R 프로그래밍

Chapter 12

Clustering, Classification

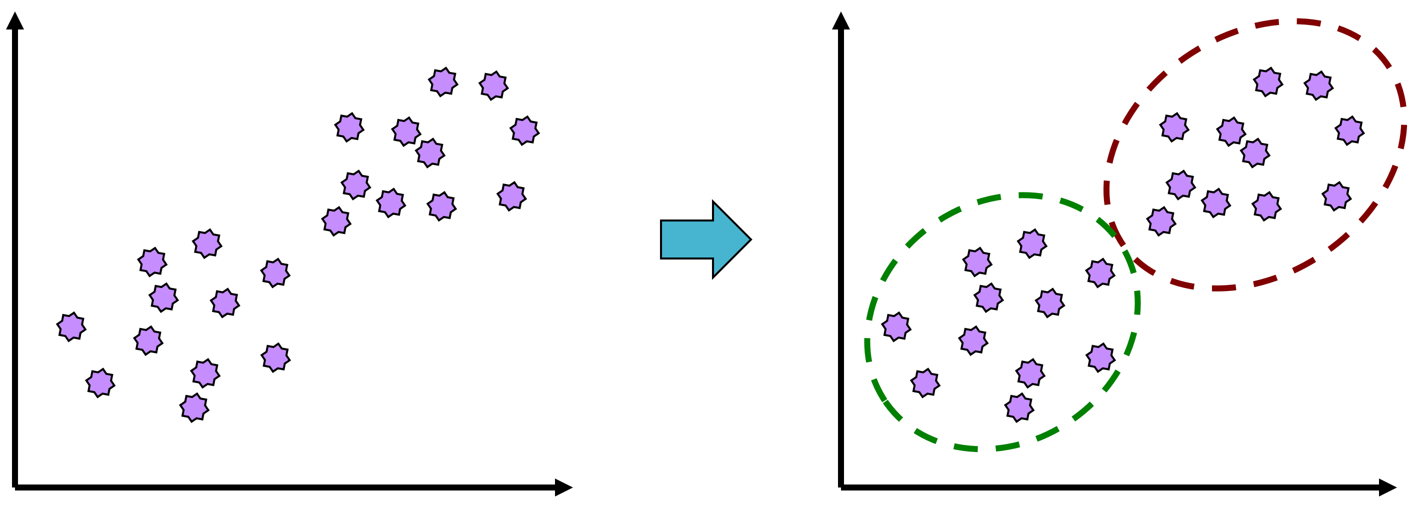
Sejong Oh
Bio Information technology Lab.

Contents

- Summary
- kmeans algorithm
- KNN algorithm
- Exercise

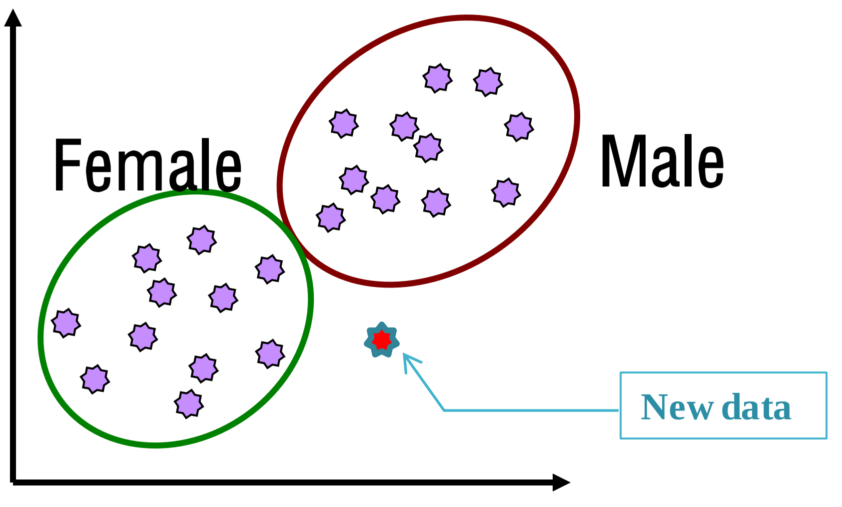
Summary

- Clustering
  - Grouping target data into some category
  - Data in same group has similar characteristics
  - Group points into clusters based on how “near” they are to one another
  - Unsupervised learning



Summary

- Classification
  - Classify new data into one of known category.
  - The category has "label"
  - Supervised learning



Summary

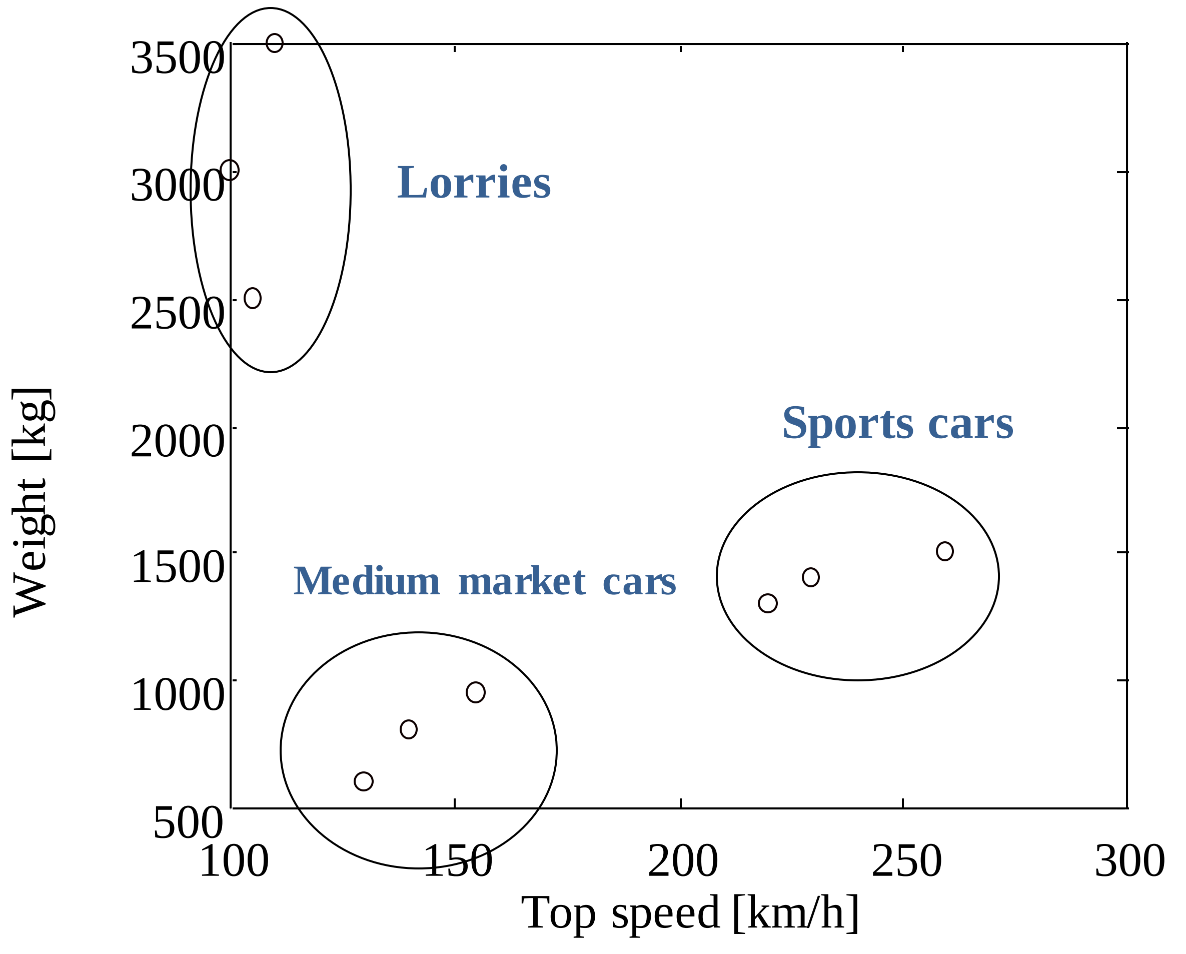
- Clustering example
  - 차량의 특성을 가지고 grouping 을 해 보자

Could see any group ?

Vehicle	Top speed km/h	Color	Air resistance	Weight Kg
V1	220	red	0.30	1300
V2	230	black	0.32	1400
V3	260	red	0.29	1500
V4	140	gray	0.35	800
V5	155	blue	0.33	950
V6	130	white	0.40	600
V7	100	black	0.50	3000
V8	105	red	0.60	2500
V9	110	gray	0.55	3500

Summary

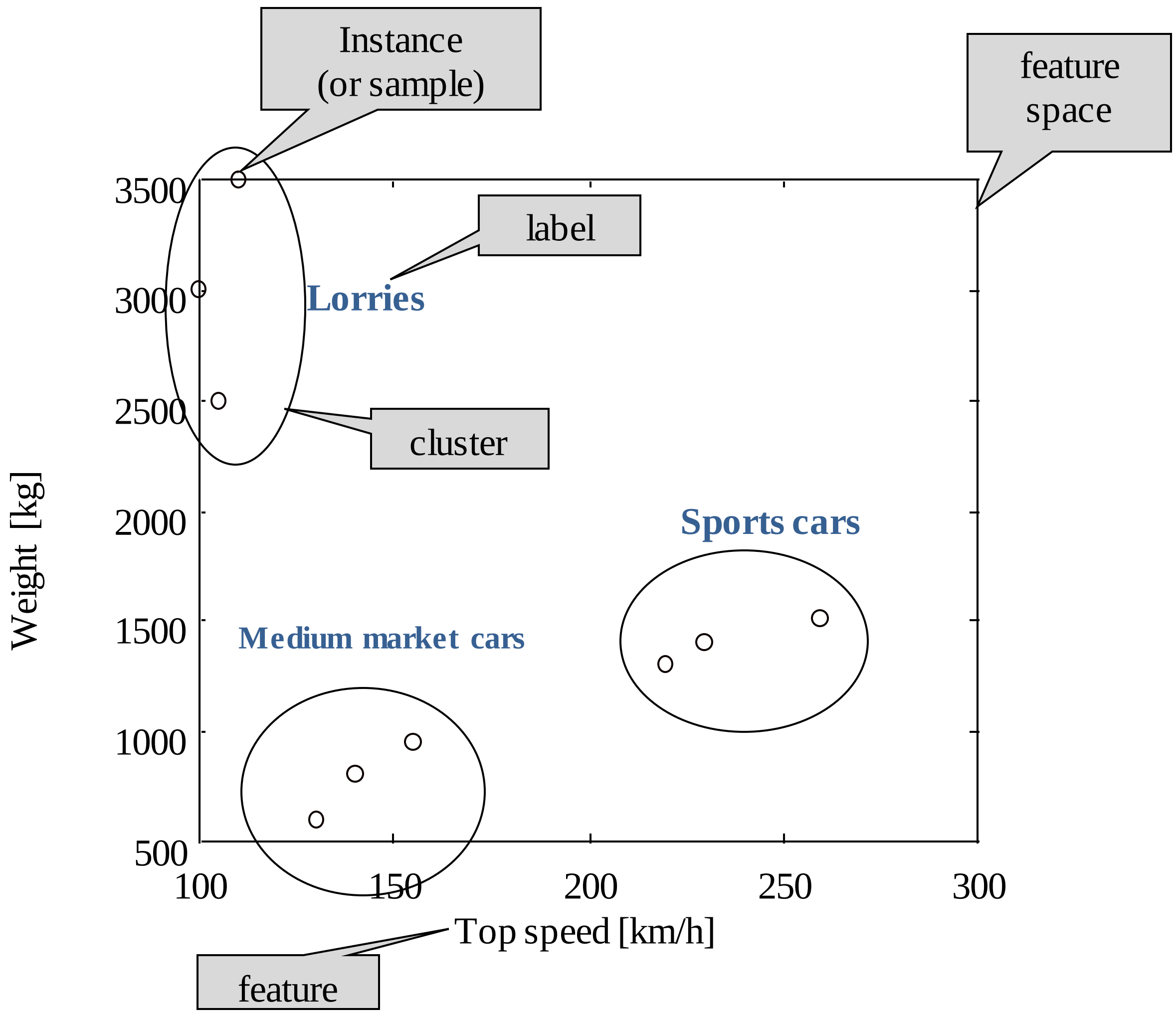
- Clustering example



Summary

- Clustering example

Terminology



Summary

- Classification example

No	Height	Weight	running hour	working hour	Category
1	0.41	0.36	0.27	0.65	Patient
2	0.23	0.37	0.34	0.68	patient
3	0.38	0.38	0.46	0.95	patient
4	0.45	0.31	0.37	0.75	patient
5	0.37	0.45	0.48	0.75	patient
6	0.28	0.26	0.36	0.86	patient
7	0.66	0.44	0.51	0.98	patient
8	0.55	0.43	0.43	0.91	patient
9	0.23	0.44	0.28	0.78	patient
10	0.41	0.53	0.46	0.86	patient
11	0.65	0.38	0.74	0.51	normal
12	0.89	0.53	0.67	0.46	normal
13	0.58	0.54	0.56	0.43	normal
14	0.78	0.55	0.67	0.34	normal
15	0.89	0.56	0.81	0.56	normal
16	0.65	0.57	0.81	0.43	normal
17	0.75	0.67	0.76	0.35	normal
18	0.46	0.48	0.65	0.42	normal
19	0.89	0.69	0.78	0.23	normal
20	0.78	0.81	0.88	0.26	normal

Disease A

Height	Weight	running hour	working hour
0.5	0.44	0.45	0.61

Patient or Normal ?

Summary

- Classification example

Apple iPhone

(1)take a picture by phone camera

kmeans

(2) Search similar image and shows detail information about it

Classification analysis procedure

1. Prepare target dataset
2. Divide target dataset into training data and test data
   - assume we don't know class labels of test data
3. Training model using training data
4. Predict class labels of test data using learning model
5. Evaluate prediction performance

$$\text{accuracy} = \frac{\text{\# of instances that are correctly predicted}}{\text{\# of total instances in test data}}$$

Summary

- Binary vs. multiple classification

- Binary classification
  - # of class is two

Male	Female
Patient	Normal
Yes	No

- multiple classification
  - # of class over two

Well-done	medium	rare

university	High school
Middle school	Elementary school

Summary

- Binary Classification Error

fact

predict

	Fact is True	Fact is False
Predict as True	TP	FP
Predict as False	FN	TN

TP : true positive FP : false positive
FN : false negative TN : true negative

Summary

- Binary Classification Error

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

Summary

- Binary Classification Error

Sensitivity = TP/(TP+FN)

환자를 환자라고 예측할 확률

Specificity = TN/(TN+FP)

정상인을 정상인이라고 예측할 확률

- Sensitivity
  - Fraction of all Class1 (True) that we correctly predicted at Class 1
  - *How good are we at finding what we are looking for*

- Specificity
  - Fraction of all Class 2 (False) called Class 2
  - *How many of the Class 2 do we filter out of our Class 1 predictions*

In both cases, the higher the better

Summary

- Supervised learning
  - First, training the algorithm, and the algorithm do the task
  - We already have well classified sample
  - Classification
- Unsupervised learning
  - No training. Algorithm do the task by itself
  - Clustering

Clustering algorithm

K-means clustering

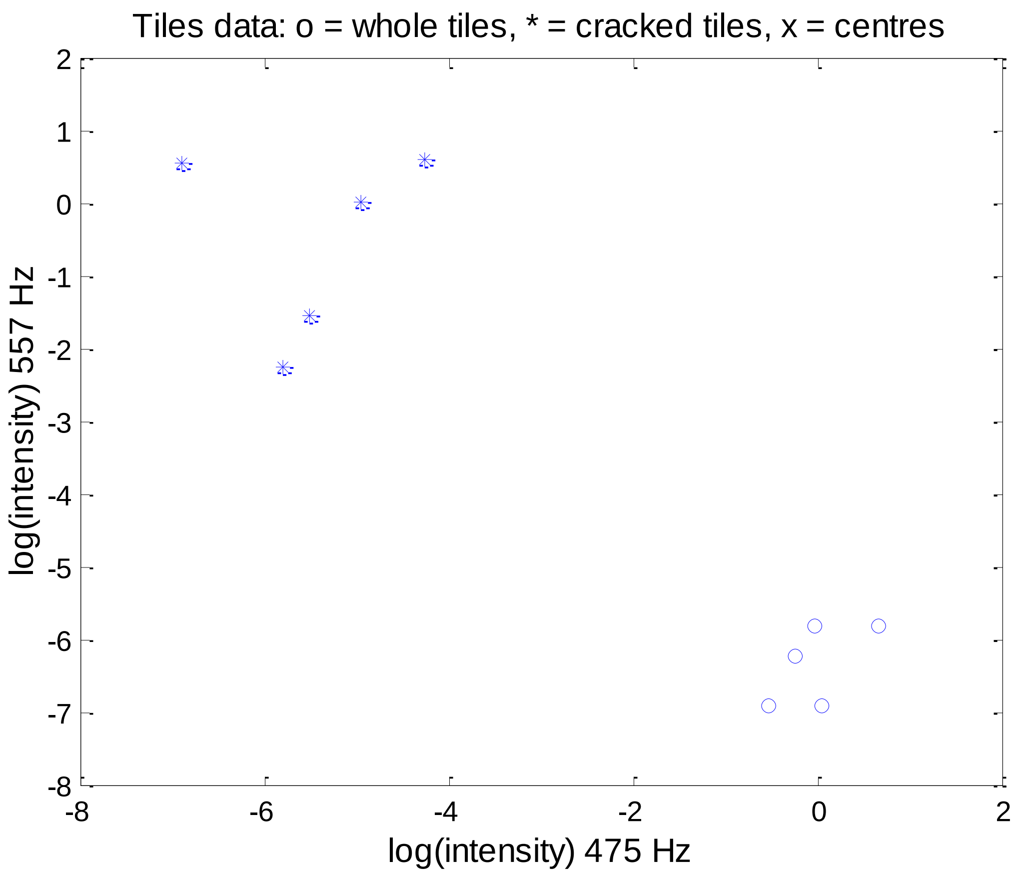
Cracked tiles example

475Hz	557Hz
0.958	0.003
1.043	0.001
1.907	0.003
0.780	0.002
0.579	0.001
0.003	0.105
0.001	1.748
0.014	1.839
0.007	1.021
0.004	0.214

Table 1: frequency intensities for ten tiles.

Tiles are made from clay moulded into the right shape, brushed, glazed, and baked. Unfortunately, the baking may produce invisible cracks. Operators can detect the cracks by hitting the tiles with a hammer, and in an automated system the response is recorded with a microphone, filtered, Fourier transformed, and normalised. A small set of data is given in TABLE 1 (adapted from MIT, 1997).

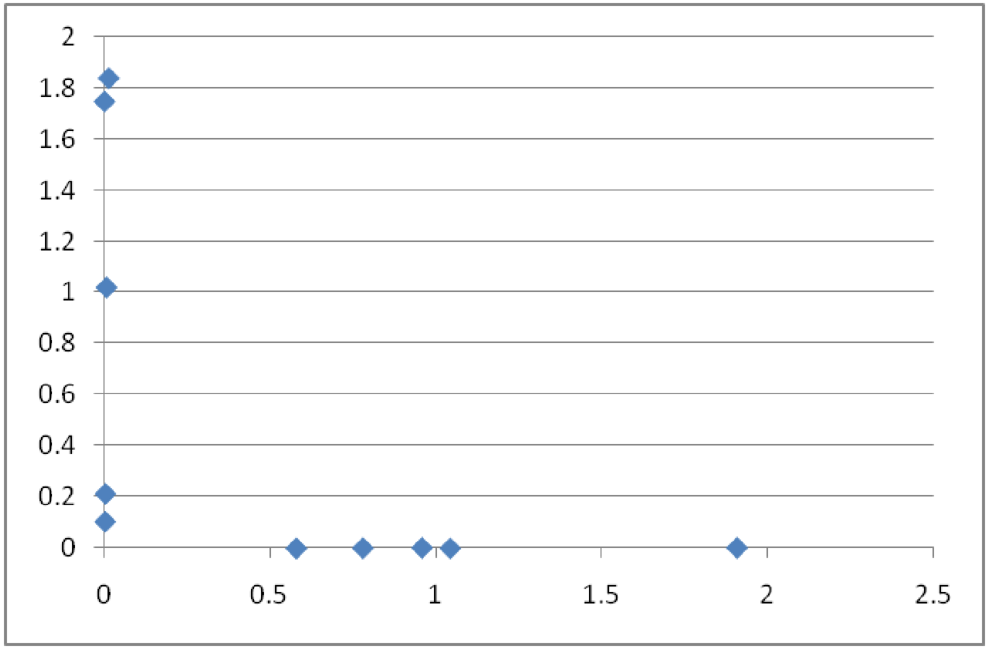
K-means



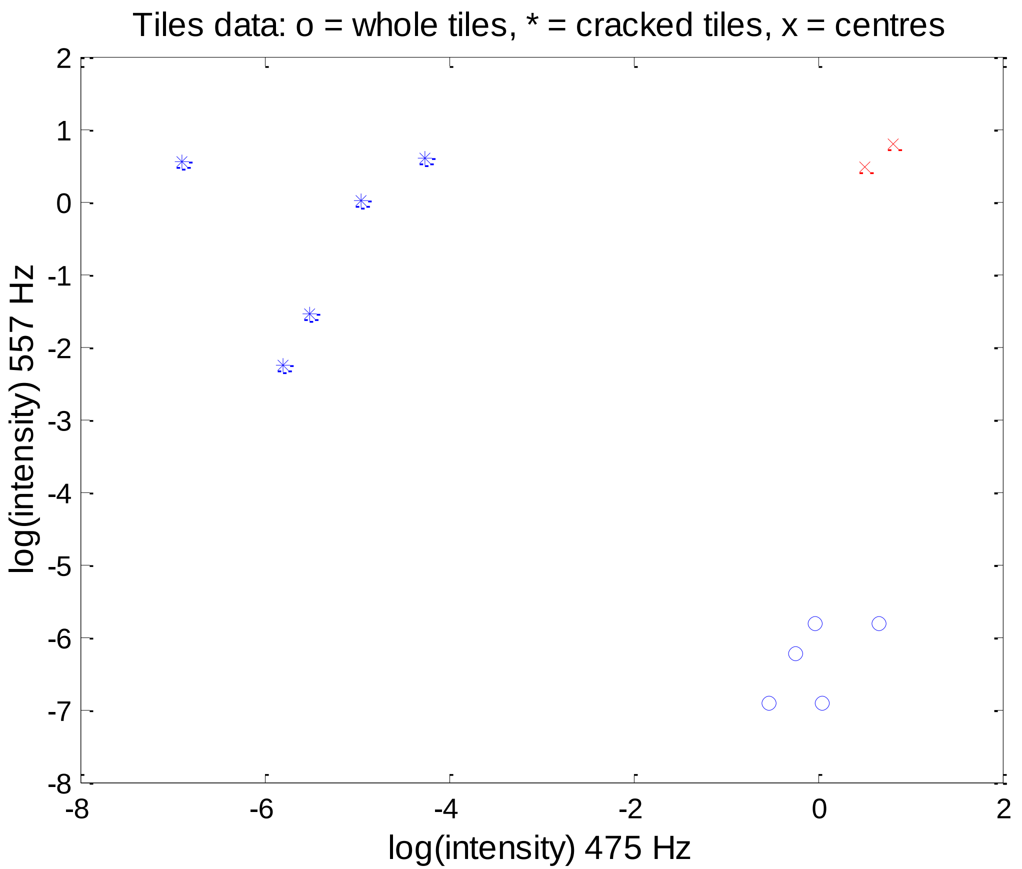
Plot of tiles by frequencies (logarithms). The whole tiles (o) seem well separated from the cracked tiles (*). The **objective** is to find the two clusters.

K-means

- Before logarithms

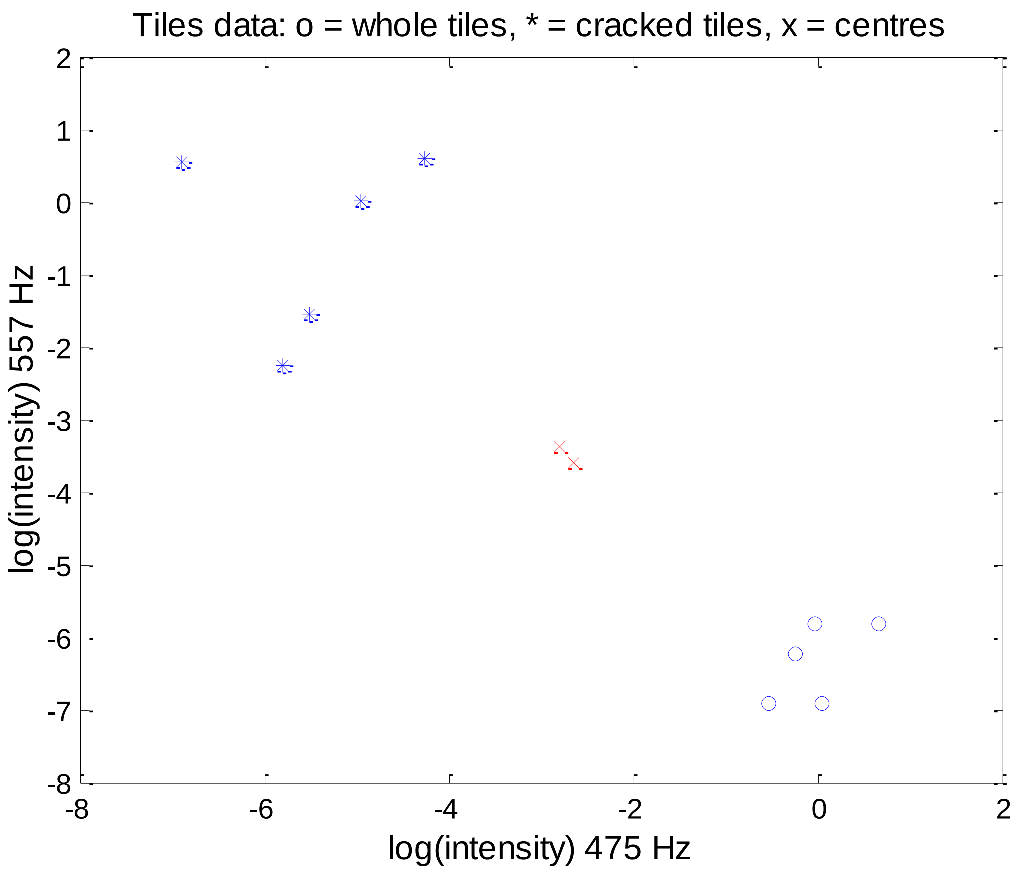


K-means



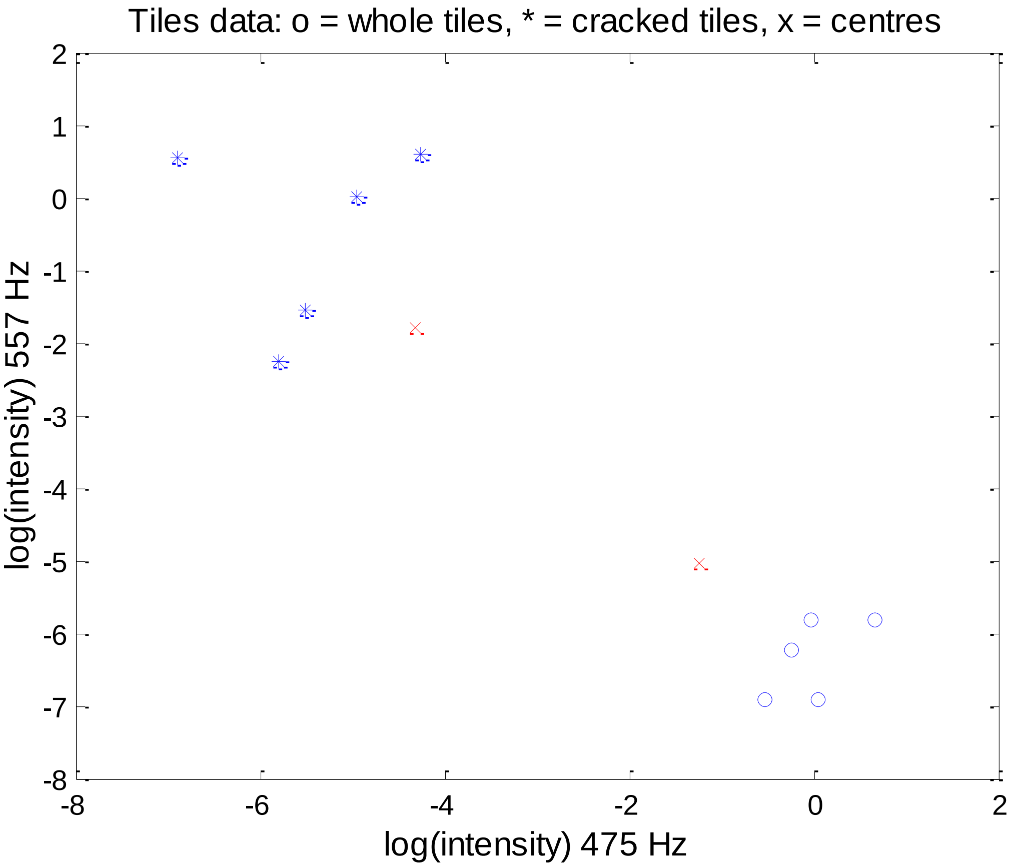
1. Place two cluster centres (x) at random.
2. Assign each data point (* and o) to the nearest cluster centre (x)

K-means



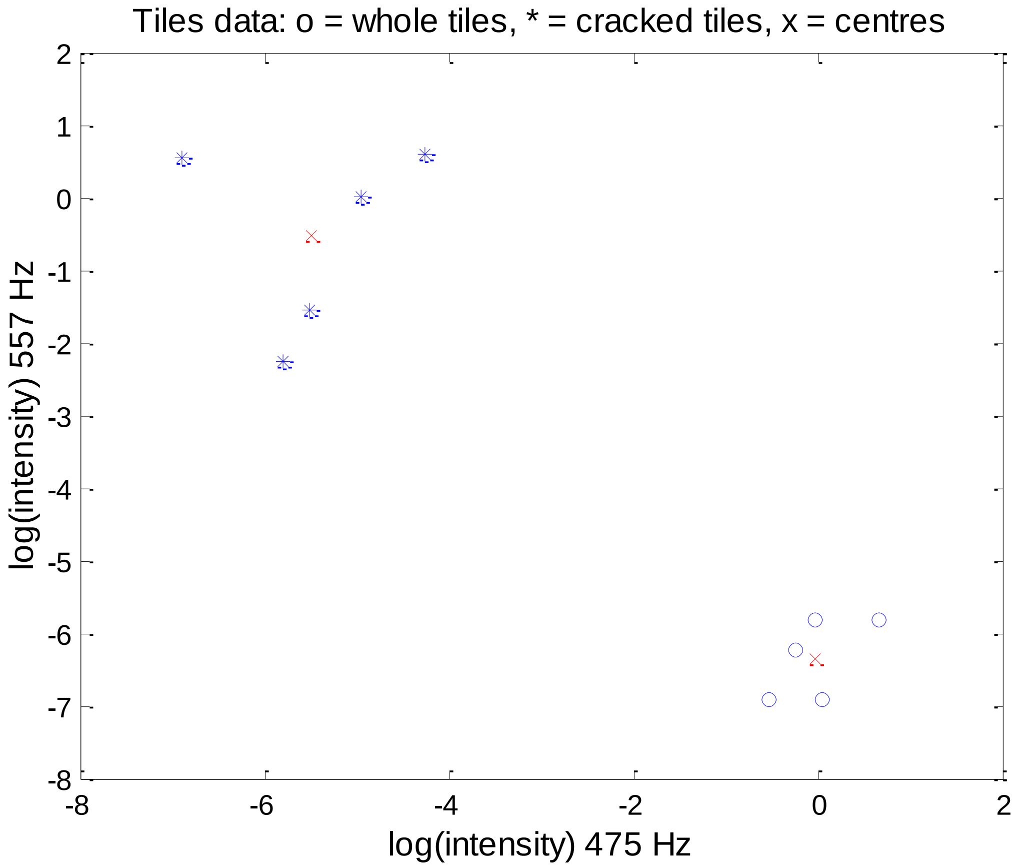
1. Compute the new centre of each class
2. Move the crosses (x)

K-means



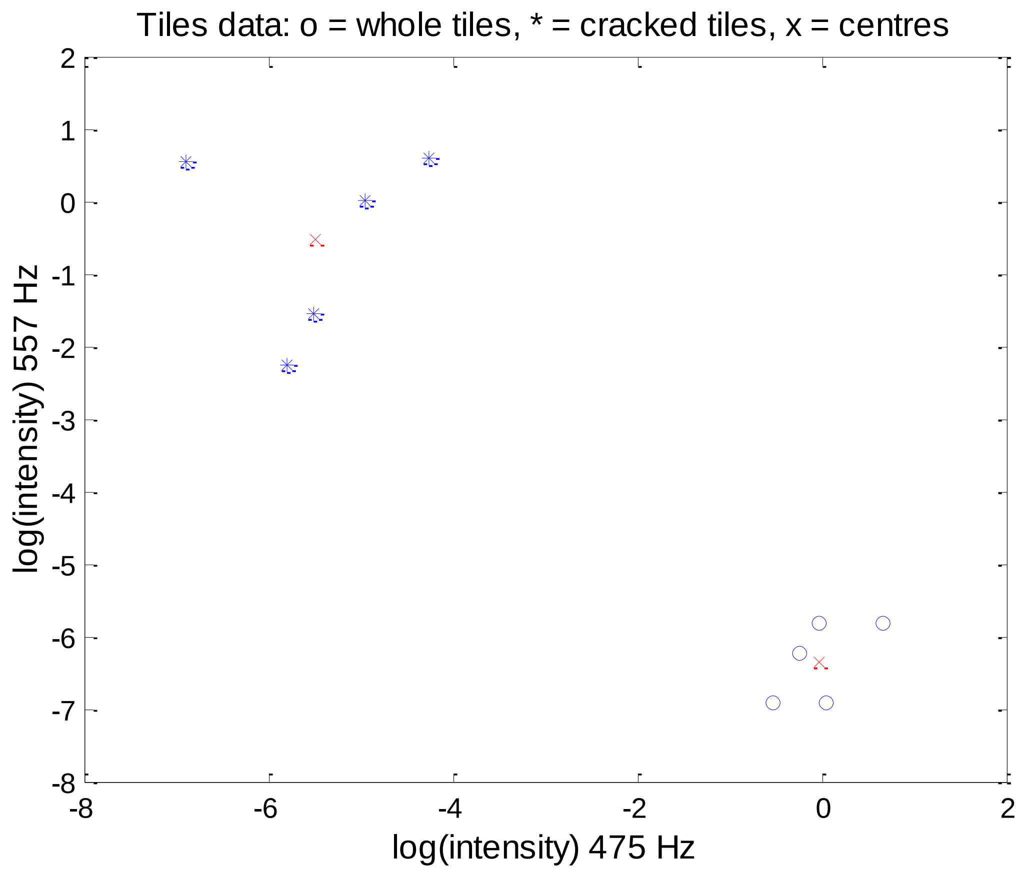
Iteration 2

K-means



Iteration 3

K-means



Iteration 4 (then stop, because no visible change)
Each data point belongs to the cluster defined by the nearest centre

K-means

475Hz	557Hz
0.958	0.003
1.043	0.001
1.907	0.003
0.780	0.002
0.579	0.001
0.003	0.105
0.001	1.748
0.014	1.839
0.007	1.021
0.004	0.214

M =

First cluster	Second cluster
0.0000	1.0000
0.0000	1.0000
0.0000	1.0000
0.0000	1.0000
0.0000	1.0000
1.0000	0.0000
1.0000	0.0000
1.0000	0.0000
1.0000	0.0000
1.0000	0.0000

The membership matrix M:

1. The last five data points (rows) belong to the first cluster (column)
2. The first five data points (rows) belong to the second cluster (column)

Euclidean distance

p= $(p_1, p_2, p_3, \ldots, p_n)$, q = $(q_1, q_2, q_3, \ldots, q_n)$

Euclidean distance

$$\mathrm{d}(\mathbf{p},\mathbf{q}) = \sqrt{(p_1-q_1)^2 + (p_2-q_2)^2 + \cdots + (p_n-q_n)^2} = \sqrt{\sum_{i=1}^{n}(p_i-q_i)^2}.$$

scolar

vector

Euclidean norm measure

$$\|\mathbf{p}\| = \sqrt{p_1^2 + p_2^2 + \cdots + p_n^2} = \sqrt{\mathbf{p}\cdot\mathbf{p}}$$

Distance using Euclidean norm measure

$$\|\mathbf{p}-\mathbf{q}\| = \sqrt{(\mathbf{p}-\mathbf{q})\cdot(\mathbf{p}-\mathbf{q})} = \sqrt{\|\mathbf{p}\|^2 + \|\mathbf{q}\|^2 - 2\mathbf{p}\cdot\mathbf{q}}.$$

$(\mathrm{p} \bullet \mathrm{q} = p_1 q_1 + p_2 q_2 + \ldots + p_n q_n)$

[R 실습] : K-means

- Usage

```
kmeans(x, centers, iter.max = 10, nstart = 1,
algorithm = c("Hartigan-Wong", "Lloyd", "Forgy",
"MacQueen"))
```

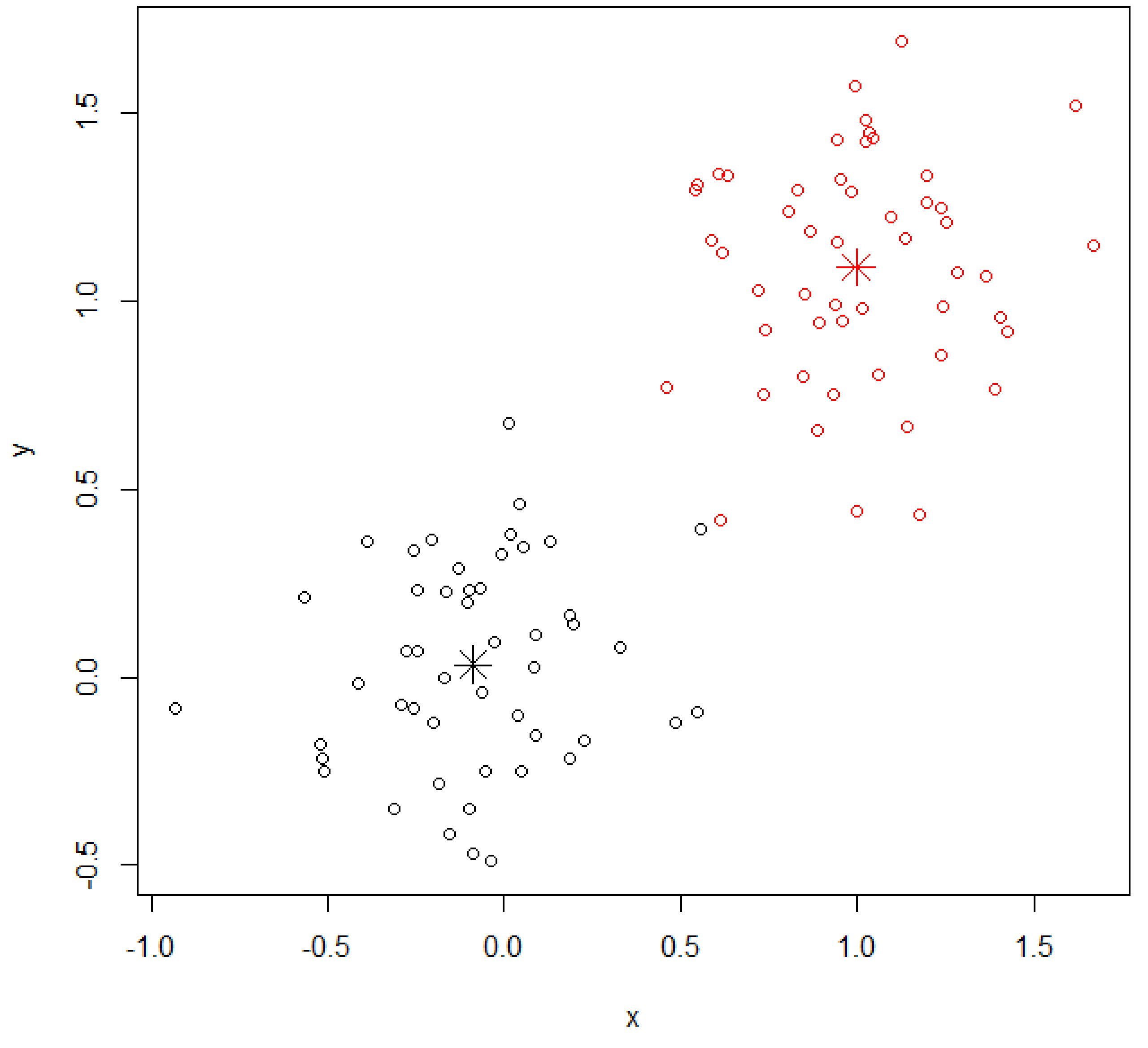
- Argument
  - **x** : numeric matrix of data, or an object that can be coerced to such a matrix (such as a numeric vector or a data frame with all numeric columns).
  - **centers** : either the number of clusters, say *k*, or a set of initial (distinct) cluster centres.
  - **iter.max** : the maximum number of iterations allowed.
  - **nstart** : if centers is a number, how many random sets should be chosen?
  - **algorithm** : character: may be abbreviated.

[R 실습] : K-means

```
require(graphics)
# create a 2-dimensional example
x <- rbind(matrix(rnorm(100, sd = 0.3), ncol = 2),
           matrix(rnorm(100, mean = 1, sd = 0.3),
           ncol = 2))
colnames(x) <- c("x", "y")
cl <- kmeans(x, 2)
cl # show clustering result

plot(x, col = cl$cluster)
points(cl$centers, col = 1:2, pch = 8, cex=2)

# random starts do help here with too many clusters
cl <- kmeans(x, 5, nstart = 25)
plot(x, col = cl$cluster)
points(cl$centers, col = 1:5, pch = 8)
```



1.5
1.0
0.5
0.0
-0.5
y
-1.0
-0.5
0.0
0.5
1.0
1.5
x

[R 실습] : K-means

- Slide 18 에 있는 데이터를 로그 변환하여 새로운 데이터셋을 만든다.
  - 로그변환 y = log(x)

- 새로운 데이터셋을 이용하여 k-means 클러스터링을 실시하여 그 결과를 그래프로 그린다. (cluster수=2)

[과제1]

(1) iris 데이터셋에 대해 kmeans 클러스터링을 하고 결과를 품종정보와 비교하여 비이시오

- Iris 데이터셋에서 품종(Species) 컬럼은 제외하시오
- 클러스터 수는 3 으로 하시오

(2) state.x77 데이터셋에 대해 kmeans 클러스터링을 하고 각 클러스터 별로 주(state)의 이름을 보이시오

- 클러스터 수는 5로 하시오
- State.x77 은 각 컬럼의 값들의 단위가 많이 차이가 나기 때문에 이를 적절히 맞추어줄 필요가 있다.

```
new.data = scale(state.x77)
```

(3) cars 데이터셋에 대해 kmeans 클러스터링을 하고 결과를 산점도로 보이시오

- 클러스터 수는 3 으로 하시오

Classification algorithm

KNN

Problem Definition

No	running hour	working hour	Category
1	0.27	0.65	Patient
2	0.34	0.68	patient
3	0.46	0.95	patient
4	0.37	0.75	patient
5	0.48	0.75	patient
6	0.36	0.86	patient
7	0.51	0.98	patient
8	0.43	0.91	patient
9	0.28	0.78	patient
10	0.46	0.86	patient
11	0.74	0.51	normal
12	0.67	0.46	normal
13	0.56	0.43	normal
14	0.67	0.34	normal
15	0.81	0.56	normal
16	0.81	0.43	normal
17	0.76	0.35	normal
18	0.65	0.42	normal
19	0.78	0.23	normal
20	0.88	0.26	normal

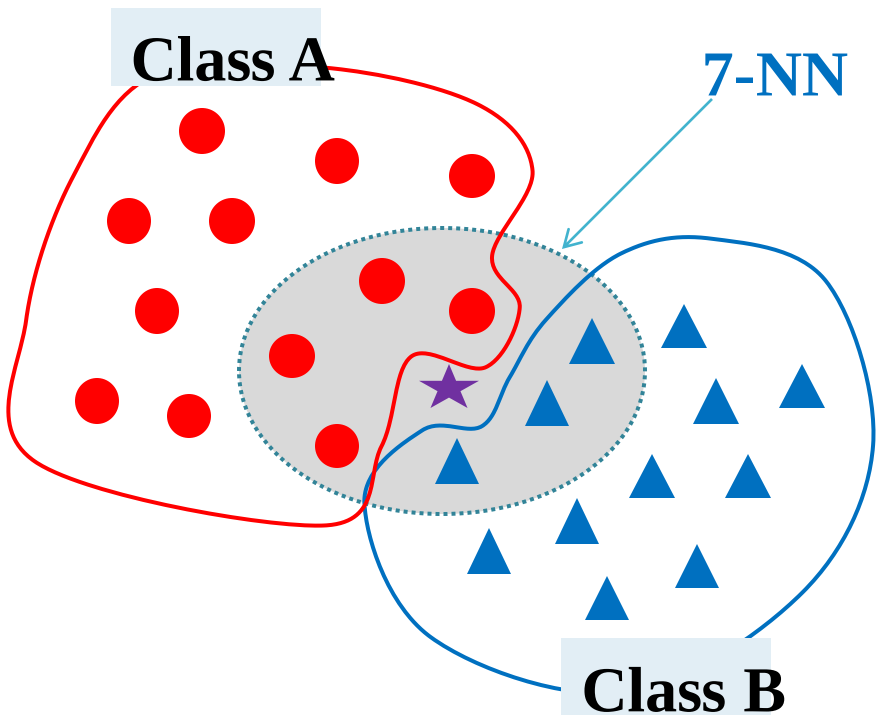
Given Classified Data

Patient or Normal ?

running hour	working hour
0.45	0.61

Idea of KNN

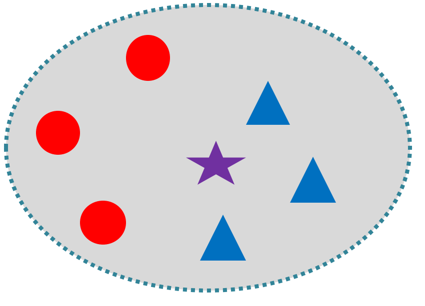
- Find K nearest neighbor for new point (★)
- Decide new point belongs to major class (class A)
  - # of neighbor of Class A > # of neighbor of Class B



Idea of KNN

- Algorithm
  - Calculate distance between new point and every point of given classes
  - Choose K nearest points by the distance
  - Choose major class from K points
    (the class is for the new point)

6-NN



???

Idea of KNN

- How to calculate the distance between two element ?
  - Using Euclidean distance

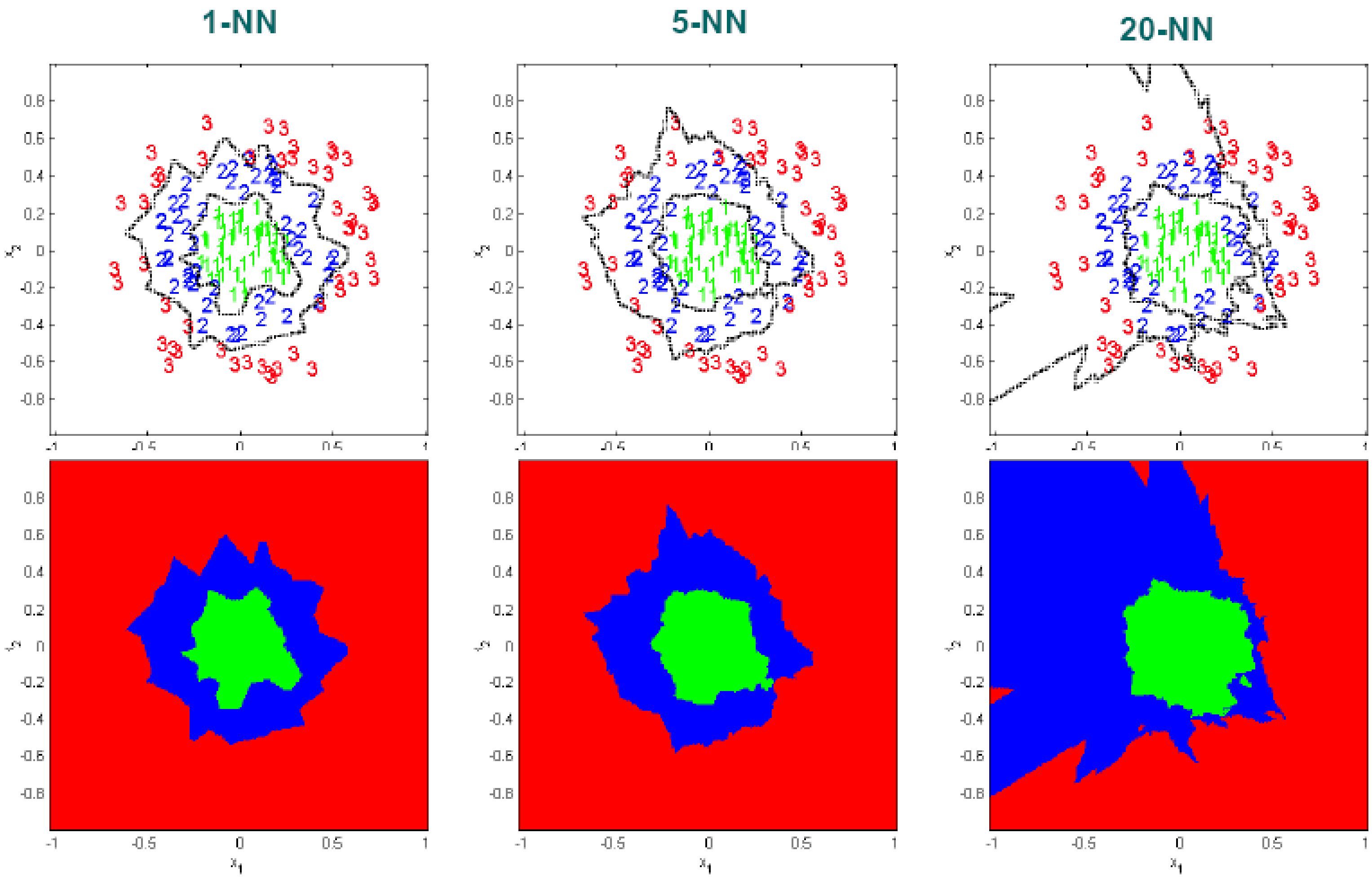
$$\mathbf{p} = (p_1, p_2, \ldots, p_n)$$
$$\mathbf{q} = (q_1, q_2, \ldots, q_n)$$

$$\mathrm{d}(\mathbf{p}, \mathbf{q}) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}.$$

1-NN vs. k-NN

- The use of large values of k has two main advantages
  - Yields smoother decision regions
  - Provides probabilistic information
    - The ratio of examples for each class gives information about the ambiguity of the decision
- However, too large a value of k is detrimental
  - It destroys the locality of the estimation since farther examples are taken into account
  - In addition, it increases the computational burden
- A good rule-of-thumb numbers is k should be less than the square root of the total number of training patterns.

1NN versus kNN



Discussion

- Advantage
  - Nonparametric architecture
  - Simple
  - Powerful
  - Requires no training time
- Disadvantage
  - Memory intensive
  - Classification/estimation is slow

[R실습] KNN

- Usage

```
knn(train, test, cl, k = 1, l = 0,
    prob = FALSE, use.all = TRUE)
```

- Parameters
  - **train** : matrix or data frame of training set cases.
  - **test** : matrix or data frame of test set cases.
  - **cl**: factor of true classifications of training set
  - **k** : number of neighbours considered.
  - **l** : minimum vote for definite decision, otherwise doubt.
  - **prob**: If this is true, the proportion of the votes for the winning class are returned as attribute prob.
  - **use.all:** controls handling of ties. If true, all distances equal to the kth largest are included. If false, a random selection of distances equal to the kth is chosen to use exactly k neighbours

[R실습] KNN

```r
require("class")

# prepare train/test data
tr.idx <- c(1:25,51:75, 101:125)
ds.tr <- iris[tr.idx, 1:4]
ds.ts <- iris[-tr.idx, 1:4]
cl.tr <- factor(iris[tr.idx, 5])
cl.ts <- factor(iris[-tr.idx, 5])

pred <- knn(ds.tr, ds.ts, cl.tr, k = 3, prob=TRUE)
pred

acc = mean(pred==cl.ts) # 예측 정확도
acc
```

- require = library
- Knn 을 이용하려면 “class” 라이브러리 필요

[R실습] KNN

```
table(pred,cl.ts)
```

```
> table(pred,cl.ts)
            cl.ts
pred         setosa versicolor virginica
  setosa         25          0         0
  versicolor      0         23         3
  virginica       0          2        22
```

[과제 2] KNN

- 다음의 데이터셋을 이용하여 KNN 알고리즘을 테스트하시오
- Target dataset : Breast Cancer Wisconsin (Diagnostic) Data Set
  - http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/wdbc.data
  - wdbc.csv 파일에 저장후 프로그램에서 읽어들인다
  - 첫번째 컬럼 : instance ID (삭제한다)
  - 두번째 컬럼 : class 정보 (M,B)

- 홀수번째 instance는 training data 로, 짝수번째 instance는 test data 로 이용한다
- K = 3,5,7 로 하여 accuracy 를 비교한다.

K-fold Cross Validation

- Only one classification experiment is enough ?

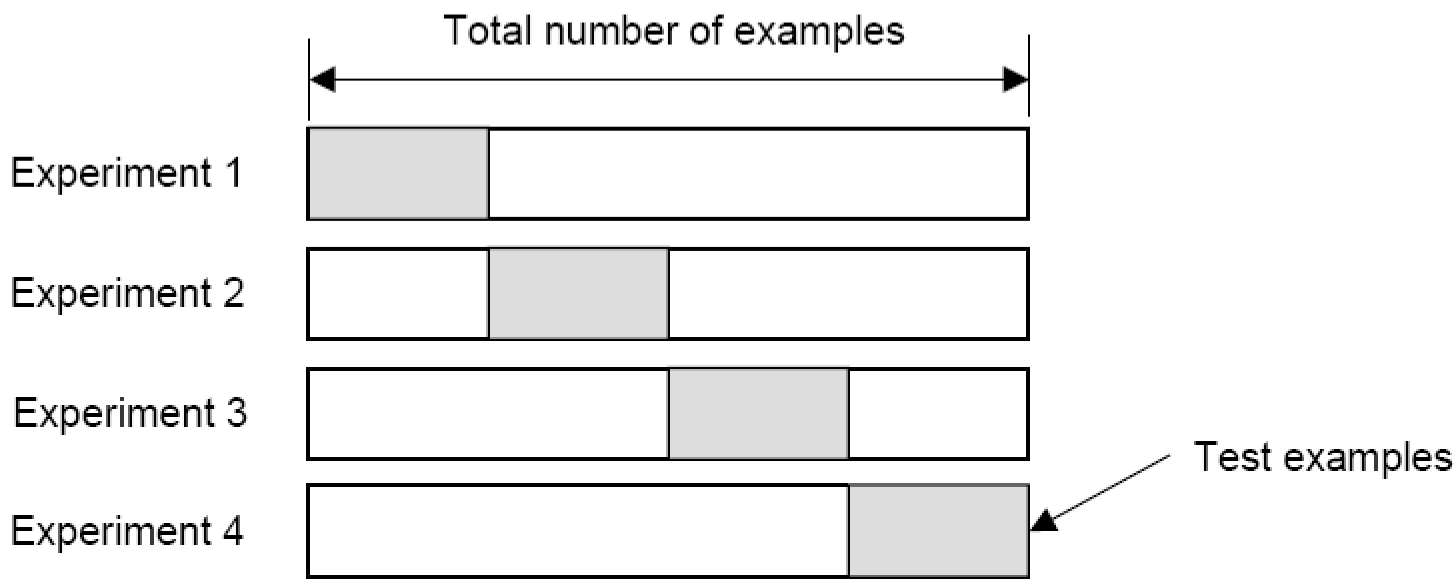
Training data	Test data

  - Classification accuracy = 0.87 (???)

- I may be produced by chance
- If we choose different training/test data, then …

K-fold Cross Validation

- Create a K-fold partition of the dataset
  - For each of K experiments, use K-1 folds for training and the remaining one for testing



  - the true error is estimated as the average error rate

$$E = \frac{1}{K}\sum_{i=1}^{K} E_i$$

K-fold Cross Validation

- 3-fold cross validation
  - Collect test examples from all classes by even rate (33%) of samples in the classes