STAT-S675

Homework 11

John Koo

Link to assignment

Exercise 1

Problem setup

We are given:

Fix $x_i \leq \cdots \leq x_n \in \mathbb{R}$ such that $\sum_{i=1}^n x_i = 0$ and $\sum_{i=1}^n x_i^2 = n$.

We want to partition the x_is into two groups by minimizing either $W(k)$ or $R(k)$ as defined below. For this case ($\mathbb{R}^1$), we can say that an optimal partition will divide the x_is into left and right points (so we don't consider partitions where we have points on the outside in one group and points in the middle in another group, for example).

Define the centers of each partition $m_1 = \frac{1}{k}\sum_{i=1}^k x_i$ and $m_2 = \frac{1}{n-k}\sum_{i=k+1}^n x_i$ (note that m_1 and m_2 are functions of k).

Then let:

$$W(k) = \sum_{i=1}^{k}(x_i - m_1)^2 + \sum_{i=k+1}^{n}(x_i - m_2)^2$$

$$R(k) = \sum_{i=1}^{k}\left(x_i + \sqrt{\frac{n-k}{k}}\right)^2 + \sum_{i=k+1}^{n}\left(x_i - \sqrt{\frac{k}{n-k}}\right)^2$$

Show that minimizing $W(k)$ is equivalent to minimizing $R(k)$.

Proof

We can do this by showing:

$$W(k) = R(k) - \frac{(R(k))^2}{4n}$$

Then we can write $W(R) = R - \frac{R^2}{4n}$ (n is fixed). Note that this function has one maximum at $R = 2n$ and is strictly increasing for $R < 2n$. $R(k)$ cannot be greater than $2n$ since if we expand $R(k)$, we get:

$$R(k) = 2n + 2\sqrt{\frac{n-k}{k}}\sum_{i=1}^{k} x_i - 2\sqrt{\frac{k}{n-k}}\sum_{i=k+1}^{n} x_i$$

and $R(k)$ is maximized wrt k either at $k = 1$ or $k = n$. Also note that since the sum of the x_is is 0 and $x_1 < \cdots < x_n$, $\sum_1^k x_i$ is always negative. When $k = 1$, the third term goes to 0 and the first term is

strictly negative, so $R(k)$ is $2n$ plus some negative quantity, meaning it must be less than $2n$. On the other hand, when $k = n - 1$, the second term goes to 0 since the sum of x_is is 0, while the third term goes to $2\sqrt{k/(n-1)}x_n$ which is always positive since x_n must be positive to ensure the sum of x_is is 0. Then $R(k)$ is again less than $2n$.

Now note that if $W(R(k))$ is strictly increasing with respect to $R(k)$, then the minimum of our domain corresponds to the minimum of our range. Therefore, $W(k)$ and $R(k)$ must share the same minimum!

Show the relationship between W and R as specified above

Our claim is:

$$W(k) = R(k) - \frac{(R(k))^2}{4n}$$

If we expand the binomials of $W(k)$, we get:

$$W(k) = \sum_1^k x_i^2 - 2m_1 \sum_1^k x_i + km1^2 + \sum_{k+1}^n x_i^2 - 2m_2 \sum_{k+1}^n x_i + m_2^2(n-k)$$

Using our definitions of m_1 and m_2 and noting that $\sum_i^n x_i^2 = n$, we can rewrite this as:

$$\begin{aligned} W(k) &= n - \frac{2}{k}(\sum_1^k x_i)^2 + \frac{1}{k}(\sum_1^k x_i)^2 - \frac{2}{n-k}(\sum_{k+1}^n x_i)^2 + \frac{1}{n-k}(\sum_{k+1}^n x_i)^2 \\ &= n - \frac{1}{k}(\sum_1^k x_i)^2 - \frac{1}{n-k}(\sum_{k+1}^n x_i)^2 \\ &= n - km_1^2 - (n-k)m_2^2 \\ &= n - km_1^2 - nm_2^2 + km_2^2 \end{aligned}$$

Now note that since $km_1 + (n-k)m_2 = 0$, we can rewrite this as $-nm_2 = k(m_1 - m_2)$. So now we can rewrite $W(k)$ as :

$$\begin{aligned} W(k) &= n - km_1^2 + nm_2m_2 + km_2^2 \\ &= n - km_1^2 + k(m_1 - m_2)m_2 + km_2^2 \\ &= n + k(-m_1^2 + m_1m_2 - m_2^2 + m_2^2) \\ &= n + k(m_1m_2 - m_1^2) \\ &= n + km_1(m_2 - m_1) \\ &= n - (n-k)m_2(m_2 - m_1) \end{aligned}$$

Using the relationship $-nm_2 = k(m_1 - m_2) \implies m_2 - m_1 = \frac{nm_2}{k}$ from before, we can again rewrite $W(k)$:

$$\begin{aligned} W(k) &= n - (n-k)m_2\frac{nm_2}{k} \\ &= n\frac{(n-k)n}{k}m_2^2 \end{aligned}$$

And we use the same relationship again: $m_2 - m_1 = \frac{n}{k}m_2 \implies m_2 = (m_2 - m_1)\frac{k}{n}$.

Then we can finally write $W(k)$ as:

$$W(k) = n - \frac{(n-k)n}{k}\frac{k^2}{n^2}(m_1 - m_2)^2$$

$$\boxed{W(k) = n - \frac{(n-k)k}{n}(m_1 - m_2)^2}$$

On the other hand, if we expand $R(k)$:

$$R(k) = \sum_1^k x_i^2 + 2\sqrt{\frac{n-k}{k}}\sum_1^k x_i + n - k + \sum_{k+1}^n x_i^2 - 2\sqrt{\frac{k}{n-k}}\sum_{k+1}^n x_i + k$$
$$= 2n + 2\sqrt{k(n-k)}m_1 - 2\sqrt{k(n-k)}m_2$$

If we expand and simplify $-\frac{(R(k))^2}{4n}$, we get:

$$-\frac{(R(k))^2}{4n} = -n - \frac{k(n-k)}{n}m_1^2 - \frac{k(n-k)}{n}m_2^2 - 2m_1\sqrt{k(n-k)} + 2m_2\sqrt{k(n-k)} + \frac{2k(n-k)}{n}m_1m_2$$

Then noting that some terms cancel each other out, $R(k) - \frac{(R(k))^2}{4n}$:

$$n - \frac{k(n-k}{n}m_1^2 - \frac{k(n-k)}{n}m_2^2 + 2\frac{k(n-k)}{n}m_1m_2$$
$$= n - \frac{k(n-k)}{n}(m_1^2 + m_2^2 - 2m_1m_2)$$
$$\boxed{= n - \frac{k(n-k)}{n}(m_1 - m_2)^2}$$

Which is exactly the same as our expression for $W(k)$.

Therefore, since we can relate $W(k)$ and $R(k)$ as $W(R)$ and this function is strictly increasing on the domain of R, the minimum of W occurs at the minimum of R.

Exercise 2

Consider:

$$X = \left[\frac{v_1}{\sigma_1}\middle|\cdots\middle|\frac{v_d}{\sigma_d}\right]$$

vs

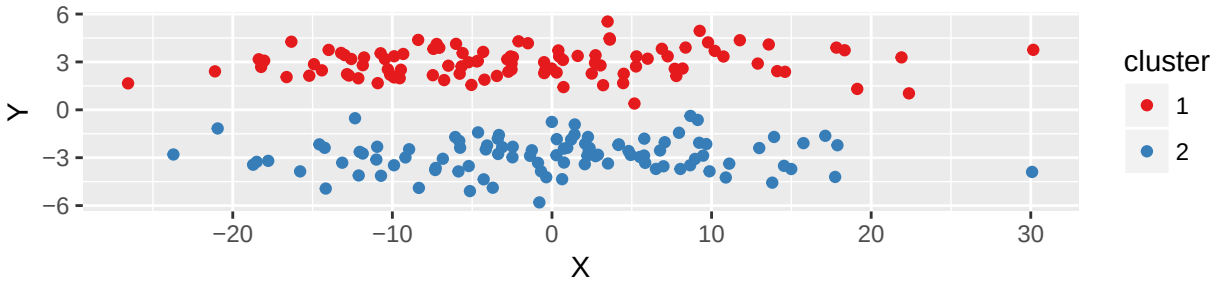
$$V = \left[v_1\middle|\cdots\middle|v_d\right]$$

Where the columns of X and V are our Euclidean feature vectors. Then X is just a scaling of V (or vice versa).

Also recall that k-means works best on spherical clusters. That means that if a cluster is too elongated, k-means may decide to split it up into two or more clusters in an attempt to make things more spherical. This is typically unwanted behavior. So scaling the data can cause k-means to create different clusters.

We can show this by constructing a dataset:

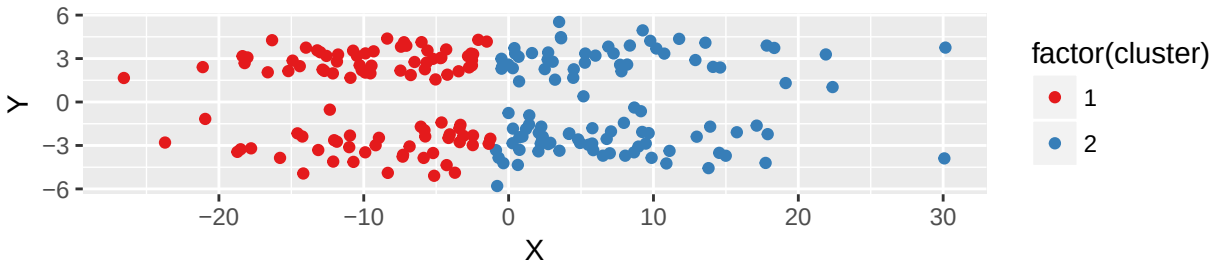
```
dp <- loadNamespace('dplyr')
import::from(magrittr, `%>%`, `%<>%`)
library(ggplot2)

set.seed(675)

cluster.1.df <- dplyr::data_frame(
  X = rnorm(100, sd = 10),
  Y = rnorm(100, mean = 3, sd = 1),
  cluster = '1'
)
cluster.2.df <- dplyr::data_frame(
  X = rnorm(100, sd = 10),
  Y = rnorm(100, mean = -3, sd = 1),
  cluster = '2'
)
cluster.df <- dp$bind_rows(cluster.1.df, cluster.2.df)

ggplot(cluster.df) +
  geom_point(aes(x = X, y = Y, colour = cluster)) +
  coord_fixed() +
  scale_colour_brewer(palette = 'Set1')
```



. . . and then clustering it via k-means:

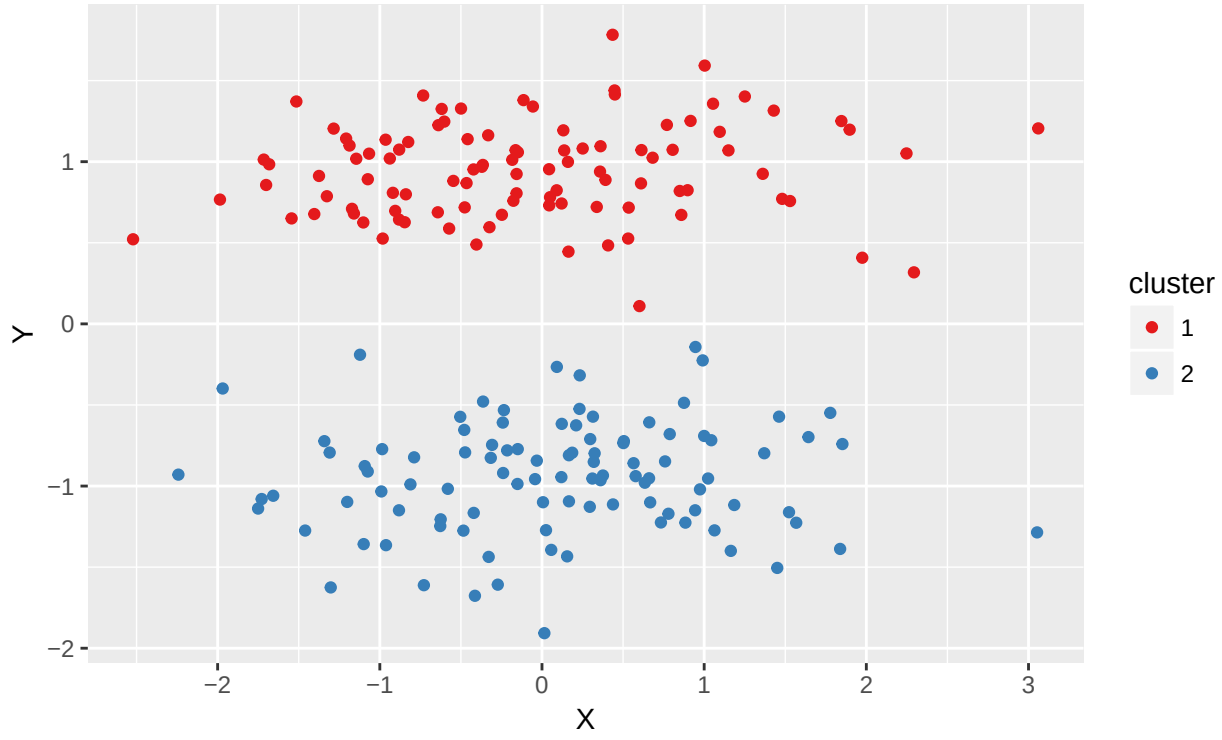
```
kmeans.lloyd <- function(X, init.center, n.iter = 100) {
  # how many clusters?
  k <- length(init.center)

  # how many objects?
  N <- nrow(X)

  # initialize
  # find the centers of the clusters
  centers.df <- lapply(init.center, function(i) X[i, ]) %>%
    dplyr::bind_rows()
  # init cluster assignments
  X$cluster <- rep(0, N)

  # iteration counter for breaking out when there's no convergence
  iter.counter <- 0

  # also need an "old centers.df" to compare against current
  # if they're the same then we're done
  old.centers.df <- dplyr::data_frame()

  # keep iterating until old.centers.df is the same as centers.df
  while(!identical(old.centers.df, centers.df)) {
    # check if we converged within set number of iterations
    iter.counter %<>% magrittr::add(1)
```

```
    if (iter.counter >= n.iter) {
      warning('did not converge :(')
      break
    }

    # assign each object to a cluster by euclidean distance to centers
    X$cluster <- sapply(seq(N), function(i) {
      sapply(seq(k), function(j) {
        sum((X[i, -ncol(X)] - centers.df[j, ]) ** 2)
      }) %>%
        which.min()
    })

    # copy current centers.df to old.centers.df
    old.centers.df <- centers.df

    # update centers.df
    centers.df <- X %>%
      dplyr::group_by(cluster) %>%
      dplyr::summarise_all(mean) %>%
      dplyr::ungroup() %>%
      dplyr::arrange(cluster) %>%
      dplyr::select(-cluster)
  }

  # compute W
  W <- X %>%
    dplyr::group_by(cluster) %>%
    dplyr::summarise_all(dplyr::funs(sum((. - mean(.)) ** 2))) %>%
    dplyr::ungroup() %>%
    dplyr::select(-cluster) %>%
    rowSums() %>%
    sum()


  return(list(X = X, centers = centers.df, W = W, k = k, niter = iter.counter))
}

kmeans <- kmeans.lloyd(dp$select(cluster.df, X, Y), c(1, 101))
ggplot(kmeans$X) +
  geom_point(aes(x = X, y = Y, colour = factor(cluster))) +
  coord_fixed() +
  scale_colour_brewer(palette = 'Set1')
```

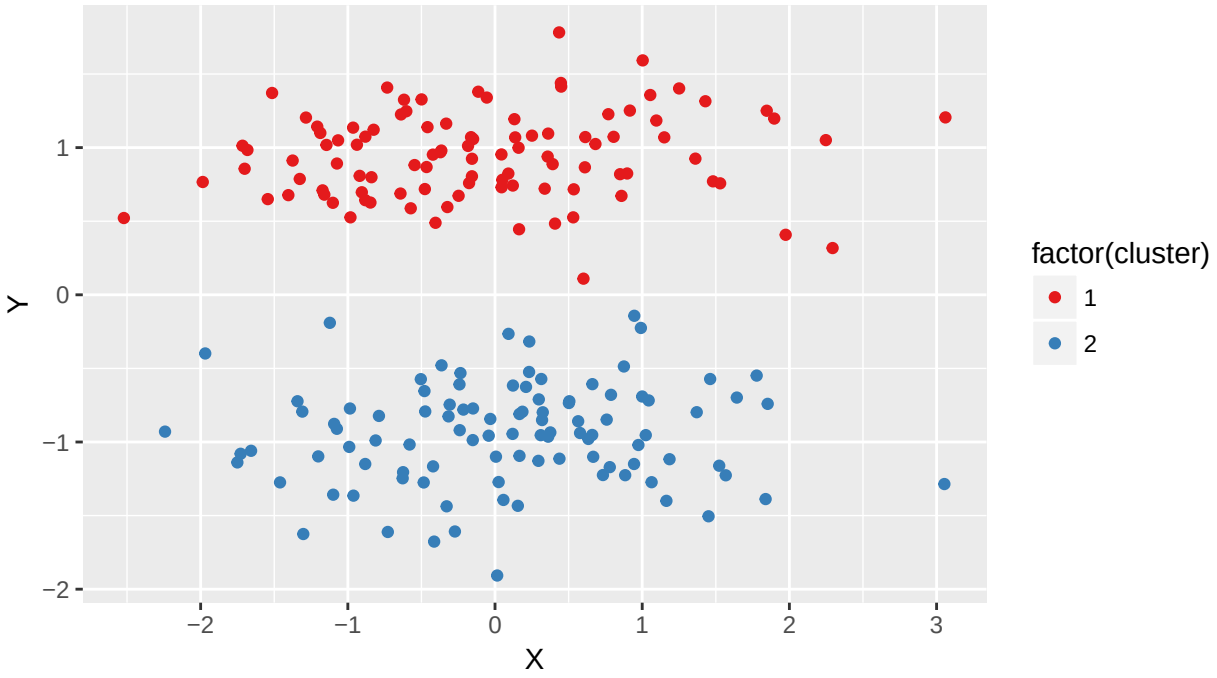


So if we rescale these data:

```
scaled.df <- cluster.df %>%
  dp$mutate(X = (X - mean(X)) / sd(X),
            Y = (Y - mean(Y)) / sd(Y))

ggplot(scaled.df) +
  geom_point(aes(x = X, y = Y, colour = cluster)) +
  scale_colour_brewer(palette = 'Set1') +
  coord_fixed()
```




```
kmeans.scaled <- kmeans.lloyd(dp$select(scaled.df, X, Y), c(1, 101))
ggplot(kmeans.scaled$X) +
  geom_point(aes(x = X, y = Y, colour = factor(cluster))) +
  coord_fixed() +
  scale_colour_brewer(palette = 'Set1')
```




```
table(kmeans$X$cluster, kmeans.scaled$X$cluster)
```

```
    1  2
1  55 43
2  45 57
```

So if the scaled vs unscaled embedding results in something like this, then we can imagine that the result of the k-means clustering will be different for each case.