Week 9 Exercises

Aaron Palumbo

October 22, 2015

Contents

7.4.1 (section 7.4.3)

Consider two clusters that are a circle and a surrounding ring, as in the running example of this section.

Under the following conditions,

i. The radius of the circle is c.

ii. The inner and outer circles forming the ring have radii i and o, respectively.

iii. All representative points for the two clusters are on the boundaries of the clusters.

iv. Representative points are moved 20% of the distance from their initial position toward the centroid of their cluster.

v. Clusters are merged if, after repositioning, there are representative points from the two clusters at distance d or less.

In terms of d, c, i, and o, under what circumstances will the ring and circle be merged into a single cluster?

$.8i - .8c \leq d$

7.5.1 (section 7.5.5)

Using the cluster representation of Section 7.5.1, represent the twelve points of Fig. 7.8 as a single cluster. Use parameter $k = 2$ as the number of close and distant points to be included in the representation. *Hint*: Since the distance is Euclidean, we can get the square of the distance between two points by taking the sum of the squares of the differences along the x- and y-axes.

```
sqDist <- function(p1, p2) {
  return( (p1[1] - p2[1])**2 + (p1[2] - p2[2])**2 )
}

cluster <- matrix(c(
  4, 10,
  7, 10,
  4, 8,
  6, 8,
  10, 5,
  12, 6,
  11, 4,
  9, 3,
  12, 3,
```

```r
  3, 4,
  2, 2,
  5, 2),
  byrow=TRUE,
  nrow=12)

adjlist <- expand.grid(1:nrow(cluster), 1:nrow(cluster))
names(adjlist) <- c("r", "c")
adjmat <- matrix(
  apply(adjlist, 1, function(lr) {sqDist(cluster[lr['r'], ],
                                          cluster[lr['c'], ])
  }), nrow=nrow(cluster))

rsum <- apply(adjmat, 1, sum)
```

The following features form the *representation* of the cluster:

1. **N, the number of points in the cluster.**
   N = 12

2. **The clustroid of the cluster, which is defiend specifically to be the point in the cluster that minimizes the sum of the squares of the distances to other other points; that is, the clustroid is the point in the cluster with the smallest ROWSUM.**

```r
(clustroid <- cluster[which.min(rsum), ])
```

```
## [1] 6 8
```

The clustroid is (6, 8)

3. **The rowsum of the clustroid of the cluster.**

$\text{rowsum}_{\text{clustroid}} = 332$

4. **For some chosen constant k, the k points of the cluster that are closest to the clustroid, and their rowsums. These points are poart of the representation in case the ddition of points to the cluster causes the clustroid to chhange. The assumption is made that the new clustroid woudl be one of these k points nesr the old clustroid.**

For $k = 2$ we have:

```r
kclosest <- (1:ncol(adjmat))[order(adjmat[which.min(rsum), ])[2:3]]
kpoints <- cbind(cluster[kclosest, ],
                 rsum[kclosest])
trash <- apply(kpoints, 1, function(r) {
  print(paste0("(", r[1], ", ", r[2], ") => rowsum = ", r[3]))
})
```

```
## [1] "(4, 8) => rowsum = 432"
## [1] "(7, 10) => rowsum = 490"
```

5. The k points of the cluster that are furthest from the clustroid and their rowsums. These points are part of the representation so that we can consider whether two clusters are clsee enough to merge. the assumption is made that if two clusters are close, then a pair of points distant from their respective clustroids woudld be close.

Again for $k = 2$ we have:

```
kfarthest <-
  (1:ncol(adjmat))[order(adjmat[which.min(rsum), ], decreasing=TRUE)[1:2]]
kpoints <- cbind(cluster[kfarthest, ],
                 rsum[kfarthest])
trash <- apply(kpoints, 1, function(r) {
  print(paste0("(", r[1], ", ", r[2], ") => rowsum = ", r[3]))
})
```

```
## [1] "(12, 3) => rowsum = 598"
## [1] "(2, 2) => rowsum = 688"
```