Lloyd's K-Means algorithm

Jin Seo Jo

Implement Lloyd's K-Means algorithm:

```
my_kmeans <- function(data, k, n_starts) {
  done = FALSE

  n = dim(data)[1] #data is a matrix, where each row is one data point

  if(k==1) {
  cluster = rep(1,n) #this vector says which cluster each point is in
  centers = apply(X = data, MARGIN=2, FUN=mean)
  cost = sum((data-centers[cluster])^2)
  return(list(cluster=cluster, cost=cost))
  }

  cluster_old = rep(1,n)
  cost_old = Inf

  for (run in 1:n_starts) {
  cluster = rep(1,n) #this vector says which cluster each point is in

  #uniformly choose initial cluster centers
  centers = data[sample(x=1:n,size = k, replace = FALSE),]

  while (!done) {
    # Do Step 2.1
    d = matrix(nrow=n,ncol=k)
    for (j in 1:k) {
      d[,j] = apply(X = data, MARGIN = 1, FUN = function(d) sum((d - centers[j,])^2))
    }

    cluster_new = apply(X=d, MARGIN=1, FUN = which.min)

    if(length(unique(cluster_new))< k ) stop("Empty cluster!")

    # Do Step 2.2
    for (i in 1:k){
      centers[i,] = apply(X = data[cluster_new==i,], MARGIN=2, FUN=mean)
    }

    # Check if the cluster assignements changed. If they have, set done=TRUE
    if (all(cluster==cluster_new)) {
      done=TRUE
    }
```

```
    cluster=cluster_new
  }

  cost = sum((data-centers[cluster,])^2)

  if (cost_old < cost) {
    cluster = cluster_old
    cost = cost_old
  }

  cost_old = cost
  cluster_old = cluster
}
return(list(cluster=cluster, cost=cost))
}
```