Ch.3 Clustering Analysis

<어서와~ 머신러닝은 처음이지>

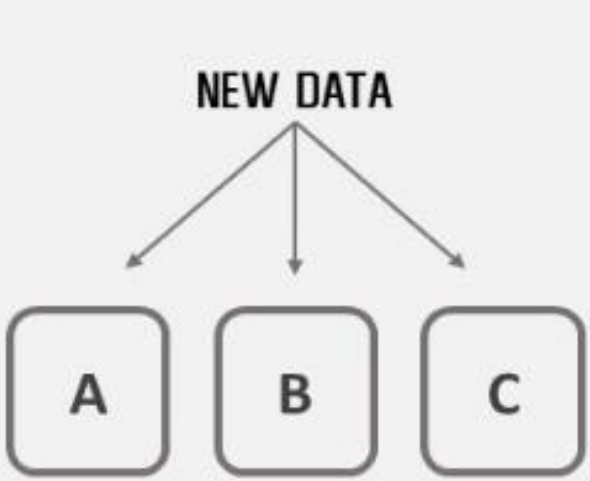
- 거리행렬 기반의 비지도 학습
- Hierarchical Clustering
- K-means Clustering
- DBSCAN(Density-Based Spatial Clustering of Applications with Noise)

장준규

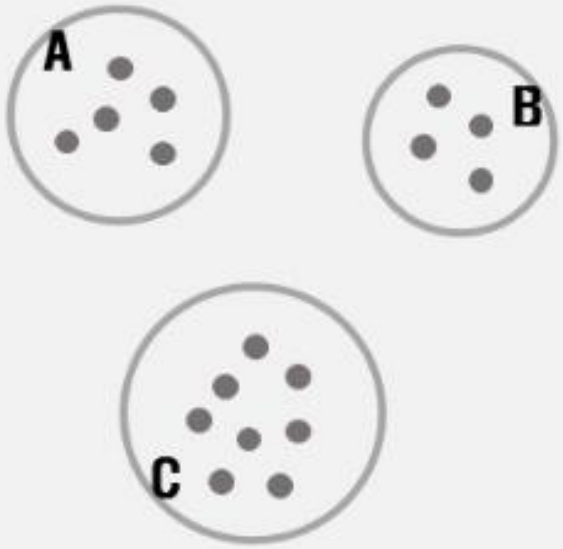
2017. 04. 24

Before We Start

분류(Classfication) VS 군집화(Clustering)



SUPERVISED LEARNING



UNSUPERVISED LEARNING

source: http://www.datamarket.kr/xe/board_BoGi29/9880

1. Hierarchical Clustering

- **가장 가까운 데이터끼리 먼저 연결하여 트리구조로 군집화**
- **군집간의 거리 계산 방법**
  - **최단연결**
  - 최장연결
  - 평균연결
  - 중심연결
  - 워드연결

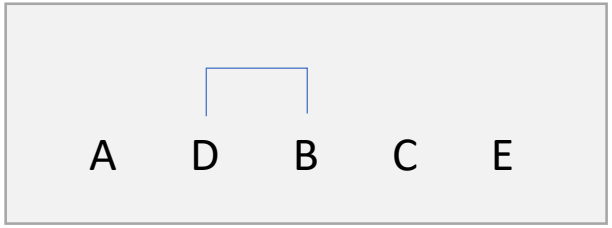
$$d_{(AB)C} = \min(d_{(AC)}, d_{(BC)})$$

(데이터 A와 군집 C와의 거리, 데이터 B와 군집 C와의 거리) 중 최소값을 선택한다.

- **Step1. 학생 5명에 대한 거리행렬**

	A	B	C	D	E
A	9999.00000	48.56954	63.85922	46.50806	33.30165
B	48.56954	9999.00000	61.29437	17.32051	32.64966
C	63.85922	61.29437	9999.00000	70.23532	78.05767
D	46.50806	17.32051	70.23532	9999.00000	31.52777
E	33.30165	32.64966	78.05767	31.52777	9999.00000

→ BD 군집 형성
→ B,D 삭제

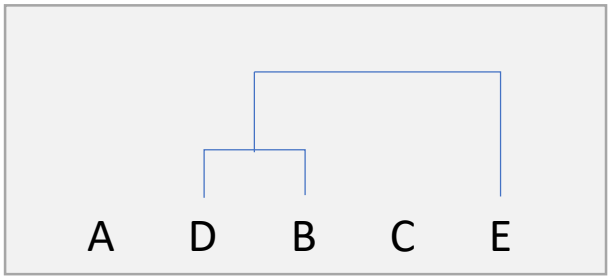


1. Hierarchical Clustering

- **Step2. BD군집 간의 거리행렬**

	A	C	E	BD
A	9999.00000	63.85922	33.30165	46.50806
C	63.85922	9999.00000	78.05767	61.29437
E	33.30165	78.05767	9999.00000	31.52777
BD	46.50806	61.29437	31.52777	9999.00000

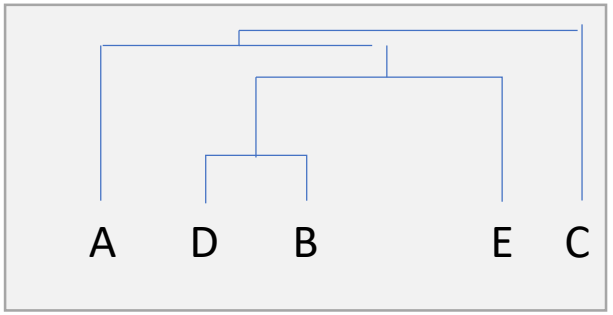
→ BDE군집 형성
→ E 삭제



- **Step3. BDE군집 간의 거리행렬**

	A	C	BDE
A	9999.00000	63.85922	33.30165
C	63.85922	9999.00000	61.29437
BDE	33.30165	61.29437	9999.00000

→ BDEA군집 형성
→ A 삭제
→ BDEA 군집 간 거리행렬
→ BDEAC 덴드로그램 완성

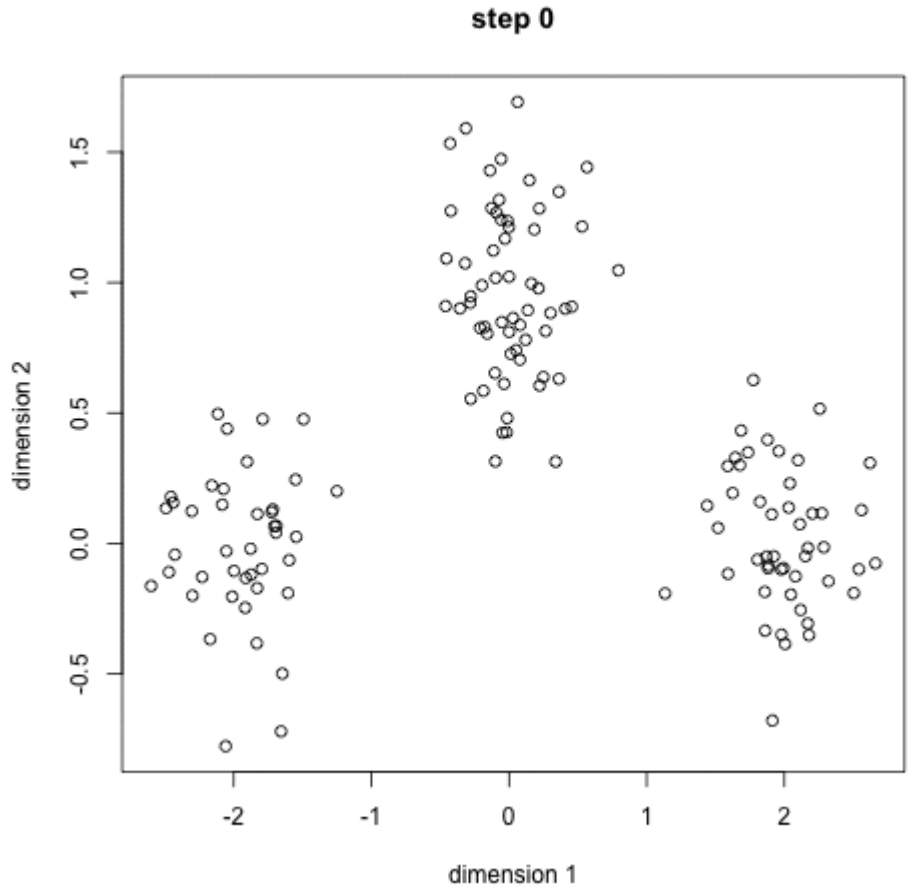


***R의 hclust 함수 제공**

2. K-means Clustering

- κ개의 중심에서 가장 가까운 데이터를 군집화

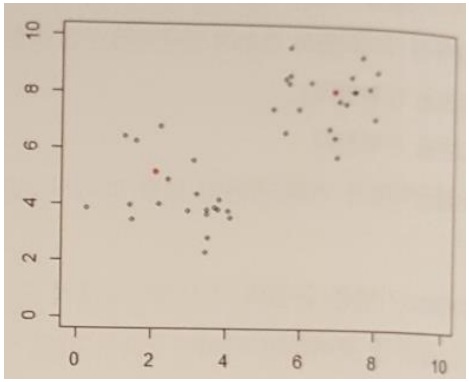
각각의 k개의 평균 vector로부터 해당군집의 분산이 최소가 되는 군집을 찾는 기법

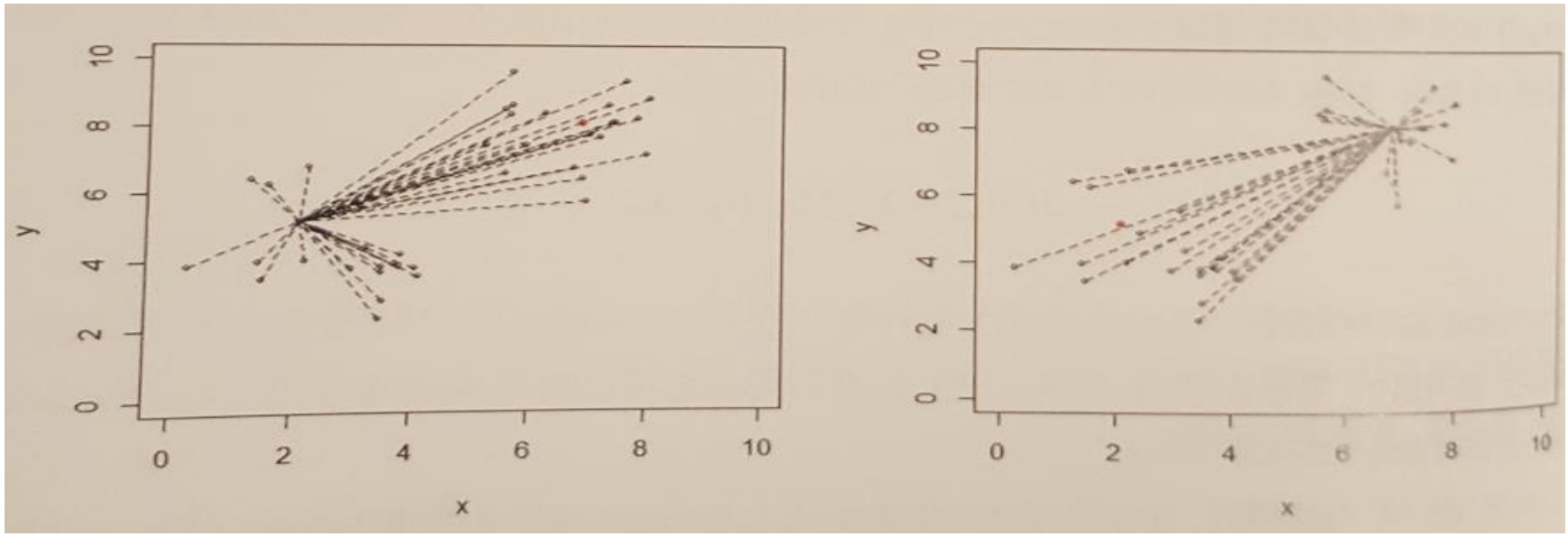


2. K-means Clustering

- **Step1. K가 2인 2-means, 임의의 두 점(군집의 중심)과 모든 점들 사이의 거리계산**

```
> #거리함수 정의(유클리드제곱거리)
> dis <- function(x,y) {
+   return ((x[1]-x[2])^2 + (y[1]-y[2])^2)
+ }
>
> #2-means시뮬레이션
> x <- c(rnorm(20,3,1) , rnorm(20,7,1))
> y <- c(rnorm(20,4,1) , rnorm(20,8,1))
> plot(x , y , cex=.5 , xlim=c(0,10) , ylim=c(0,10))
```





distance1

distance2

2. K-means Clustering

- **Step2. 가장 가까운 거리를 가지는 점에 대해 군집화**

```
> #각각의 점들마다 2개의 군집중 가까운 군집을 해당군집으로 정한다.
> clusters <- c()
> f <- factor(distance1 > distance2)
> levels(f) <- c("1" , "2")
> f
 [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
Levels: 1 2
```

- **Step2. 군집마다 평균(중심점) 계산**

```
> #새로운 군집이 정해졌으면 각각의 군집마다 평균(중심점)을 구한다.
> x1_var <- mean(x[f == "1"])
> x2_var <- mean(x[f == "2"])
> y1_var <- mean(y[f == "1"])
> y2_var <- mean(y[f == "2"])
> x1_var;x2_var;y1_var;y2_var
[1] 6.876146
[1] 3.047911
[1] 8.151848
[1] 4.082133
```

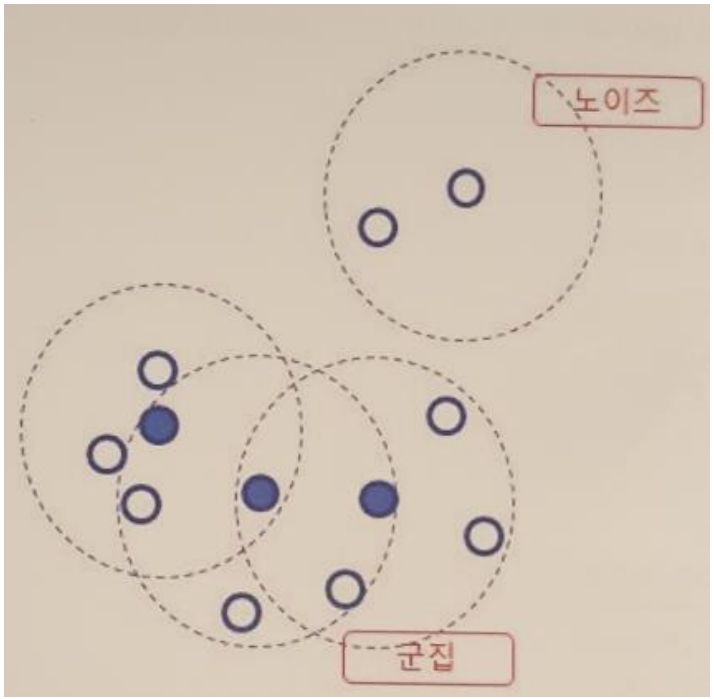
- **Step3. 임의의 두 점과 평균(중심점)의 변화확인**

```
> #중심점의 변화
> c$x - c(x1_var , x2_var)
[1] 0.2540362 0.2050163
> c$y - c(y1_var , y2_var)
[1] -0.47664804 -0.02135478
```

→ 변화가 큼
→ 변화가 거의 없을 때까지 반복
***R의 kmeans 함수 이용 가능**

3. DBSCAN(Density-Based Spatial Clustering of Applications with Noise)

- **데이터 밀도를 이용한 군집화**
- **노이즈 데이터가 군집에 영향을 주지 않음**



n을 4라고 가정

노이즈는 아웃벡터가 4보다 작다

- **이웃벡터** : 한 데이터로부터 반경 ε의 원안에 포함된 데이터벡터(점객체)
- **핵심벡터** : n개 이상의 이웃벡터를 갖는 데이터벡터
- 군집(cluster) : 한 핵심벡터 p에 대하여 접근 가능한 모든 데이터 벡터들의 집합이며 한 군집내의 모든 데이터벡터들은 서로 연결되어 있다.
- 노이즈(noise) : 어떠한 군집에도 속하지 않는 데이터 집합

Ch.4 KNN

<어서와~ 머신러닝은 처음이지>

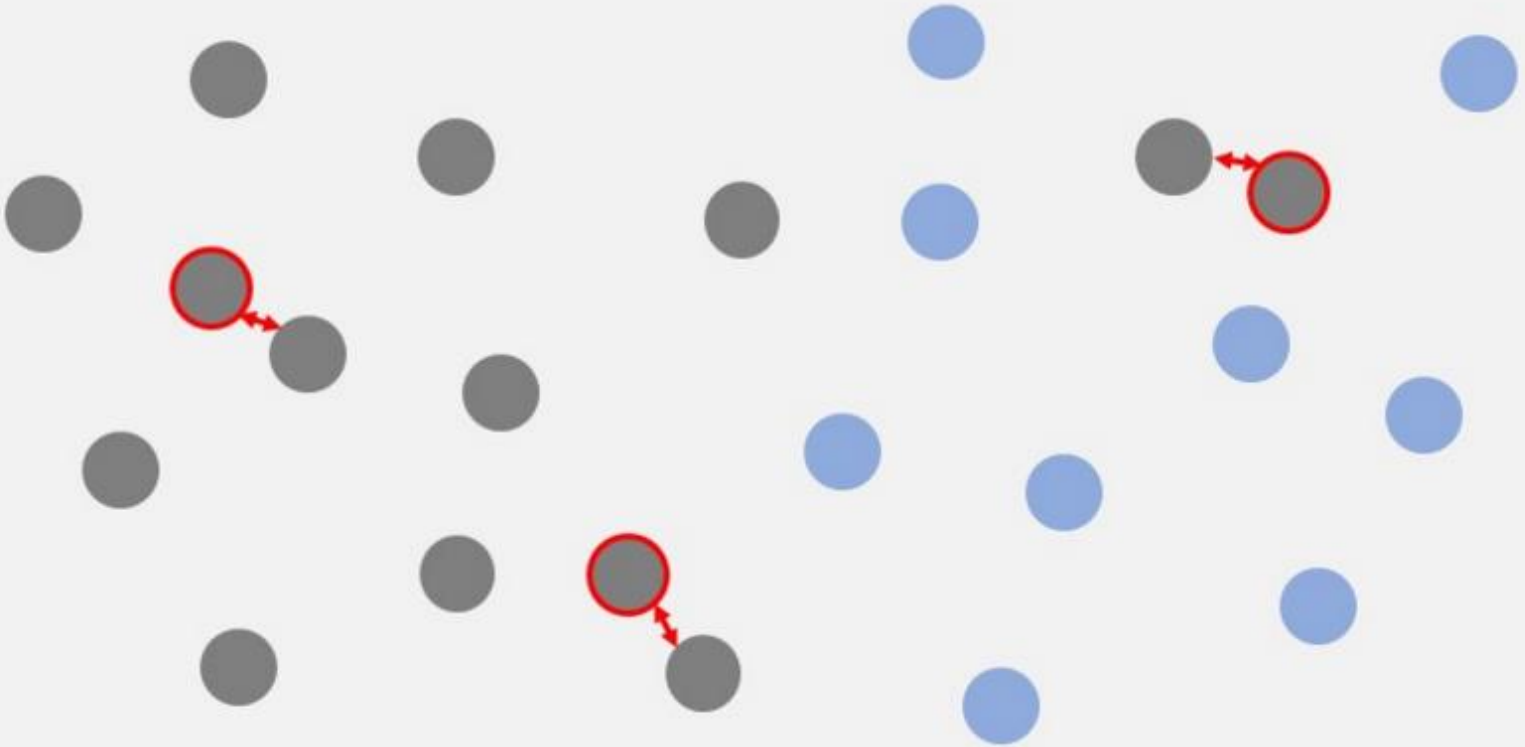
- 거리행렬 기반의 지도 학습
- K개의 인접이웃의 가장많은 Label을 다수결로 선발

$$y = \mathrm{argmax}_v \sum_{p_i} I(v = y_i)$$

장준규

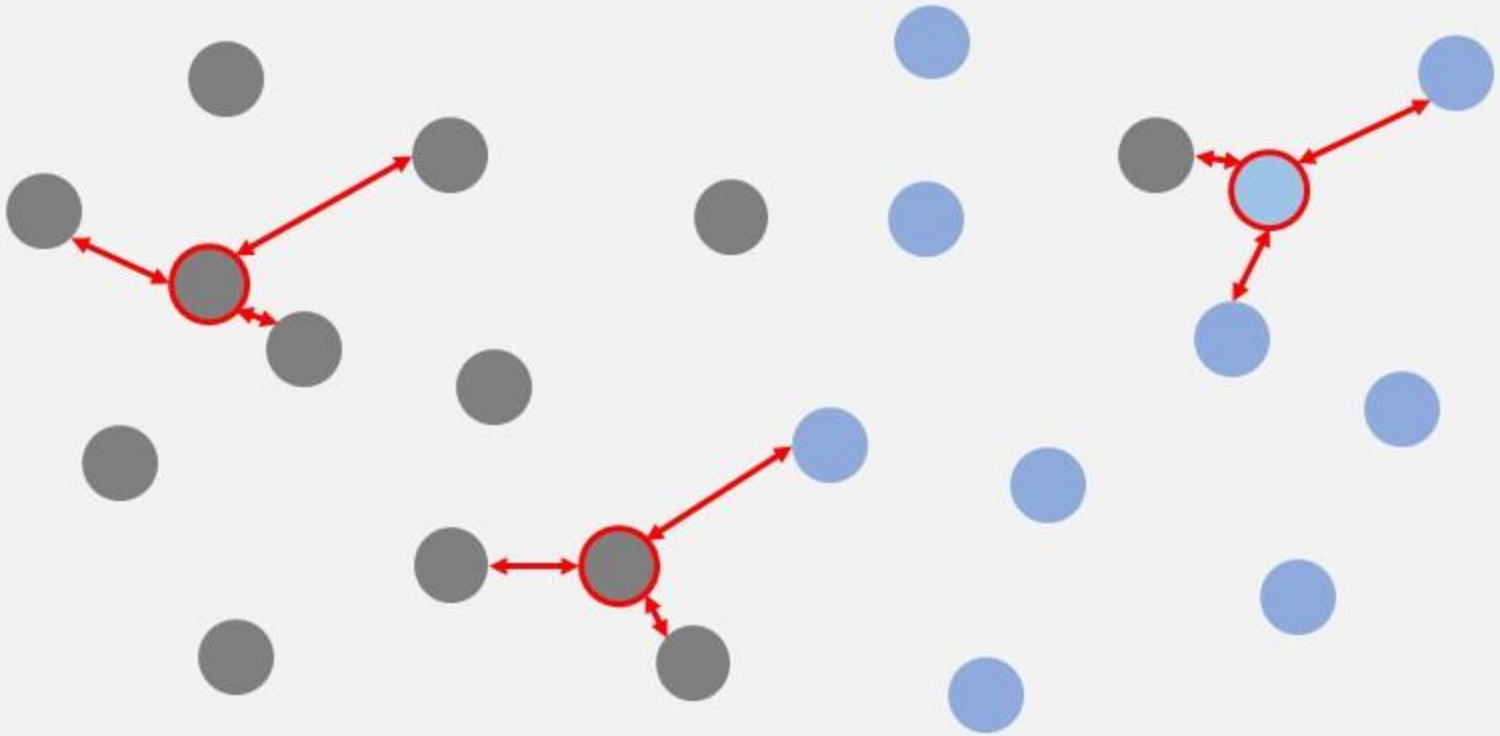
2017. 04. 24

K-Nearest Neighbor



source: http://www.datamarket.kr/xe/board_BoGi29/9880

K-Nearest Neighbor



source: http://www.datamarket.kr/xe/board_BoGi29/9880

(과거) 아이린의 남자호감 타입

1타입: 호감 2타입: 보통 3타입: 극혐

말이 많은 정도	책을 좋아하는 정도	여행을 좋아하는 정도	학교성적	키	피부가 좋은 정도	근육질의 정도	호감LABEL
30	80	40	40	90	90	50	1타입
60	50	70	50	60	60	90	2타입
50	40	80	60	50	70	80	2타입
90	50	60	30	40	20	30	3타입
50	70	30	70	70	80	60	1타입
70	30	60	40	40	40	30	3타입
40	90	60	60	60	80	60	1타입
30	60	90	60	40	70	70	2타입
80	40	20	80	60	40	50	3타입
50	70	70	50	70	90	50	1타입
80	50	60	40	50	40	70	3타입
30	70	30	90	70	80	60	1타입
40	60	90	60	50	70	80	2타입
80	30	50	50	80	40	60	3타입

새로운 남자는 어떤 타입일까?

```
> test <- data.frame(talk=70 , book=50 , travel=30 , school=70 , tall=70 , skin=40 , muscle=50)
> test
  talk book travel school tall skin muscle
1   70   50     30     70   70   40     50
```

K가 3, 4 일 때

```
> library(class)
Warning message:
패키지 'class'는 R 버전 3.0.3에서 작성되었습니다
> train <- like[,-8]
> group <- like[,8]
> knnpred1 <- knn(train , test , group , k=3 , prob=TRUE)
> knnpred2 <- knn(train , test , group , k=4 , prob=TRUE)
> knnpred1
[1] 3타입
attr(,"prob")
[1] 0.6666667
Levels: 1타입 2타입 3타입
> knnpred2
[1] 3타입
attr(,"prob")
[1] 0.75
Levels: 1타입 2타입 3타입
```

3타입: 극혐

주의1. 거리 가중치

앞의 knn은 거리를 계산하지 않고 반경에만 들어가면 다수결로 label이 결정
→ 원점에서 가까운 거리일수록 가중치를 주자

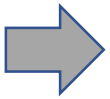
거리가중치 분류방법

$$y = \arg\max_v \sum_{D_z} w_i I(v = y_i),\ w_i = \frac{1}{d(x, x_i)^2}$$

주의2. 표준화

표준화를 하지 않은 채로 거리를 계산시 월수입이 미치는 영향이 나이보다 크게됨
→ R의 scale함수를 통해서 표준화를 시키자

	A	B
1	나이	월수입
2	26	160
3	35	210
4	26	220
5	29	260
6	22	110
7	32	210
8	37	310
9	21	110



```
> buy <- read.csv("buy.csv" , stringsAsFactors = F
> buy$age <- scale(buy$나이)
> buy$pay <- scale(buy$월수입)
> buy
  나이 월수입 상품구매여부
1   26    160         구매 -0.6602858 -0.79243753
2   35    210       비구매  0.9904287 -0.25336438
3   26    220       비구매 -0.6602858 -0.14554975
4   29    260         구매 -0.1100476  0.28570877
5   22    110
```