Lab 13 - Jin Kweon (3032235207)

Jin Kweon

11/25/2017

K-means Clustering

https://www.youtube.com/watch?v=_aWzGGNrcic

https://sites.google.com/site/dataclusteringalgorithms/k-means-clustering-algorithm

https://home.deib.polimi.it/matteucc/Clustering/tutorial_html/kmeans.html

https://en.wikipedia.org/wiki/K-means_clustering

https://stats.stackexchange.com/questions/56500/what-are-the-main-differences-between-k-means-and-k-nearest-neighbours

https://www.quora.com/What-is-the-difference-between-k-means-and-hierarchical-clustering

I will use Euclidean distance for distance check.

Q. I remember Professor say Cluster is belonged to categorical unsupervised method in the lecture 1. But, why is that....??? Seems like its continuous unsupervised method....

Q. Does it always converge 100%? Or, is it possible to getting closer (meaning almost no changes) but not 100% converge?

Q. I am little bit confused... in lecture slide 24 & hw 5, when we do when X is a vetor, not matrix, we have sum of squares being matrix, not a number... But, how can ours be just a number...?

```
#iris %>% ggvis(~Sepal.Length, ~Sepal.Width, fill = ~Species) %>% layer_points()
#iris %>% ggvis(~Petal.Length, ~Petal.Width, fill = ~Species) %>% layer_points()

summary(iris)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
##        Species
##  setosa    :50
##  versicolor:50
##  virginica :50
##
##
##
```

```
euclid <- function(x, w){
  dist <- sqrt(sum((x - w)^2))
  return(dist)
}

total <- function(predictors){
  center <- apply(predictors, 2, mean)
```

```
  v <- rep(0, ncol(predictors))
  for(i in 1:nrow(predictors)){
    v <- v + (predictors[i,] - center)^2
  }
  return(sum(v))
}

wss <- function(x, y, means){
  if(nrow(x) != length(y)){
    stop("predictor variable and response variable have different lengths...")
  }else{
    combined <- as.data.frame(cbind(y = y, x = x))
    combined$y <- as.factor(combined$y)
    splited <- split(combined, combined$y)
    ans <- c()

    for(i in 1:length(levels(combined$y))){
      center <- means[i,]
      v <- 0
      for(j in 1:nrow(splited[[i]])){
        v <- v + sum((splited[[i]][j,-1] - center)^2)
      }
      ans[i] <- v
    }
    return(ans)
  }
}

my_kmeans <- function(X, k){
  initial_iter <- sample.int(dim(X)[1], k)
  initial_mean <- X[initial_iter, ]

  group <- c()
  for(i in 1:dim(X)[1]){
    distance <- c()
    for(j in 1:k){
      distance[j] <- euclid(as.vector(initial_mean[j, ]), as.vector(X[i, ]))
    }
    group[i] <- order(distance)[1] #tie ignore like knn
  }

  new_mean <- as.data.frame(matrix(0, k, dim(X)[2]))
    for(i in 1:k){
      new_mean[i, ] <- apply(X[group == i, ], 2, mean)
    }

    newgroup <- c()
    for(i in 1:dim(X)[1]){
      distance <- c()
      for(j in 1:k){
        distance[j] <- euclid(as.vector(new_mean[j, ]), as.vector(X[i, ]))
      }
      newgroup[i] <- order(distance)[1] #tie ignore like knn
```

```
    }
  oldgroup <- group
  group <- newgroup
  condition <- 100
  while (condition != 0) {

    new_mean <- as.data.frame(matrix(0, k, dim(X)[2]))
    for(i in 1:k){
      new_mean[i, ] <- apply(X[group == i, ], 2, mean)
    }

    newgroup <- c()
    for(i in 1:dim(X)[1]){
      distance <- c()
      for(j in 1:k){
        distance[j] <- euclid(as.vector(new_mean[j, ]), as.vector(X[i, ]))
      }
      newgroup[i] <- order(distance)[1] #tie ignore like knn
    }

    oldgroup <- group
    group <- newgroup
    condition <- sum(group != oldgroup)
  }
  cluster_size <- as.vector(table(group))

  new_mean <- as.data.frame(matrix(0, k, dim(X)[2]))
  for(i in 1:k){
    new_mean[i, ] <- apply(X[group == i, ], 2, mean)
  }
  cluster_means <- new_mean
  clustering_vector <- group

  wss_cluster <- wss(X, clustering_vector, cluster_means)

  totalwss <- sum(wss_cluster)
  bss_over_tss <- ((total(X) - totalwss) / total(X)) * 100

  lists <- list(cluster_size, as.matrix(cluster_means), clustering_vector, wss_cluster, bss_over_tss)
  return(lists)
}

my_kmeans(X = iris[,1:4], k = 3)
```

```
## [[1]]
## [1] 62 38 50
##
## [[2]]
##           V1       V2       V3       V4
## [1,] 5.901613 2.748387 4.393548 1.433871
## [2,] 6.850000 3.073684 5.742105 2.071053
## [3,] 5.006000 3.428000 1.462000 0.246000
##
## [[3]]
```

1. Randomly select K rows of the dataset and treat them as the initial cluster centroids $g_1, ..., g_K$.
2. For each observation x_i,
   a. compute $d(x_i, g_k) = ||x_i - g_k||_2^2$ for each k.
   b. Find $k^* = \arg\min_{k=1,...,K} d(x_i, g_k)$.
   c. Assign x_i to cluster k^*.
3. For each cluster C_k, compute the cluster centroid $g_k = |C_k|^{-1} \sum_{i \in C_k} x_i$.
4. Repeat Step 2 and Step 3 until convergence (that is, the cluster assignment does not change).

Figure 1: Algorithm

```
##   [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
##  [36] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [71] 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 2 2
## [106] 2 1 2 2 2 2 2 2 1 1 2 2 2 2 1 2 1 2 1 2 2 1 1 2 2 2 2 2 1 2 2 2 2 1 2
## [141] 2 2 1 2 2 2 1 2 2 1
##
## [[4]]
## [1] 39.82097 23.87947 15.15100
##
## [[5]]
## [1] 88.42753
```

```
kmeans(iris[,1:4], centers = 3)
```

```
## K-means clustering with 3 clusters of sizes 50, 62, 38
##
## Cluster means:
##   Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1     5.006000    3.428000     1.462000    0.246000
## 2     5.901613    2.748387     4.393548    1.433871
## 3     6.850000    3.073684     5.742105    2.071053
##
## Clustering vector:
##   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [36] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [71] 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 3 3 3
## [106] 3 2 3 3 3 3 3 3 2 2 3 3 3 3 2 3 2 3 2 3 3 2 2 3 3 3 3 3 2 3 3 3 3 2 3
## [141] 3 3 2 3 3 3 2 3 3 2
##
## Within cluster sum of squares by cluster:
## [1] 15.15100 39.82097 23.87947
##  (between_SS / total_SS =  88.4 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

Hierarchial Clustering

Q. How to answer the last question: "Based on the results from cutree(), how does each linkage perform? Do they all separate (roughly) the observations into correct group?"

```
#summary(hclust(dist(iris[,1:4])^2, method  = "complete"))
hc.complete <- iris[,1:4] %>% dist(method = "euclidean") %>% hclust(method = "complete")
summary(hc.complete)
```

```
##             Length Class  Mode
## merge       298    -none- numeric
## height      149    -none- numeric
## order       150    -none- numeric
## labels        0    -none- NULL
## method        1    -none- character
## call          3    -none- call
## dist.method   1    -none- character
```

```
hc.average <- iris[,1:4] %>% dist(method = "euclidean") %>% hclust(method = "average")
summary(hc.average)
```

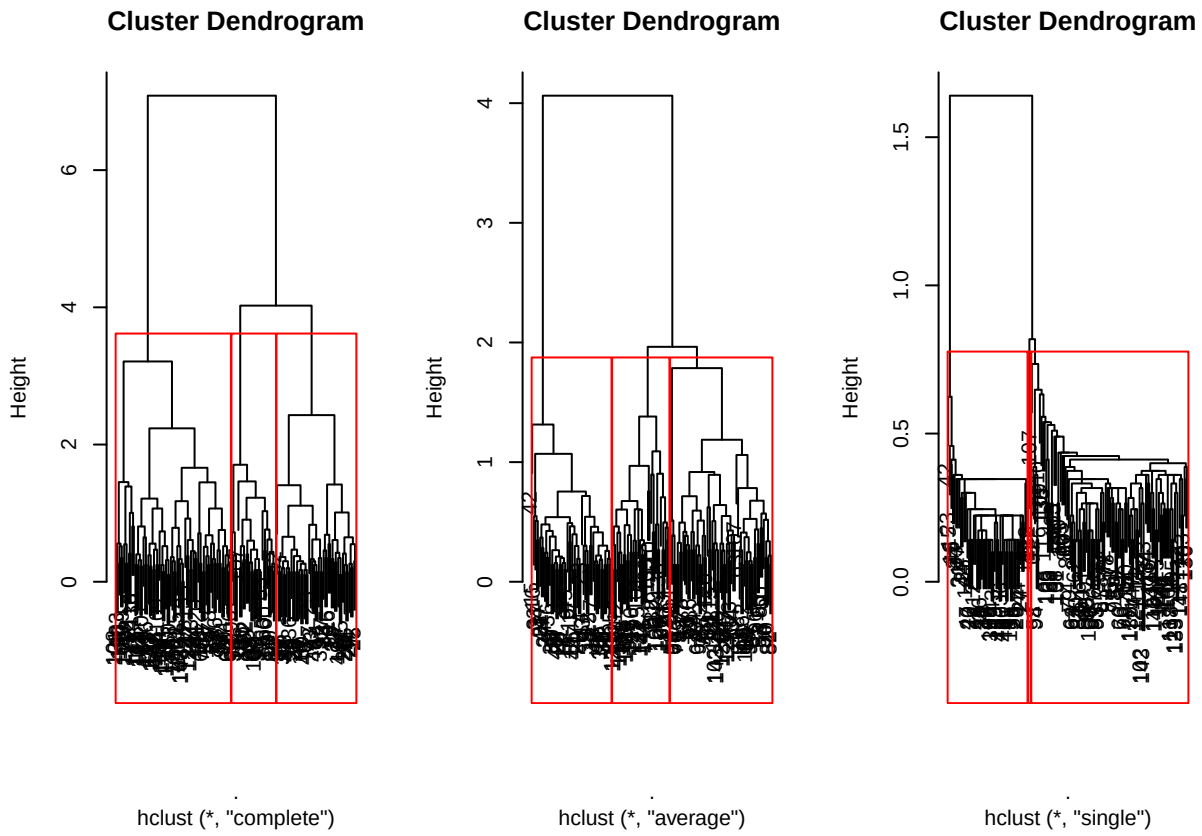
```
##             Length Class  Mode
## merge       298    -none- numeric
## height      149    -none- numeric
## order       150    -none- numeric
## labels        0    -none- NULL
## method        1    -none- character
## call          3    -none- call
## dist.method   1    -none- character
```

```
hc.single <- iris[,1:4] %>% dist(method = "euclidean") %>% hclust(method = "single")
summary(hc.single)
```

```
##             Length Class  Mode
## merge       298    -none- numeric
## height      149    -none- numeric
## order       150    -none- numeric
## labels        0    -none- NULL
## method        1    -none- character
## call          3    -none- call
## dist.method   1    -none- character
```

```
par(mfrow = c(1,3))
plot(hc.complete)
rect.hclust(hc.complete, k = 3, border="red")
plot(hc.average)
rect.hclust(hc.average, k = 3, border="red")
```

```
plot(hc.single)
rect.hclust(hc.single, k = 3, border="red")
```




```
a <- cutree(hc.complete, k = 3)
b <- cutree(hc.average, k = 3)
c <- cutree(hc.single, k = 3)
table(a, b)
```

```
##    b
## a    1  2  3
##   1 50  0  0
##   2  0 36 36
##   3  0 28  0
```

```
table(a, c)
```

```
##    c
## a    1  2  3
##   1 50  0  0
##   2  0 70  2
##   3  0 28  0
```

```
table(b, c)
```

```
##    c
## b    1  2  3
##   1 50  0  0
##   2  0 64  0
##   3  0 34  2
```

```
a
```

```
##   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [36] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 3 2 3 2 3 2 3 3 3 3 2 3 2 3 3 2 3
##  [71] 2 3 2 2 2 2 2 2 2 3 3 3 3 2 3 2 2 2 3 3 3 2 3 3 3 3 3 2 3 3 2 2 2 2 2
## [106] 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [141] 2 2 2 2 2 2 2 2 2 2
```

```
b
```

```
##   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [36] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [71] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 3 3 3
## [106] 3 2 3 3 3 3 3 3 2 2 3 3 3 3 2 3 2 3 2 3 3 2 2 3 3 3 3 2 3 3 3 3 2 3
## [141] 3 3 2 3 3 3 2 3 3 2
```

```
c
```

```
##   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [36] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [71] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [106] 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2
## [141] 2 2 2 2 2 2 2 2 2 2
```

```
table(a)
```

```
## a
##  1  2  3
## 50 72 28
```

```
table(b)
```

```
## b
##  1  2  3
## 50 64 36
```

```
table(c)
```

```
## c
##  1  2  3
## 50 98  2
```