Data Mining with R

4. 군집화 (Clustering)

Contents

Clustering (군집화, Cluster Analysis)

- Cluster (군집): 데이터의 집합
  - 동일 그룹내의 다른 데이터와는 유사
  - 다른 그룹내의 데이터와는 관련 없음
- Cluster analysis (clustering, 군집화)
  - 데이터 간의 유사성을 평가하여 유사한 데이터들을 군집으로 묶는 것
- 비교사 학습 (Unsupervised learning): 기 정의된 class가 없음
- 응용 분야
  - 데이터의 분포에 대한 영감을 얻기 위한 도구
  - 다른 알고리즘의 전처리 단계에서 사용되는 도구

Data 이해를 위한 Clustering

- Biology: 생명체에 대한 taxonomy: kingdom, phylum, class, order, family, genus and species
- Information retrieval: document clustering
- Land use: Identification of areas of similar land use in an earth observation database
- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- City-planning: Identifying groups of houses according to their house type, value, and geographical location
- Earth-quake studies: Observed earth quake epicenters should be clustered along continent faults
- Climate: understanding earth climate, find patterns of atmospheric and ocean
- Economic Science: market resarch

전처리를 위한 Clustering

- Summarization:
  - Preprocessing for regression, PCA, classification, and association analysis
- Compression:
  - Image processing: vector quantization
- Finding K-nearest Neighbors
  - Localizing search to one or a small number of clusters
- Outlier detection
  - Outliers are often viewed as those “far away” from any cluster

좋은 Clustering이란?

- A good clustering method will produce high quality clusters
  - high intra-class similarity: cohesive within clusters
  - low inter-class similarity: distinctive between clusters
- The quality of a clustering method depends on
  - the similarity measure used by the method
  - its implementation, and
  - its ability to discover some or all of the hidden patterns

Clustering의 품질 측정

- Dissimilarity/Similarity metric
  - Similarity is expressed in terms of a distance function, typically metric: d(i, j)
  - The definitions of distance functions are usually rather different for interval-scaled, boolean, categorical, ordinal ratio, and vector variables
  - Weights should be associated with different variables based on applications and data semantics
- Quality of clustering:
  - There is usually a separate "quality" function that measures the "goodness" of a cluster.
  - It is hard to define "similar enough" or "good enough"
    - The answer is typically highly subjective

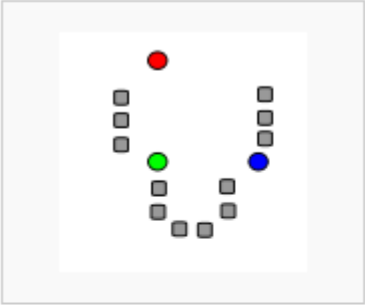
Clustering 시 유의 사항

- Partitioning criteria
  - Single level vs. hierarchical partitioning (often, multi-level hierarchical partitioning is desirable)
- Separation of clusters
  - Exclusive (e.g., one customer belongs to only one region) vs. non-exclusive (e.g., one document may belong to more than one class)
- Similarity measure
  - Distance-based (e.g., Euclidian, road network, vector) vs. connectivity-based (e.g., density or contiguity)
- Clustering space
  - Full space (often when low dimensional) vs. subspaces (often in high-dimensional clustering)

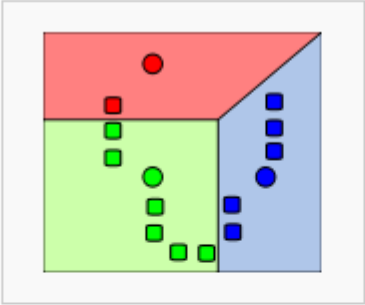
Contents

- 군집화의 개념
- Partitioning Clustering
- Hierarchical Clustering

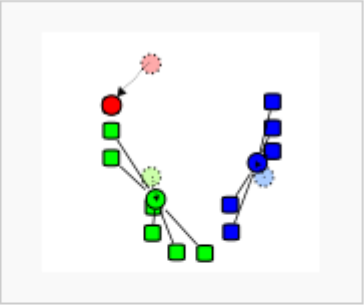
K-means clustering



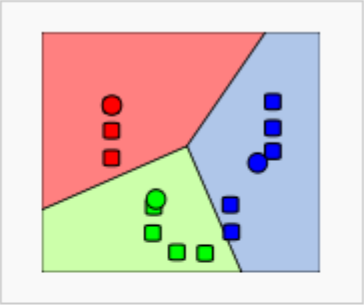
1. *k* initial "means" (in this case *k*=3) are randomly generated within the data domain (shown in color).



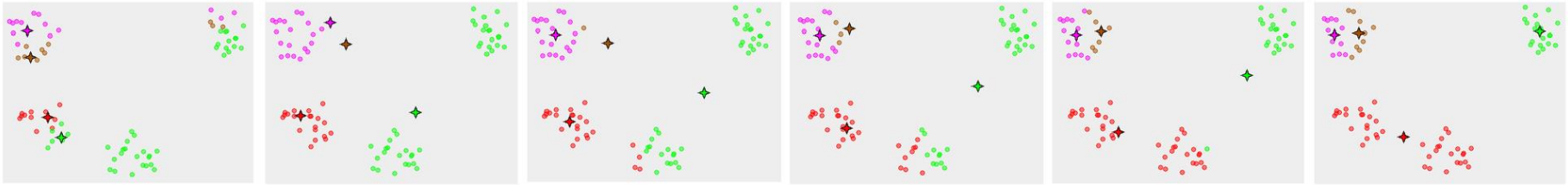
2. *k* clusters are created by associating every observation with the nearest mean. The partitions here represent the Voronoi diagram generated by the means.



3. The centroid of each of the *k* clusters becomes the new mean.



4. Steps 2 and 3 are repeated until convergence has been reached.

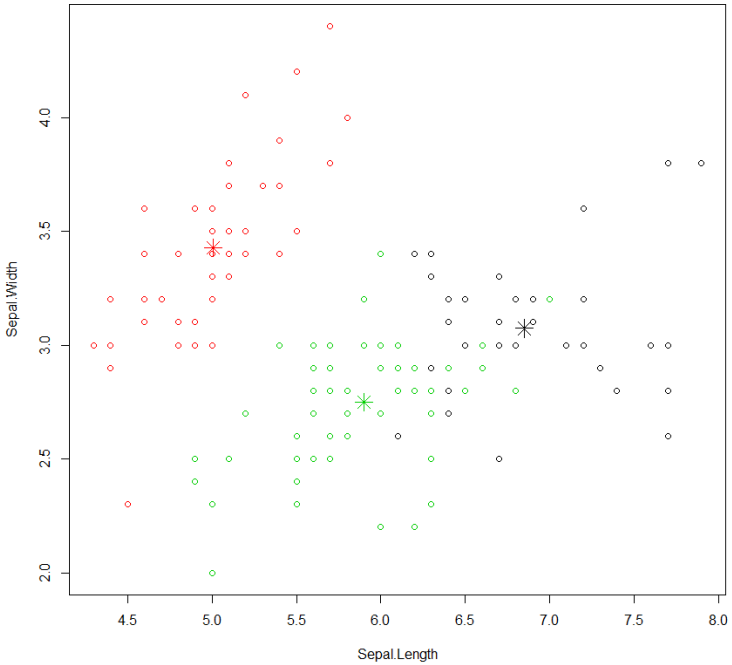


k-means clustering

```
> set.seed(8953)
> iris2 <- iris
> iris2$Species <- NULL
> (kmeans.result <- kmeans(iris2, 3))
K-means clustering with 3 clusters of sizes 38, 50, 62

Cluster means:
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1     6.850000    3.073684     5.742105    2.071053
2     5.006000    3.428000     1.462000    0.246000
3     5.901613    2.748387     4.393548    1.433871

Clustering vector:
  [1] 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 2 2 2 2
 [42] 2 2 2 2 2 2 2 2 2 3 3 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 3 3 3 3
 [83] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 3 1 1 1 1 3 1 1 1 1 1 3 3 1 1 1 1 3 1 3 1
[124] 3 1 1 3 3 1 1 1 1 1 3 1 1 1 1 3 1 1 1 3 1 1 1 3 1 1 3

Within cluster sum of squares by cluster:
[1] 23.87947 15.15100 39.82097
 (between_SS / total_SS =  88.4 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"         "iter"         "ifault"
```

Results of k-means clustering

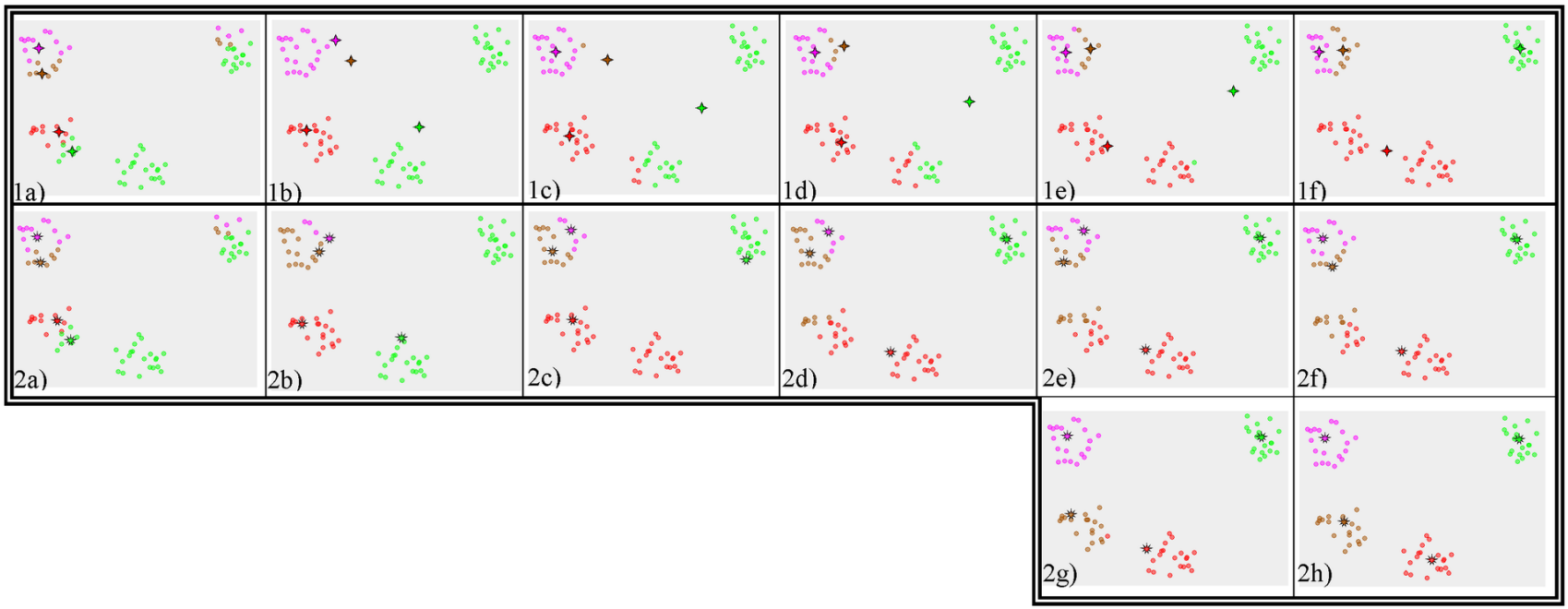
```
> table(iris$Species, kmeans.result$cluster)

             1  2  3
  setosa     0 50  0
  versicolor  2  0 48
  virginica  36  0 14
> plot(iris2[c("Sepal.Length", "Sepal.Width")], col=kmeans.result$cluster)
> points(kmeans.result$centers[,c("Sepal.Length", "Sepal.Width")], col = 1:3, pch = 8, cex
= 2)
```



k-Medoids Clustering

- Difference from k-means: a cluster is represented with its center in the k-means algorithm, but with the object closest to the center of the cluster in the k-medoids clustering.
- more robust than k-means in presence of outliers
- PAM (Partitioning Around Medoids) is a classic algorithm for k-medoids clustering.
- The CLARA algorithm is an enhanced technique of PAM by drawing multiple samples of data, applying PAM on each sample and then returning the best clustering. It performs better than PAM on larger data.
- Functions pam() and clara() in package *cluster*
- Function pamk() in package *fpc* does not require a user to choose k.

k-medoids vs. k-means

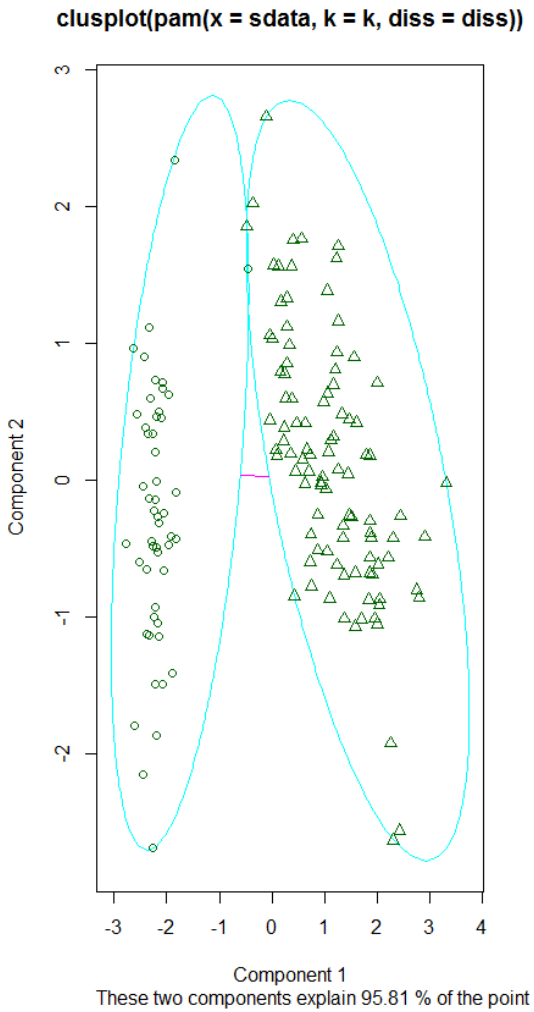


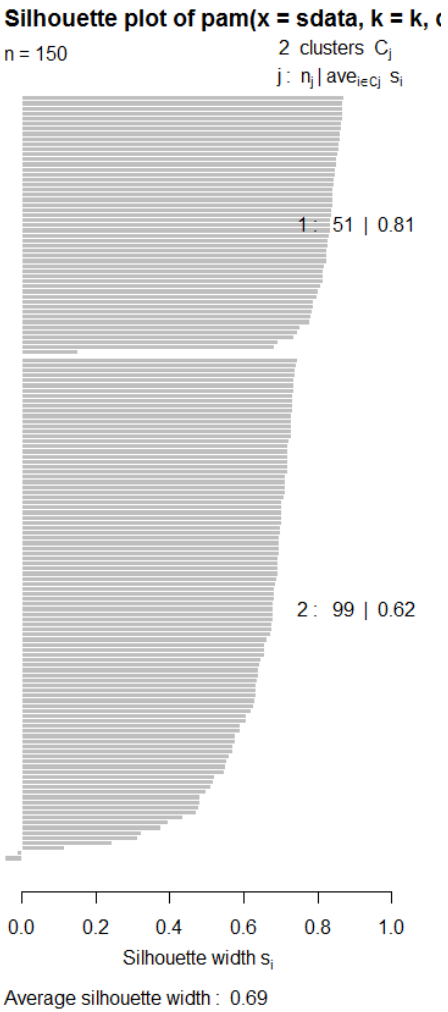
Clustering with pamk()

```
> library(fpc)
Error in nchar(homeDir) : invalid multibyte string, element 1
> pamk.result <- pamk(iris2)
>
> pamk.result$nc
[1] 2
>
> table(pamk.result$pamobject$clustering, iris$Species)

    setosa versicolor virginica
  1     50          1         0
  2      0         49        50
```

Results of pamk()

```
> layout(matrix(c(1,2), 1, 2))
> plot(pamk.result$pamobject)
>
> layout(matrix(1))
```



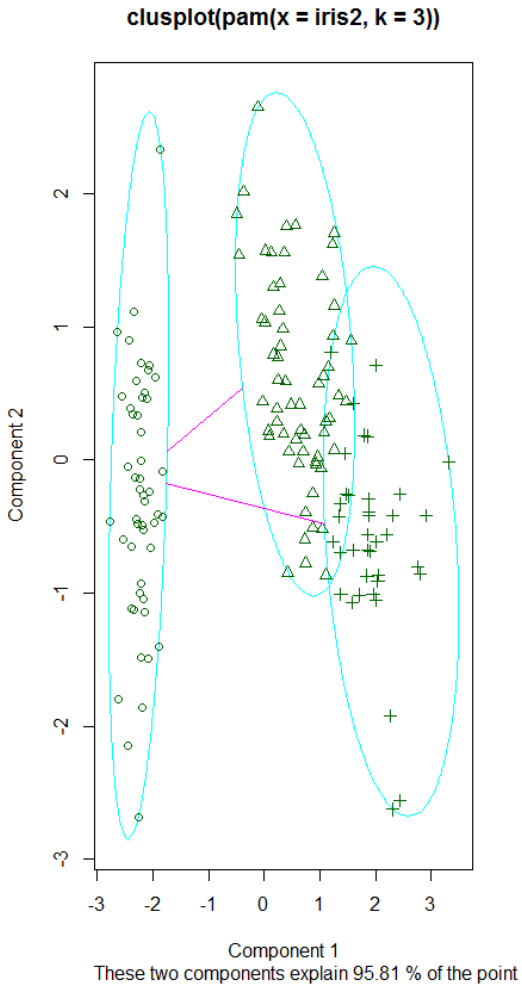


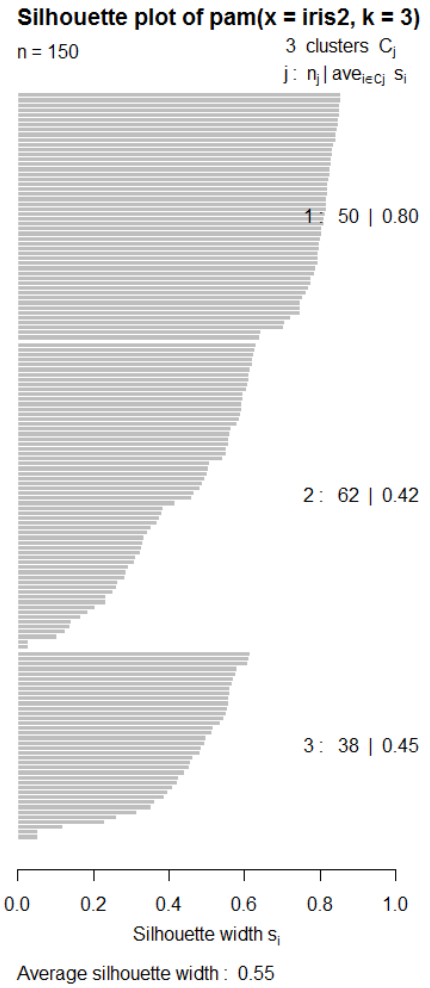
Clustering with pam()

```
> library(cluster)
>
> pam.result <- pam(iris2, 3)
> table(pam.result$clustering, iris$Species)

    setosa versicolor virginica
1      50          0         0
2       0         48        14
3       0          2        36
```

Results of pam()

```
> layout(matrix(c(1,2), 1, 2))
> plot(pam.result)
> layout(matrix(1))
```

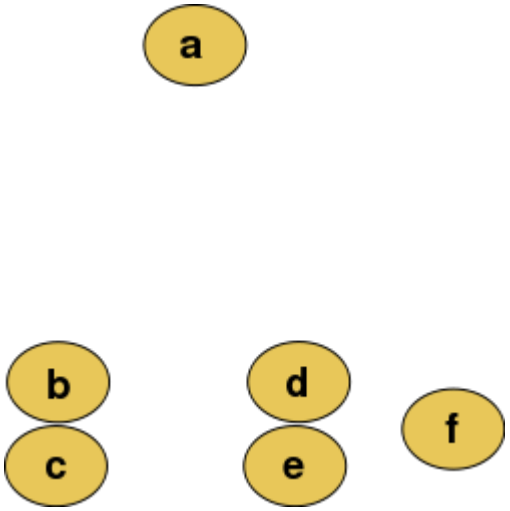




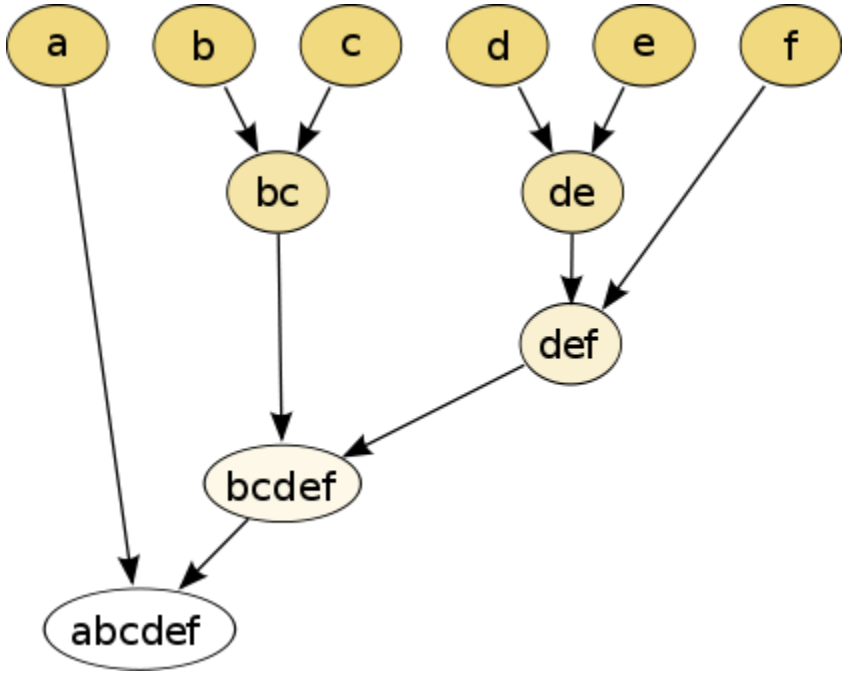
Contents

- 군집화의 개념
- Partitioning Clustering
- Hierarchical Clustering

Hierarchical clustering



Raw data



Hierarchical clustering dendrogram

Hierarchical Clustering

```
> set.seed(2835)
>
> idx <- sample(1:dim(iris)[1], 40)
> irisSample <- iris[idx,]
>
> irisSample$Species <- NULL
>
> hc <- hclust(dist(irisSample), method = 'ave')
>
> hc

Call:
hclust(d = dist(irisSample), method = "ave")

Cluster method   : average
Distance         : euclidean
Number of objects: 40
```

Result of Hierarchical clustering

```
> plot(hc, hang = -1, labels = iris$Species[idx])
>
> rect.hclust(hc, k = 3)
>
> groups <- cutree(hc, k =3)
```