STAT 154: Homework 3

Release date: **Thursday, February 21**

Due by: **11 PM, Wednesday, Mar 6**

This homework follows regular submission format, i.e., it is to be submitted by each student in the class *individually*–no teams!

1 A closer look at EM (25 points)

In this question, we consider a simple mixture model and work our way through a derivation of the EM updates.[1] *While the question looks long, please be patient in reading it–the description itself will help you understand EM better and also different parts of the problem only appear long because of the detailed explanation.*

We work with the following simple two mixture model:

$$\begin{aligned} Z &\sim \text{Bernoulli}(1-w)+1 \\ X|Z=1 &\sim \mathcal{N}(\mu_1,1), \quad \text{and} \\ X|Z=2 &\sim \mathcal{N}(\mu_2,1), \end{aligned} \tag{1}$$

where Z denotes the label of the Gaussian from which X is drawn. Given a set of observations only for X (i.e., the labels are unobserved), our goal is to infer the maximum-likelihood parameters for μ_1, μ_2 and w. Note that to simplify your calculations, we have fixed the variance parameter and assumed it to be known.

(a) (3 points) Let $\theta = (\mu_1, \mu_2, w)$ denote the parameters of the model. **Write down the expressions of the joint likelihood $p(X = x, Z = 1; \theta)$ and $p(X = x, Z = 2; \theta)$. What is the marginal likelihood $p(X = x; \theta)$ and the log-likelihood $\ell(X = x; \theta)$? Given n i.i.d. samples $\{x_1, \ldots, x_n\}$, write the expression for the log-likelihood $\ell(X_1 = x_1, \ldots, X_n = x_n; \theta)$.**
Ans. We derive the general expression with variance parameters as

$$p(X=x,Z=1;\theta) = p(Z=1;\theta)p(X=x|Z=1;\theta) = w\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{(x-\mu_1)^2}{2}\right)$$
$$p(X=x,Z=2;\theta) = p(Z=2;\theta)p(X=x|Z=2;\theta) = (1-w)\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{(x-\mu_2)^2}{2}\right)$$

[1] The question borrows ideas from one of the homeworks of machine learning class CS 189, Spring 2018.

and hence the total likelihood $p(X = x; \theta)$ is given by

$$\begin{aligned} p(X = x; \theta) &= p(X = x, Z = 1; \theta) + p(X = x, Z = 2; \theta) \\ &= w \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x-\mu_1)^2}{2}\right) + (1-w)\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x-\mu_2)^2}{2}\right) \end{aligned}$$

and the log-likelihood $\ell(X = x; \theta)$ is thereby given by

$$\ell(X = x; \theta) = \log\left[\frac{w}{\sqrt{2\pi}} \exp\left(-\frac{(x-\mu_1)^2}{2}\right) + \frac{1-w}{\sqrt{2\pi}} \exp\left(-\frac{(x-\mu_2)^2}{2}\right)\right].$$

The log-likelihood for a dataset with i.i.d. points is just a sum of the likelihoods for individual points in the dataset and hence we have

$$\begin{aligned} &\ell(X_1 = x_1, \ldots, X_n = x_n; \theta) \\ &\sum_{i=1}^{n} \log\left[\frac{w}{\sqrt{2\pi}} \exp\left(-\frac{(x_i-\mu_1)^2}{2}\right) + \frac{1-w}{\sqrt{2\pi}} \exp\left(-\frac{(x_i-\mu_2)^2}{2}\right)\right] \end{aligned}$$

Note that finding MLE by differentiating this likelihood is not trivial (try differentiating with respect to μ_1 and setting it to zero!) and hence we feel the need of an algorithm like EM.

(b) (4 points) To simplify notation, from now on, we use the notation

$$\ell(x; \theta) = \ell(X = x; \theta), \quad \text{and} \quad p(x, k; \theta) = p(X = x, Z = k; \theta).$$

Let q denote a distribution on the (hidden) labels $\{Z_i\}_{i=1}^n$ given by

$$q(Z_1 = z_1, \ldots, Z_n = z_n) = \prod_{i=1}^{n} q_i(Z_i = z_i). \tag{2}$$

Note that since $Z \in \{1, 2\}$, q has n parameters, namely $\{q_i(Z_i = 1), i = 1, \ldots, n\}$. **Show that for a given point x_i, we have**

$$\ell(x_i; \theta) \geq \mathcal{F}_i(\theta; q_i) := \underbrace{\sum_{k=1}^{2} q_i(k) \log p(x_i, k; \theta)}_{\mathcal{L}(x_i; \theta, q_i)} + \underbrace{\sum_{k=1}^{2} q_i(k) \log\left(\frac{1}{q_i(k)}\right)}_{H(q_i)}, \tag{3}$$

where $H(q_i)$ denotes the Shannon-entropy of the distribution q_i. Thus **conclude that we obtain the following lower bound on the log-likelihood:**

$$\ell(\{x_i\}_{i=1}^n; \theta) \geq \mathcal{F}(\theta; q) := \sum_{i=1}^{n} \mathcal{F}_i(\theta; q_i). \tag{4}$$

Hint: Jensen's inequality, the concave-$\cap$ nature of the log, and reviewing lecture notes might be useful.

Ans. We prove this result in generality, without using the particular expressions for the Gaussian distribution.

$$\begin{aligned}\ell(x_i;\theta) = \log p(x_i;\theta) = \log \sum_{k=1}^{2} p(x_i,k;\theta) &= \log \sum_{k=1}^{2} q_i(k)\frac{p(x_i,k;\theta)}{q_i(k)} \\ &= \log \mathbb{E}_{Z\sim q_i}\left[\frac{p(x_i,Z;\theta)}{q_i(Z)}\right] \\ &\overset{(i)}{\geq} \mathbb{E}_{Z\sim q_i}\left[\log \frac{p(x_i,Z;\theta)}{q_i(Z)}\right] \\ &= \sum_{k=1}^{2} q_i(k)\log p(x_i,k;\theta) + \sum_{k=1}^{2} q_i(k)\log\frac{1}{q_i(k)}.\end{aligned}$$

where step (i) follows from the Jensen's inequality: For any concave function f, we have $f(\mathbb{E}[X]) \geq \mathbb{E}(f(X))$.

(c) (2 points) The EM algorithm can be considered a coordinate-ascent[2] algorithm on the lower bound $\mathcal{F}(\theta;q)$ derived in the previous part, where we ascend with respect to θ and q in an alternating fashion. More precisely, one iteration of the EM algorithm is made up of 2-steps:

$$\begin{aligned} q^{t+1} &= \arg\max_{q} \mathcal{F}(\theta^t;q) && \text{(E-step)} \\ \theta^{t+1} &\in \arg\max_{\theta} \mathcal{F}(\theta;q^{t+1}). && \text{(M-step)}\end{aligned}$$

Given an estimate θ^t, the previous part tells us that $\ell(\{x_i\}_{i=1}^n;\theta^t) \geq \mathcal{F}(\theta^t;q)$. **Verify that equality holds in this bound if we plug in $q(Z_1 = z_1,\ldots,Z_n = z_n) = \prod_{i=1}^n p(Z = z_i|X = x_i;\theta^t)$ and hence we can conclude that**

$$q^{t+1}(Z_1 = z_1,\ldots,Z_n = z_n) = \prod_{i=1}^{n} p(Z = z_i|X = x_i;\theta^t). \tag{5}$$

is a valid maximizer for the problem $\max_q \mathcal{F}(\theta^t;q)$ and hence a valid E-step update.

Ans. We have

$$\mathcal{F}(\theta^t;q) = \sum_{i=1}^{n}\sum_{k=1}^{2} q_i(k)\log\frac{p(x_i,k;\theta^t)}{q_i(k)}.$$

Substituting $q_i(k) = p(Z_i = k|X = x_i;\theta^t)$ (short-form $p(k|x_i;\theta^t)$, we obtain that the RHS is

$$\sum_{i=1}^{n}\sum_{k=1}^{2} q_i(k)\log\frac{p(x_i,k;\theta^t)}{q_i(k)} = \sum_{i=1}^{n}\sum_{k=1}^{2} p(k|x_i;\theta^t)\log\frac{p(x_i,k;\theta^t)}{p(k|x_i;\theta^t)}.$$

[2] A coordinate-ascent algorithm is just one that fixes some coordinates and maximizes the function with respect to the others as a way of taking iterative improvement steps.

Note that we have $p(x_i, k; \theta^t) = p(x_i; \theta^t)p(k|x_i; \theta^t)$. Also since $p(\cdot|x_i; \theta^t)$ is valid conditional distribution on Z, we have that $\sum_{k=1}^{k} p(\cdot|x_i; \theta^t) = 1$. Using these two facts, we obtain that

$$\begin{aligned}
\sum_{i=1}^{n}\sum_{k=1}^{2} q_i(k) \log \frac{p(x_i, k; \theta^t)}{q_i(k)} &= \sum_{i=1}^{n}\sum_{k=1}^{2} p(k|x_i; \theta^t) \log \frac{p(x_i, k; \theta^t)}{p(k|x_i; \theta^t)} \\
&= \sum_{i=1}^{n}\sum_{k=1}^{2} p(k|x_i; \theta^t) \log \frac{p(x_i)p(k|x_i; \theta^t)}{p(k|x_i; \theta^t)} \\
&= \sum_{i=1}^{n}\sum_{k=1}^{2} p(k|x_i; \theta^t) \log p(x_i) \\
&= \sum_{i=1}^{n} \log p(x_i) \underbrace{\sum_{k=1}^{2} p(k|x_i; \theta^t)}_{=1} \\
&= \sum_{i=1}^{n} \log p(x_i) = \ell(\{x_i\}_{i=1}^{n}; \theta^t).
\end{aligned}$$

Using this equality and the fact that $\ell(\{x_i\}_{i=1}^{n}; \theta^t) \geq \mathcal{F}(\theta^t; q)$ for any q, we conclude that

$$q^{t+1}(Z_1 = z_1, \ldots, Z_n = z_n) = \prod_{i=1}^{n} p(Z = z_i|X = x_i; \theta^t).$$

is a valid maximizer for the problem $\max_q \mathcal{F}(\theta^t; q)$ and hence a valid E-step update.

(d) (2 points) **Derive the expressions for $p(Z = 1|X = x_i; \theta^t)$ and $p(Z = 2|X = x_i; \theta^t)$ to complete the E-step computations where $\theta^t = (\mu_1^t, \mu_2^t, w^t)$.**
Ans. We have

$$\begin{aligned}
q_i^{t+1}(Z_i = 1) = p(Z = 1|X = x_i; \theta^t) &= \frac{p(Z = 1, X = x_i; \theta^t)}{p(X = x_i; \theta^t)} \\
&= \frac{p(Z = 1, X = x_i; \theta^t)}{p(Z = 1, X = x_i; \theta^t) + p(Z = 2, X = x_i; \theta^t)} \\
&= \frac{w^t \exp\left(-\frac{(x_i - \mu_1^t)^2}{2}\right)}{w^t \exp\left(-\frac{(x_i - \mu_1^t)^2}{2}\right) + (1 - w^t) \exp\left(-\frac{(x_i - \mu_2^t)^2}{2}\right)}
\end{aligned}$$

Since Z only takes two values, we can directly compute

$$\begin{aligned}
q_i^{t+1}(Z_i = 2) = p(Z = 2|X = x_i; \theta^t) &= 1 - p(Z = 1|X = x_i; \theta^t) \\
&= \frac{(1 - w^t) \exp\left(-\frac{(x_i - \mu_2^t)^2}{2}\right)}{w^t \exp\left(-\frac{(x_i - \mu_1^t)^2}{2}\right) + (1 - w^t) \exp\left(-\frac{(x_i - \mu_2^t)^2}{2}\right)}
\end{aligned}$$

(e) (3 points) We now discuss the M-step. Using the definitions from equations (3) and (4), we have that

$$\mathcal{F}(\theta; q^{t+1}) = \sum_{i=1}^{n} (\mathcal{L}(x_i; \theta, q_i^{t+1}) + H(q_i)) = H(q^{t+1}) + \sum_{i=1}^{n} \mathcal{L}(\mathbf{x}_i; \theta, q_i^{t+1}),$$

where we have used the fact that entropy in this case is given by $H(q^{t+1}) = \sum_{i=1}^{n} H(q_i^{t+1})$. Notice that although (as computed in previous part), q^{t+1} depends on θ^t, the M-step only involves maximizing $\mathcal{F}(\theta; q^{t+1})$ with respect to just the parameter θ while keeping the parameter q^{t+1} fixed. Now, noting that the entropy term $H(q^{t+1})$ does not depend on the parameter θ, we conclude that the M-step simplifies to solving for

$$\arg\max_{\theta} \underbrace{\sum_{i=1}^{n} \mathcal{L}(\mathbf{x}_i; \theta, q_i^{t+1})}_{=:\mathcal{L}(\theta; q^{t+1})}.$$

We use the simplified notation

$$q_i^{t+1} := q_i^{t+1}(Z_i = 1) \quad \text{and} \quad 1 - q_i^{t+1} := q_i^{t+1}(Z_i = 2)$$

and recall that $\theta = (\mu_1, \mu_2, w)$. **Show that the expression for $\mathcal{L}(\theta; q^{t+1})$ for the 2-mixture case is given by**

$$\mathcal{L}((\mu_1, \mu_2, w); q^{t+1}) = C + \sum_{i=1}^{n} \left[q_i^{t+1} \left(\log w - \frac{(x_i - \mu_1)^2}{2} \right) + (1 - q_i^{t+1}) \left(\log(1 - w) - \frac{(x_i - \mu_2)^2}{2} \right) \right],$$

where C is a constant that does not depend on θ or q^{t+1}.

Ans. Using the shorthand $p(x_i, k) = p(X = x_i, Z = k)$, we have

$$\begin{aligned}
&\mathcal{L}((\mu_1, \mu_2, w); q^{t+1}) \\
&= \sum_{i=1}^{n} (q_i^{t+1} \log p(x_i, 1) + (1 - q_i^{t+1}) \log p(x_i, 2)) \\
&= \sum_{i=1}^{n} (q_i^{t+1} \log(w p(x_i | Z_i = 1)) + (1 - q_i^{t+1}) \log ((1 - w) p(x_i | Z_i = 2)) \\
&= \sum_{i=1}^{n} (q_i^{t+1} \log \left(w \frac{1}{\sqrt{2\pi}} e^{-\frac{(x_i - \mu_1)^2}{2}} \right) + (1 - q_i^{t+1}) \log \left((1 - w) \frac{1}{\sqrt{2\pi}} e^{-\frac{(x_i - \mu_2)^2}{2}} \right) \\
&= \sum_{i=1}^{n} \left[q_i^{t+1} \left(\log w - \frac{(x_i - \mu_1)^2}{2} - \frac{1}{2} \log 2\pi \right) \right. \\
&\qquad \left. + (1 - q_i^{t+1}) \left(\log(1 - w) - \frac{(x_i - \mu_2)^2}{2\sigma_2^2} + \frac{1}{2} \log 2\pi \right) \right] \\
&= C + \sum_{i=1}^{n} \left[q_i^{t+1} \left(\log w - \frac{(x_i - \mu_1)^2}{2} \right) + (1 - q_i^{t+1}) \left(\log(1 - w) - \frac{(x_i - \mu_2)^2}{2} \right) \right].
\end{aligned}$$

(f) (6 points) Using the expression of $\mathcal{L}$ from the previous part, **derive the expressions for the gradients of $\mathcal{L}(\theta; q^{t+1})$ with respect to μ_1, μ_2, w. By setting these gradients to zero, show that the M-step updates are given by**

$$\mu_1^{t+1} = \frac{\sum_{i=1}^n q_i^{t+1} x_i}{\sum_{i=1}^n q_i^{t+1}}, \quad \mu_2^{t+1} = \frac{\sum_{i=1}^n (1 - q_i^{t+1}) x_i}{\sum_{i=1}^n (1 - q_i^{t+1})}, \quad \text{and} \quad w^{t+1} = \frac{\sum_{i=1}^n q_i^{t+1}}{n}.$$

This complete the derivation of EM updates in the simpler model introduced in the beginning of the problem.

Ans.

$$\frac{\partial \mathcal{L}}{\partial \mu_1} = -\sum_{i=1}^n q_i^{t+1}(\mu_1 - x_i), \qquad \frac{\partial \mathcal{L}}{\partial \mu_2} = -\sum_{i=1}^n (1 - q_i^{t+1})(\mu_2 - x_i),$$

$$\frac{\partial \mathcal{L}}{\partial w} = \sum_{i=1}^n \frac{q_i^{t+1}}{w} - \sum_{i=1}^n \frac{1 - q_i^{t+1}}{1 - w}.$$

By setting them to zero, we obtain the given updates.

(g) (5 points) We now see EM in action and compare it with K-means. (No code is required in the submission.) Generate 1000 samples from the following mixture of two Gaussians in two dimensions:

$$\begin{aligned} Z &= \text{Bernoulli}(0.5) + 1 \\ X|Z = 1 &\sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \\ X|Z = 2 &\sim \mathcal{N}\left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix} \right) \end{aligned} \tag{6}$$

where I_2 denotes the identity matrix in two dimensions. **Generate both (Z, X) for the data and scatter plot the X values with color based on Z. Run K-means on X with $K = 2$ and report the cluster centers and scatter plot the data with estimated labels. Run EM on X to fit two clusters too (use Mclust package with $G = 2$) and report the mean parameters and scatter plot the data with estimated labels. Note that we need 3 plots for this part. Justify qualitatively why the cluster centers obtained by K-means and EM and the estimated label are different.**

Ans. We write couple of functions to generate the data, and then run the two clustering algorithms on it and observe the difference in output. We see that in this case (and you are free to try several other cases) EM seems to perform better than K-means. In fact, The probabilistic model for EM allows us to model the difference in mixture weights (different sizes of the clusters), different variances in the clusters. On the other hand, the vanilla K-means suffers from poor clustering if the mixtures are not well-separated, or have different scalings/variances in different clusters. Try drawing a few

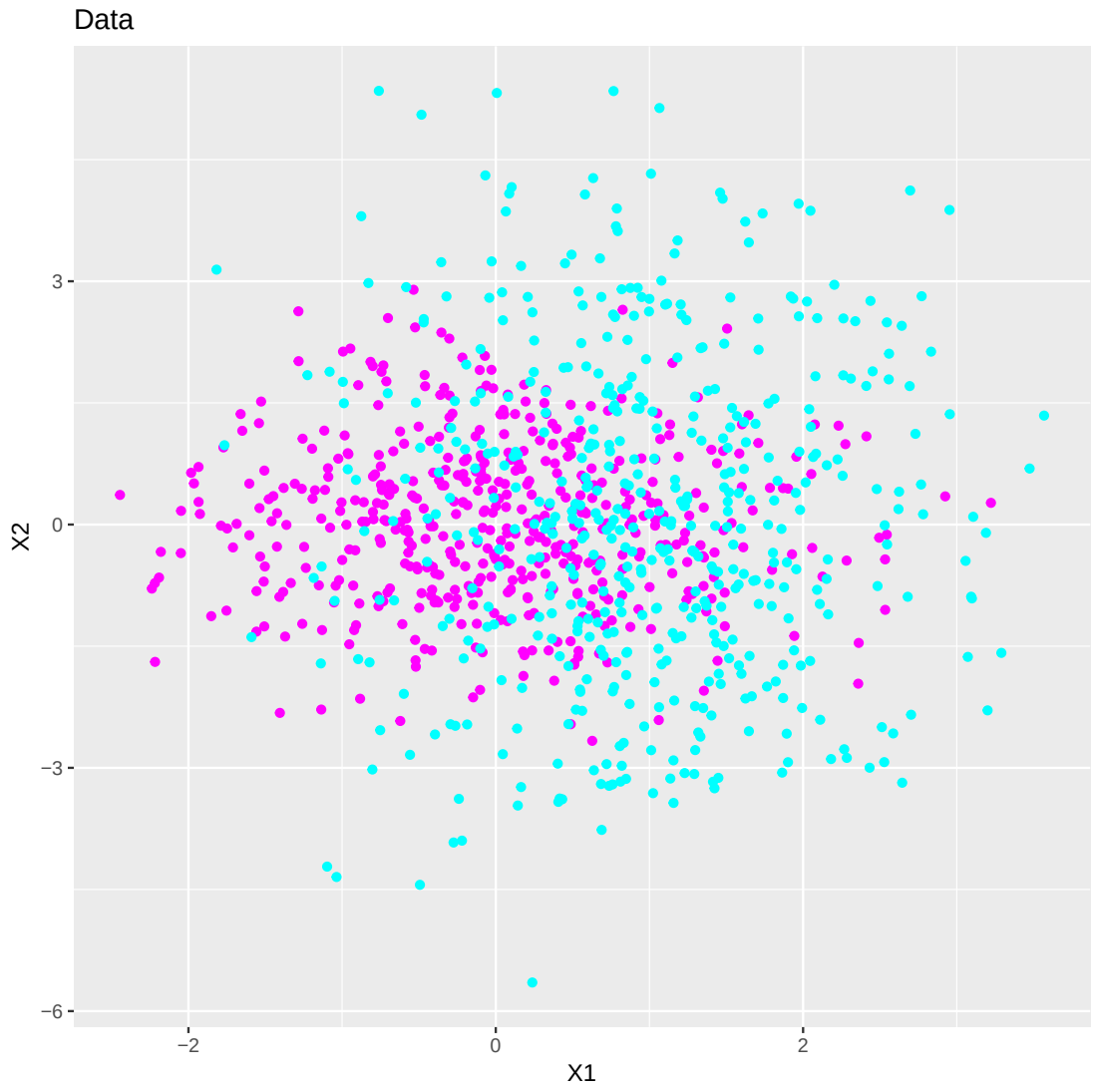
unbalanced mixtures or mixture weights and see for yourself what boundary K-means would lead to. Also notice the curvy nature of the boundary in EM classification rule.
Note that EM is not only useful for Gaussian mixture model, although it turns out to be very powerful for such a data generating mechanism. In general, if the probabilistic generation is a good assumption for the data, EM algorithm (or slight variants) of it work pretty well and K-means would work well if the clusters are relatively of similar size, well-seprated and have look locally spherical. *Note that both K-means and EM are known to get in local optima. So it is recommended to run both the algorithms with multiple restarts.*

```
library(MASS)
library(ggplot2)

# function to generate the data
gen_mixture_data <- function(probs, mus, sigs, N){


  # probs = K length vector with mixture weights
  # mus = K by d (d=2 for us) matrix with mixture means
  # sigs = d x d x K (d=2 for us) matrix with mixture covariances
  # N = number of data points to be generated

  K <- length(probs)

  #generate the mixture sizes
  num_points <- table(sample(1:K,prob=probs[1:K],size=N,replace=TRUE))

  x <- matrix(, nrow = 0, ncol=2)
  z <- c()
  for (k in 1:K){
    # functon to generate the multivariate Gaussian
    x <- rbind(x, mvrnorm(num_points[k], mus[k, ], sigs[,,k]))
    z <- c(z, rep(k, num_points[k]))
  }


  return(list("x"=x, "z"=z))
}

colors = c("magenta","cyan")

# function to scatter plot data using labels
plot_data <- function(x, z, title='Data'){
```

```
  p<- ggplot() + geom_point(aes(x=x[, 1], y=x[, 2]), color=colors[z]) +
  labs(x = "X1", y = "X2", title=title)
  return(p)
}
```

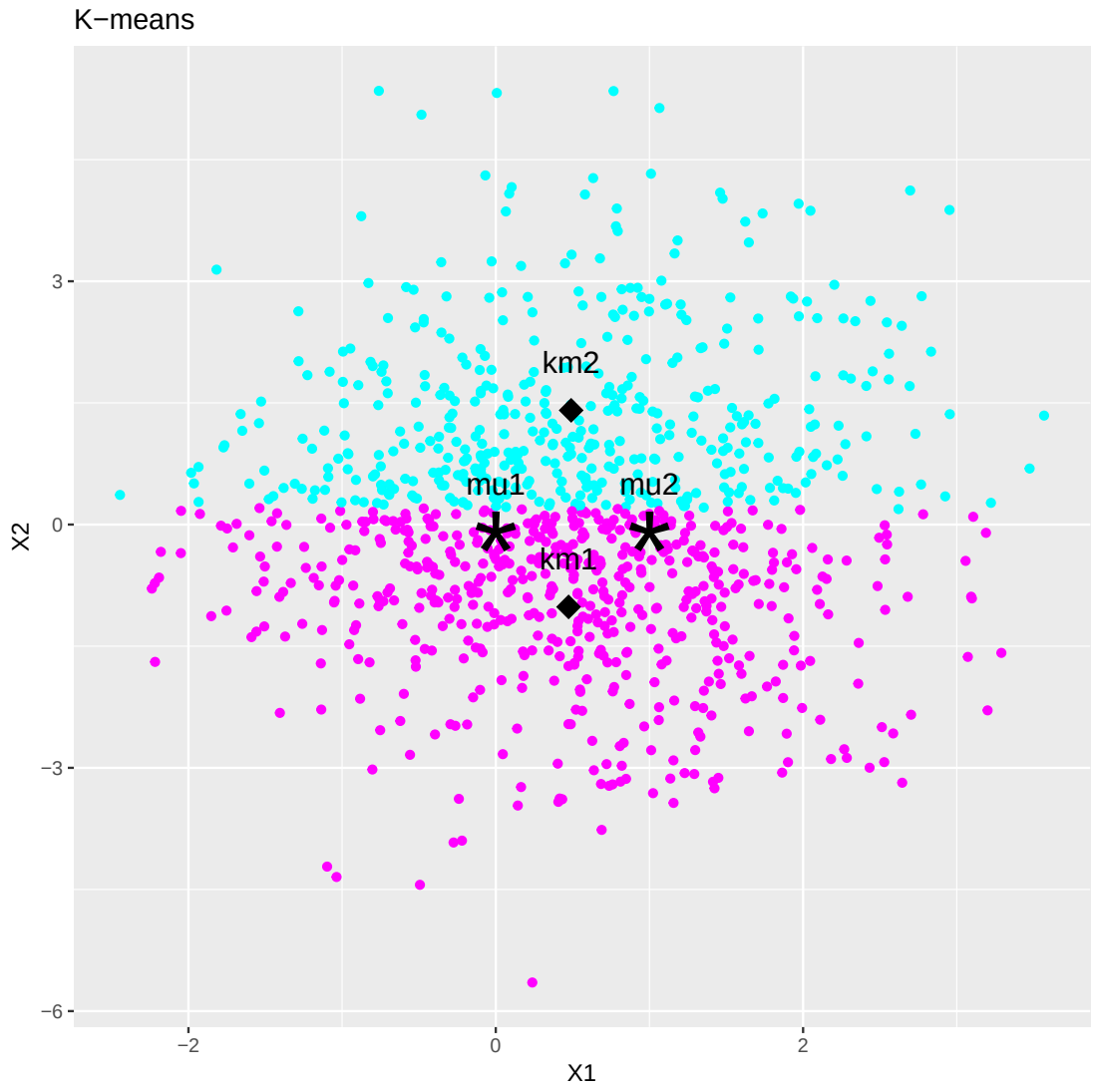
We now generate the data and scatter plot it.

```
N <- 1000
K <- 2
probs <- c(0.4,0.4)
mus <- matrix(c(0,0, 1, 0), 2, 2, byrow = TRUE)
sigs <- array(rep(NA,2*2*K), c(2,2,K))  # 3D matrix
sigs[, , 1] <- diag(2)
sigs[, , 2] <- diag(2)
sigs[2, 2, 2] <- 4

mixture_data <- gen_mixture_data(probs, mus, sigs, N)

x <- mixture_data$x
z <- mixture_data$z

p <- plot_data(x, z)
show(p)
```



We now run K-means and see that it produces slightly off-estimates.

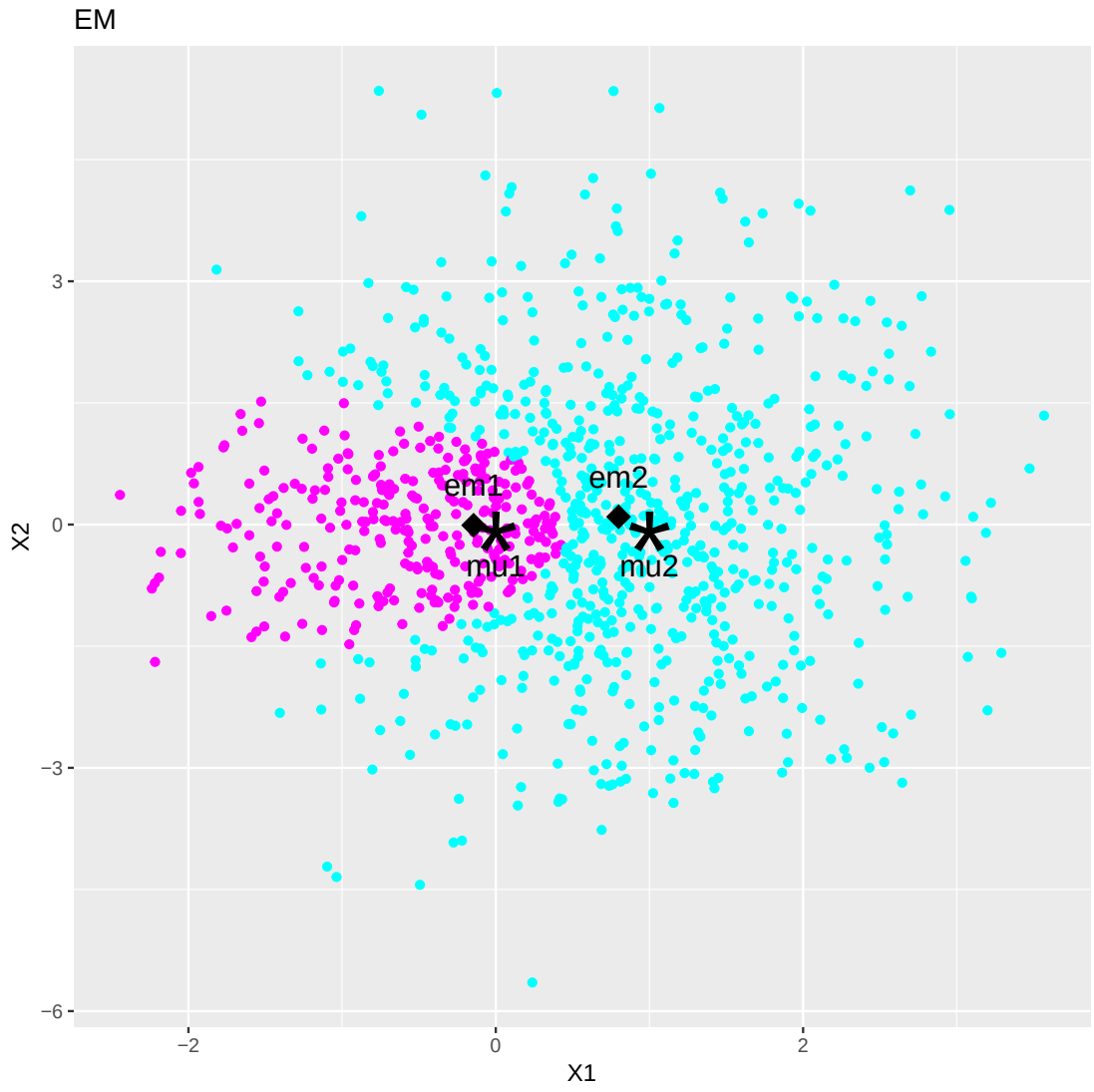
```
cl <- kmeans(x, K, 5)
z_km <- cl$cluster
p<- plot_data(x, z_km, "K-means")
p <- p + geom_point(aes(x=cl$centers[, 1], y=cl$centers[, 2]), shape=18, size=5) +
    geom_text(aes(x=cl$centers[, 1], y=cl$centers[, 2]+0.6,
                  label=c('km1', 'km2')), size=5) +
    geom_text(aes(x=mus[1:K, 1], y=mus[1:K, 2]+0.5, label=c('mu1', 'mu2')), size=5) +
    geom_point(aes(x=mus[1:K, 1], y=mus[1:K, 2]), shape='*', size=18)
show(p)
```



We now run EM and find that it produces accurate estimates.

```
library(mclust)

## Package 'mclust' version 5.4.2
## Type 'citation("mclust")' for citing this R package in publications.

model <- Mclust(x, G=2)
z_em = model$classification
em_means <- model$parameters$mean
p<- plot_data(x, z_em, "EM")
p<- p + geom_point(aes(x=em_means[1, ], y=em_means[2, ]), shape=18, size=5) +
  geom_text(aes(x=em_means[1, ], y=em_means[2, ]+0.5, label=c('em1', 'em2')), size=5) +
  geom_text(aes(x=mus[1:K, 1], y=mus[1:K, 2]-0.5, label=c('mu1', 'mu2')), size=5) +
```

```
  geom_point(aes(x=mus[1:K, 1], y=mus[1:K, 2]), shape='*', size=18)
  labs(title='EM')

## $title
## [1] "EM"
##
## attr(,"class")
## [1] "labels"

show(p)
```



```r
show(mus)
```

```
##      [,1] [,2]
## [1,]    0    0
## [2,]    1    0
```

```r
show((em_means))
```

```
##              [,1]       [,2]
## [1,] -0.144445078 0.79916829
## [2,] -0.007981805 0.09563068
```

```r
show((t(cl$centers)))
```

```
##               1         2
## [1,]  0.4738593 0.4904471
## [2,] -1.0154664 1.4083750
```