Computing this for each sample (and, for each one of them, for each cluster), we'll be able to gather the following (calculations will be shown in their entirety for the first sample, only final results for the remaining ones):

$$\begin{aligned}
\mathcal{N}(x_1; \mu_1, \sigma_1^2) &= \frac{1}{\sqrt{2\pi\sigma_1^2}} \exp\left(-\frac{(x_1 - \mu_1)^2}{2\sigma_1^2}\right) \\
&= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x_1 - 1)^2}{2}\right) = 0.004432 \\
\mathcal{N}(x_1; \mu_2, \sigma_2^2) &= \frac{1}{\sqrt{2\pi\sigma_2^2}} \exp\left(-\frac{(x_1 - \mu_2)^2}{2\sigma_2^2}\right) \\
&= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x_1 - 0)^2}{2}\right) = 0.0001338
\end{aligned}$$

$$\gamma_{11} = \frac{0.004432 \cdot 0.5}{0.004432 \cdot 0.5 + 0.242 \cdot 0.5} = 0.9707, \quad \gamma_{12} = \frac{0.242 \cdot 0.5}{0.004432 \cdot 0.5 + 0.242 \cdot 0.5} = 0.02931$$

Performing similar computations for the following samples, we're able to gather the following results:

$$\mathcal{N}(x_2; \mu_1, \sigma_1^2) = 0.242, \quad \mathcal{N}(x_2; \mu_2, \sigma_2^2) \;= 0.3989, \quad \mathcal{N}(x_3; \mu_1, \sigma_1^2) = 0.3989, \quad \mathcal{N}(x_3; \mu_2, \sigma_2^2) \;= 0.242$$

$$\gamma_{21} = 0.3775, \quad \gamma_{22} = 0.6225, \quad \gamma_{31} \;= 0.6225, \quad \gamma_{32} = 0.3775$$

Having calculated the posteriors (and, subsequently, being able to perform an *estimate of the probabilities of a sample belonging to each cluster*), we can move on to the **M-step**, where we'll update the parameters of the distributions, utilizing these new estimations. In this case, we'll be updating the means and std. deviations of each distribution, plus updating the priors afterward (note that when updating σ, we use the newly updated means).

$$\mu_k = \frac{\sum_{n=1}^{N} \gamma_{nk} x_n}{\sum_{n=1}^{N} \gamma_{nk}}, \quad \sigma_k = \sqrt{\frac{\sum_{n=1}^{N} \gamma_{nk} (x_n - \mu_k)^2}{\sum_{n=1}^{N} \gamma_{nk}}}, \quad \pi_k = \frac{1}{N} \sum_{n=1}^{N} \gamma_{nk}$$

k_1:

$$\mu_1 = \frac{0.9707 \cdot 4 + 0 + 0.6225 \cdot 1}{0.9707 + 0.3775 + 0.6225} = 2.286$$
$$\sigma_1 = 1.724,$$
$$\pi_1 = 0.6569$$

k_2:

$$\mu_2 = \frac{0.02931 \cdot 4 + 0 + 0.3775 \cdot 1}{0.02931 + 0.6225 + 0.3775} = 0.4807$$
$$\sigma_2 = 0.769,$$
$$\pi_2 = 0.3431$$

After one EM epoch, we'll have the following:

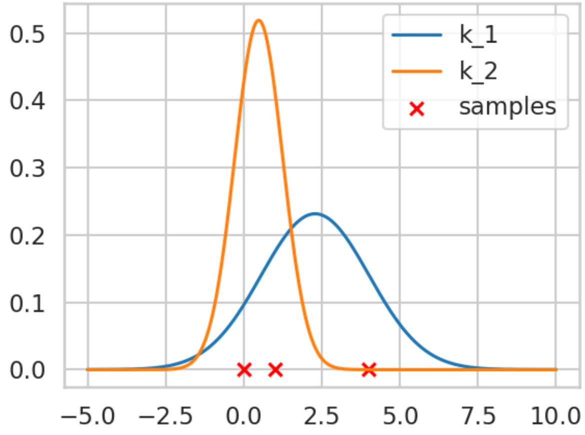


Figure 1: EM's cluster distributions after epoch 1