

MSMS 206 : Practical 01

Ananda Biswas

March 10, 2025



Question : Perform k -means clustering for $\{2, 4, 10, 12, 3, 20, 30, 11, 25\}$ for $k = 2$. Assume 2 and 4 as initial cluster centroids.

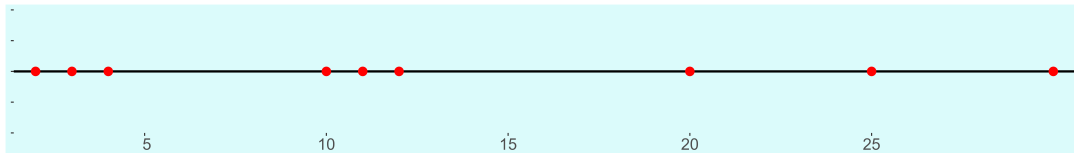
⊕ After a choice of initial centroids, the k -means clustering algorithm is as follows :

- (1) calculate the distance of each data-point from each of the centroids
- (2) assign each of the data-points to its closest centroid
- (3) relocate the centroids to the average location of the data-points of similar group

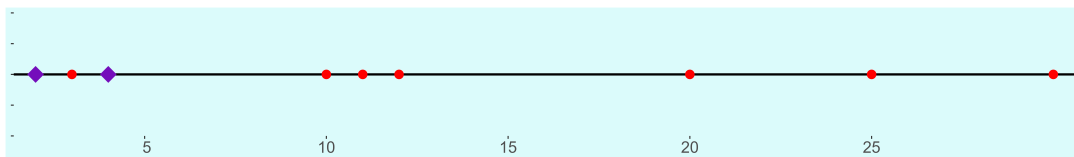
And we repeat this procedure until the assignments don't change after the centroid locations were recomputed.

```
df <- data.frame(x = c(2, 4, 10, 12, 3, 20, 30, 11, 25))
```

Let us have a look at the data-points.



Now We put the initial centroids.



```
m <- dim(df)[1] # number of data-points
n <- dim(df)[2] # dimension of data-points

k <- 2 # number of clusters
```

```
X <- as.matrix(df)
```

Now we initialize the centroids as 2 and 4.

```
centroid <- matrix(data = c(2,
                           4),
                  nrow = k, ncol = 1, byrow = TRUE)
```

We now deploy our k -means clustering algorithm. We created a list named *iteration_record()* for visualization of the process that will come later.

```
cluster <- c()

iteration_record <- list()

repeat{
  dist_mat <- matrix(0, nrow = m, ncol = k)

  for (i in 1:k) {
    d <- apply(X, 2, FUN = function(x) return(x - as.vector(centroid[i, ])))

    dist_mat[,i] <- sqrt(diag( d %*% t(d) ) )
  }


  cluster <- apply(dist_mat, 1, FUN = function(x) return(which(x == min(x))[1]))

  new_centroid <- matrix(data = 0, nrow = k, ncol = n)

  for (i in 1:k) {
    new_centroid[i, ] <- mean(X[which(cluster == i), ])
  }

  iteration_record <- append(iteration_record,
                            list(list(mat = cbind(X, dist_mat, cluster),
                                      new_centroid = new_centroid)))

  if(any(centroid - new_centroid != 0)){
    centroid <- new_centroid
  } else{
    break
  }
}
```

 The final clustering of the data-points is as follows :

```
cluster
## [1] 1 1 1 1 1 2 2 1 2
```

```
length(iteration_record)
```

```
## [1] 5
```

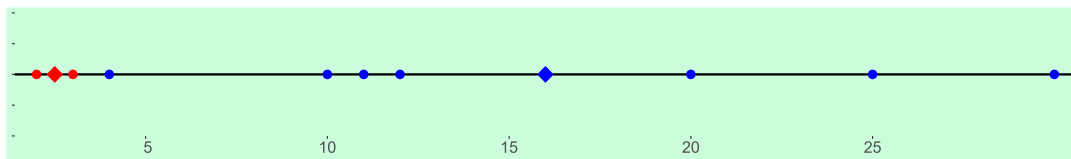
There were 5 iterations, we shall take a look at them one by one.

👉 Iteration 1

```
iteration_record[[1]]$mat
```

```
##           x distance_from_centroid_1 distance_from_centroid_2 cluster
## [1,]  2              0              2              1
## [2,]  4              2              0              2
## [3,] 10              8              6              2
## [4,] 12             10              8              2
## [5,]  3              1              1              1
## [6,] 20             18             16              2
## [7,] 30             28             26              2
## [8,] 11              9              7              2
## [9,] 25             23             21              2
```

The data-points along with relocated centroids are as follows :

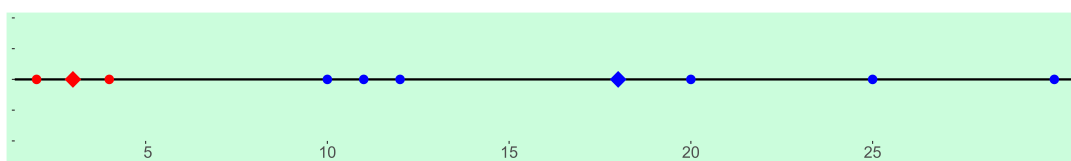


👉 Iteration 2

```
iteration_record[[2]]$mat
```

```
##           x distance_from_centroid_1 distance_from_centroid_2 cluster
## [1,]  2              0.5             14              1
## [2,]  4              1.5             12              1
## [3,] 10              7.5              6              2
## [4,] 12              9.5              4              2
## [5,]  3              0.5             13              1
## [6,] 20             17.5              4              2
## [7,] 30             27.5             14              2
## [8,] 11              8.5              5              2
## [9,] 25             22.5              9              2
```

The data-points along with relocated centroids are as follows :

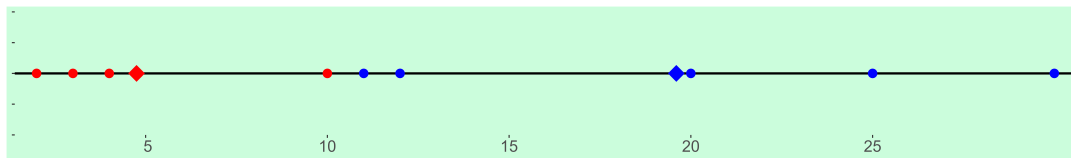


Iteration 3

```
iteration_record[[3]]$mat
```

##		x	distance_from_centroid_1	distance_from_centroid_2	cluster
##	[1,]	2	1	16	1
##	[2,]	4	1	14	1
##	[3,]	10	7	8	1
##	[4,]	12	9	6	2
##	[5,]	3	0	15	1
##	[6,]	20	17	2	2
##	[7,]	30	27	12	2
##	[8,]	11	8	7	2
##	[9,]	25	22	7	2

The data-points along with relocated centroids are as follows :

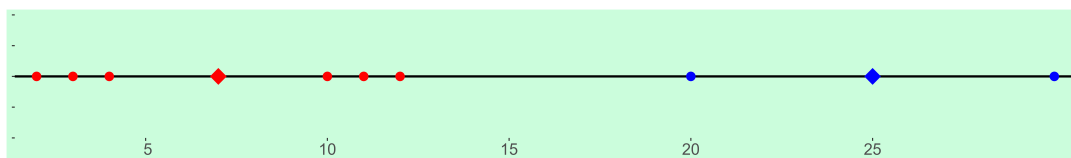


Iteration 4

```
iteration_record[[4]]$mat
```

##		x	distance_from_centroid_1	distance_from_centroid_2	cluster
##	[1,]	2	2.75	17.6	1
##	[2,]	4	0.75	15.6	1
##	[3,]	10	5.25	9.6	1
##	[4,]	12	7.25	7.6	1
##	[5,]	3	1.75	16.6	1
##	[6,]	20	15.25	0.4	2
##	[7,]	30	25.25	10.4	2
##	[8,]	11	6.25	8.6	1
##	[9,]	25	20.25	5.4	2

The data-points along with relocated centroids are as follows :

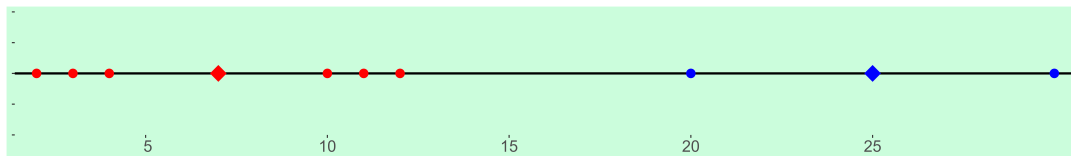



Iteration 5

```
iteration_record[[5]]$mat
```

##		x	distance_from_centroid_1	distance_from_centroid_2	cluster
##	[1,]	2	5	23	1
##	[2,]	4	3	21	1
##	[3,]	10	3	15	1
##	[4,]	12	5	13	1
##	[5,]	3	4	22	1
##	[6,]	20	13	5	2
##	[7,]	30	23	5	2
##	[8,]	11	4	14	1
##	[9,]	25	18	0	2

The data-points along with relocated centroids are as follows :



 We notice that there is no change in location centroids from Iteration 4 to Iteration 5. So the process stops and we get our final set of clusters.