고용노동부 한국산업인력공단 국가인적자원개발컨소시엄

빅데이터

R을 이용한 데이터 마이닝

(교육기간 : 2018년 08월 25일 ~ 09월 01일)

지태창

대한민국 소프트웨어 기술진흥 · 인력양성 대표기관
한국소프트웨어기술진흥협회

분류 (Classification)

학습지식 개요/요점

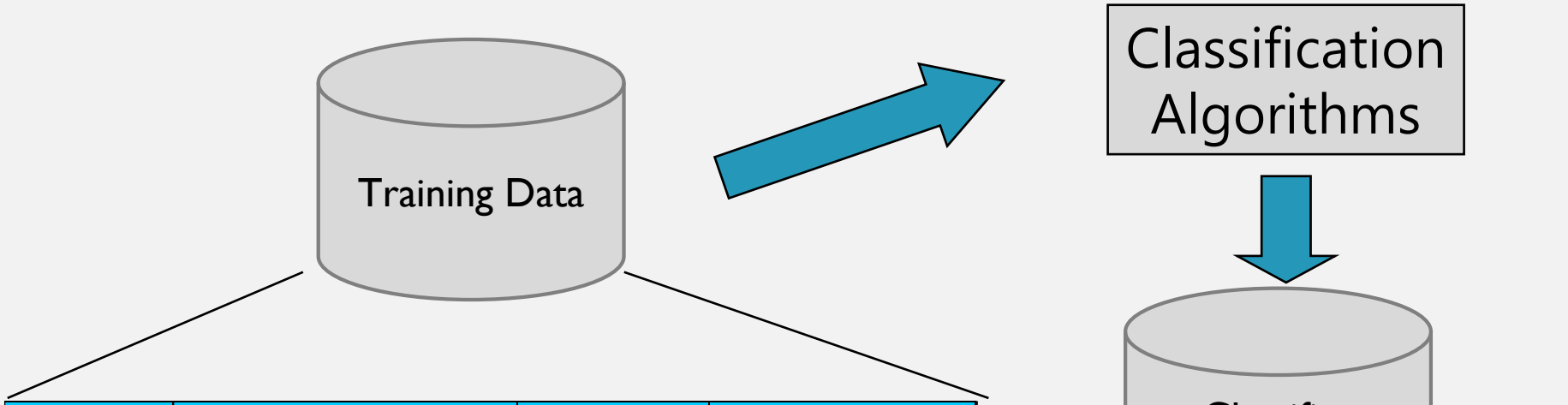
- 데이터를 이미 주어진 클래스에 할당하기 위한 분류의 개념을 확인한다.
- 통계적 기법의 분류 방법인 의사 결정 나무 (Decision Trees)의 개념을 살펴보고, 실행해 본다.
- 인공지능의 분류 방법인 인공 신경 회로망 (Neural Networks)의 개념을 살펴보고, 실행해 본다.
- 통계적 분류 방법인 Support Vector Machine을 살펴보고, 실행해 본다.

Contents

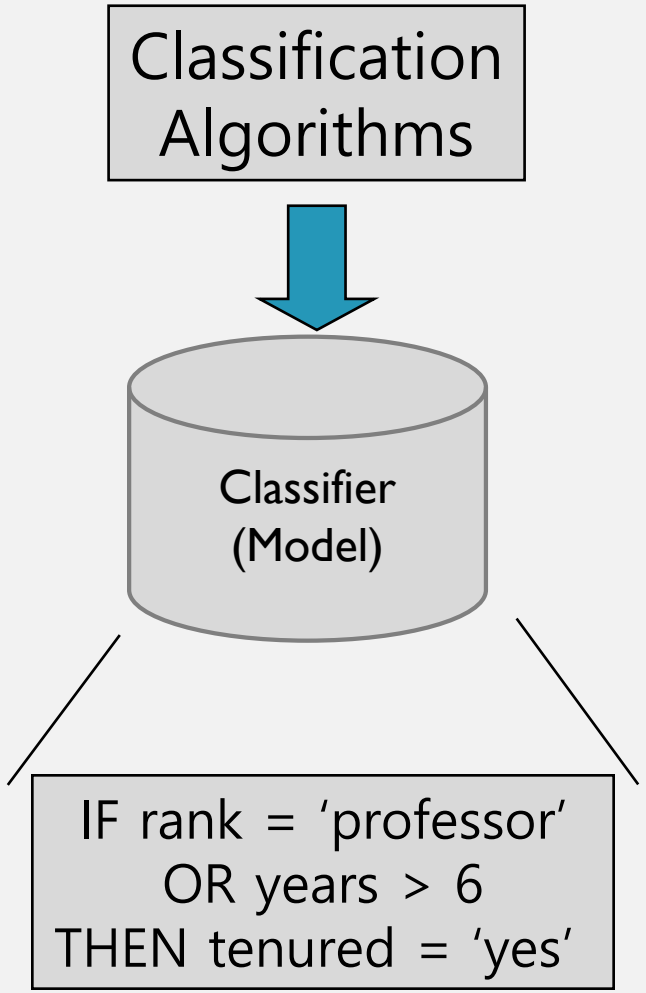
Classification vs. Regression

- Classification
  - 범주형 데이터의 label 예측 (이산형, 명사형)
  - 학습 집합과 이에 맞는 범주 label을 이용하여 학습하고, 새로운 데이터에 적용한다.
- Regression
  - 연속적인 값에 대한 함수/모델
- 주요 사례
  - 신용 평가
  - Target Marketing
  - 의료 진단
  - Fraud detection

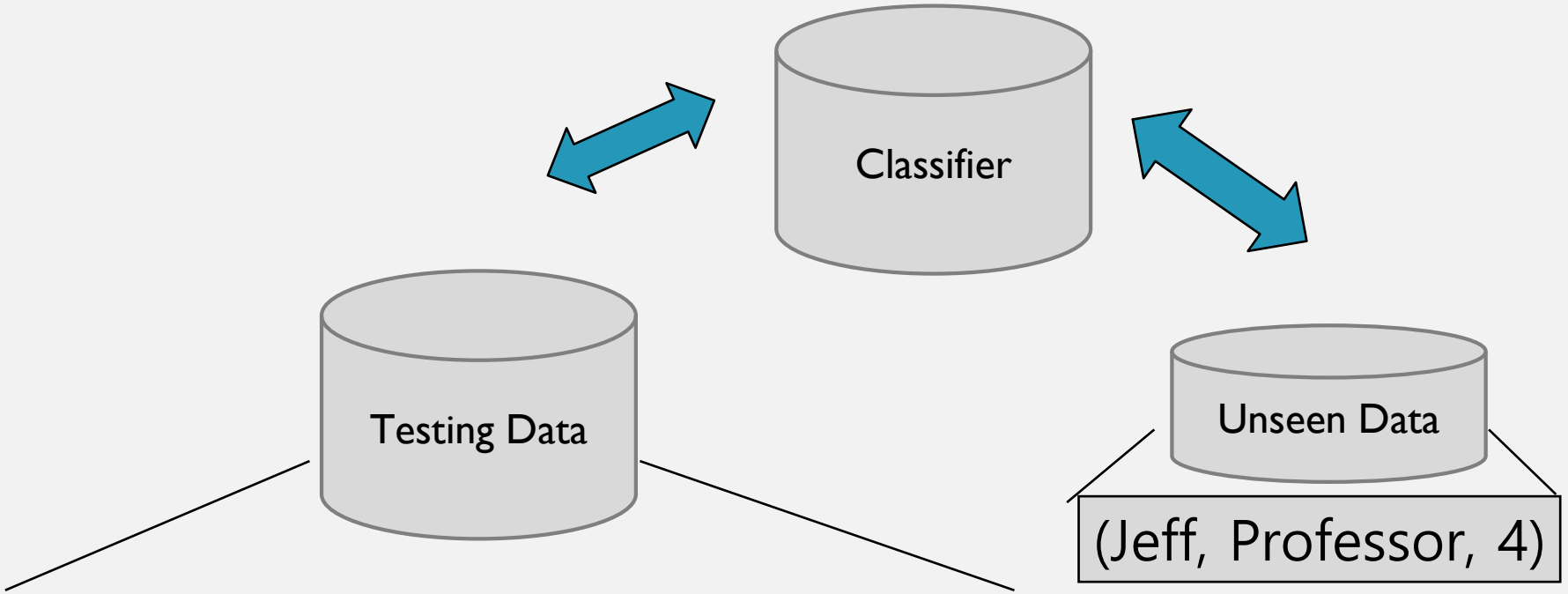
Model construction



NAME	RANK	YEARS	TENURED
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no



Using the Model



NAME	RANK	YEARS	TENURED
Tom	Assistant Prof	2	no
Merlisa	Associate Prof	7	no
George	Professor	5	yes
Joseph	Assistant Prof	7	yes

Data 준비

- Data cleaning
  - Noise(이상치)를 줄이거나 missing value(결측값)을 처리하기 위한 데이터 전처리
- 관련성 분석 (feature selection)
  - 관련이 없거나 불필요한 요소 제거
- 데이터 변환
  - 데이터의 일반화 또는 정규화

Classification 방법 평가

- 정확도
  - 분류 정확도 : class의 label 예측
  - 예측 정확도 : 예측 속성의 값 추정
- 속도
  - 모델 생성 속도 (training time)
  - 모델 사용 속도 (classification / prediction time)
- Robustness
  - Noise와 결측치 처리
- Scalability
  - 처리할 수 있는 데이터 량
- Interpretability
  - 모델에 대한 이해도와 통찰력
- 기타
  - Goodness of rules (의사 결정 나무의 크기 등)

Contents

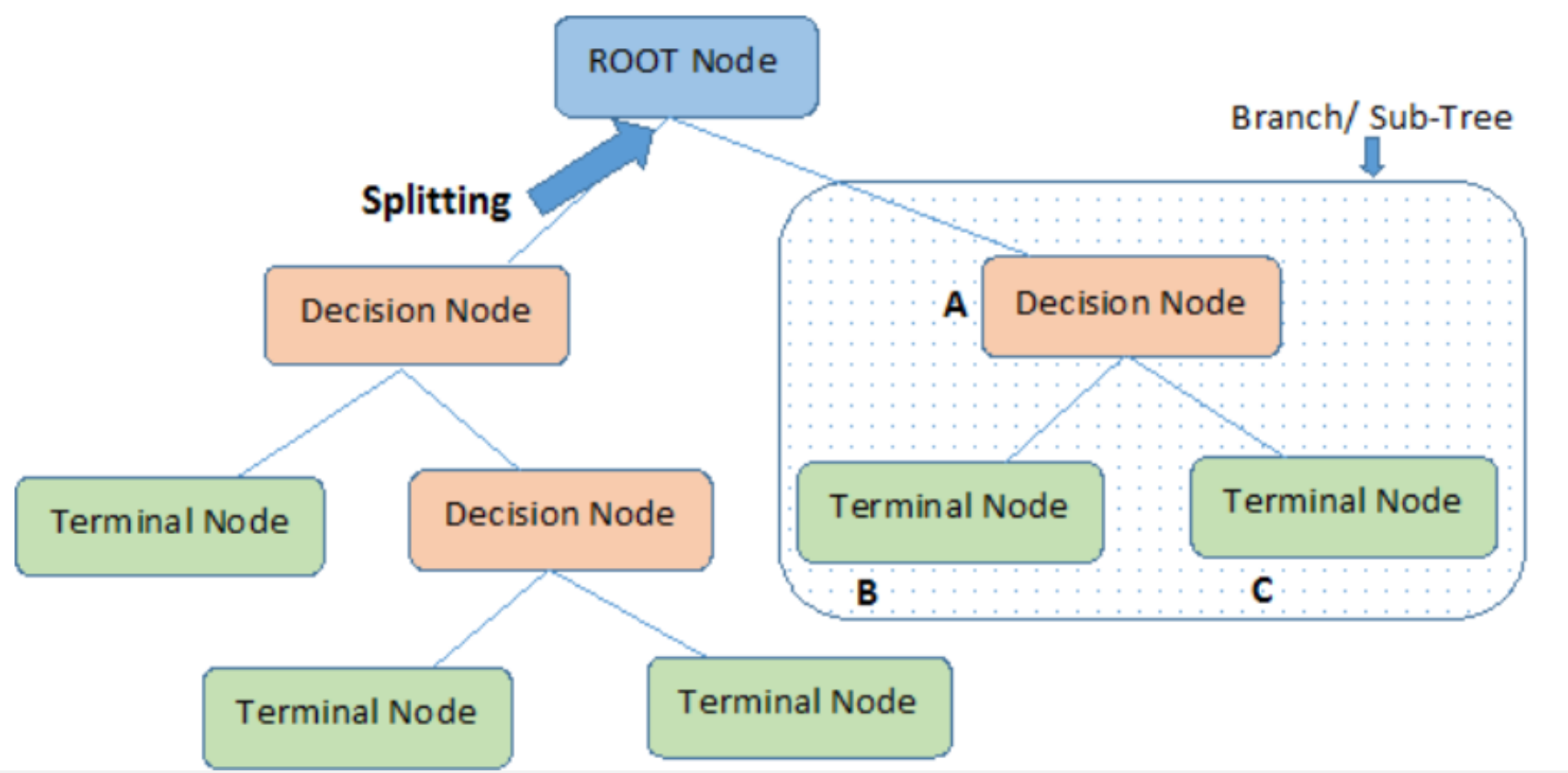
Decision Tree (의사 결정 나무)

- 과거 데이터를 학습하여 데이터 내 존재하는 규칙(Rule)을 자동으로 발견하고, 이를 모델화하여 모델 적용 대상 데이터를 분류/회귀하는 지도학습 알고리즘
- 예측하는 결과가 범주라면 분류 나무(Classification Tree), 연속되는 숫자라면 회귀 나무(Regression Tree)라 함
- 장점
  - 모델 생성 과정을 상대적으로 이해가 쉬움
  - 수학적 배경이 없어도 해석할 수 있는 모델을 생성함
  - 결과가 직관적이기 때문에 투명한 결과 공유가 가능함
- 단점
  - 과적합이 일어나기 쉬움
  - 나무의 규모가 커지면 직관적인 이해가 불가능함

Decision Tree (의사 결정 나무)

- 복합 RULE 기반의 모델이 생성 됨
  - IF (소득 > 2천만원) AND (가족 수 < 3명) THEN '가입'
  - IF (주량 > 2병) AND (흡연량 > 2갑) THEN '위험'

Decision Tree (의사 결정 나무)



Decision Tree (의사 결정 나무)

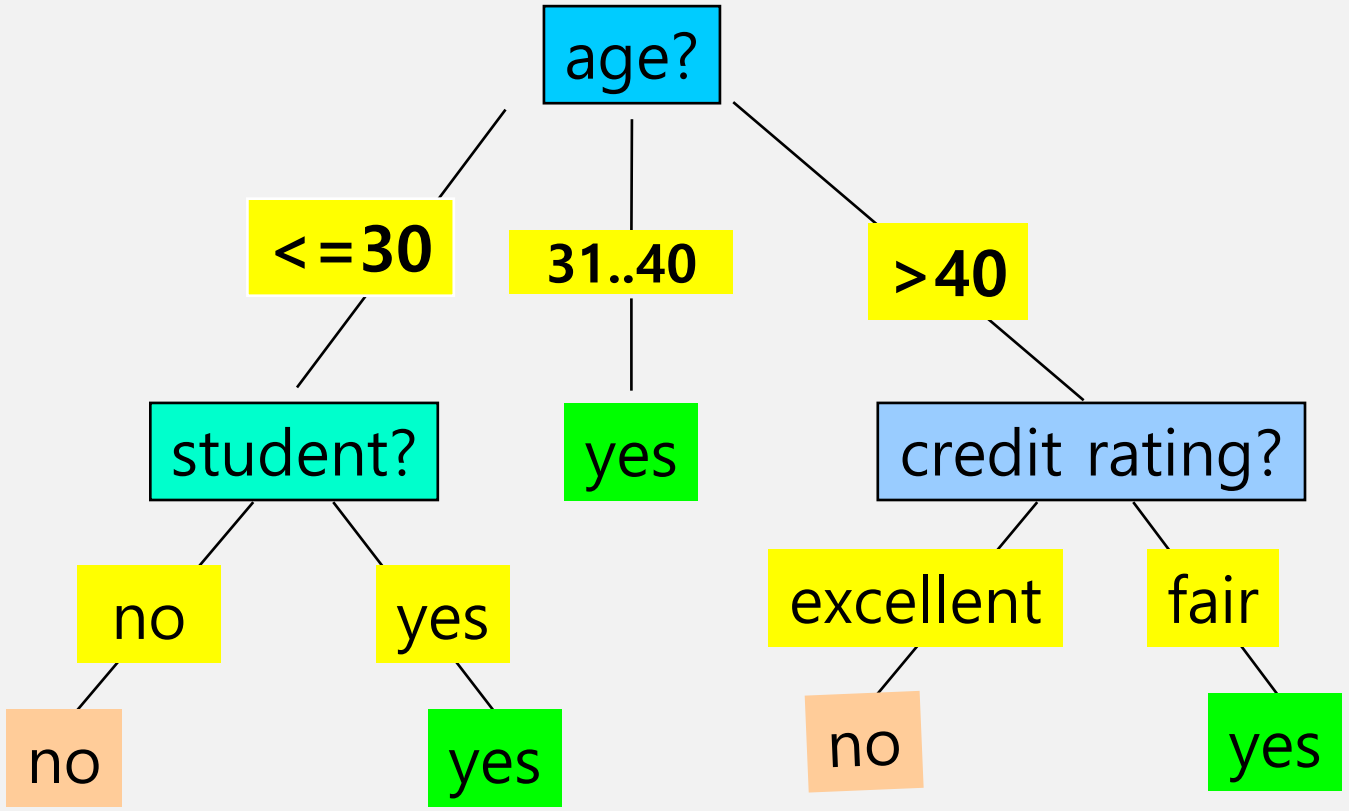
- 루트 노드(Root Node)
  - 아직 의사결정나무 모델이 적용되기 전으로, 데이터 셋 모두 속해 있는 노드를 의미함
- 결정 노드(Decision Node)
  - 특정 노드가 다른 2개의 다른 하위 노드로 분리될 경우, 이를 결정 노드라고 함
- 분리(Splitting)
  - 특정 노드를 2개의 하위 노드로 분리하는 것을 의미함
- 종말 노드(Terminal Node 또는 Leaf)
  - 더 이상 분리 되지 않는 노드를 의미함
- 부모 노드(Parent Node)
  - 노드가 분리 될 경우, 분리되는 노드를 의미함
- 자식 노드(Child Node)
  - 노드가 분리 될 경우, 분리 후의 하위 노드를 의미함
- 가지(Branch)
  - 의사결정나무의 일부분을 가지(Branch) 또는 하위 나무(Sub-Tree)라고 함

Decision Tree (의사 결정 나무)

- 학습 집합

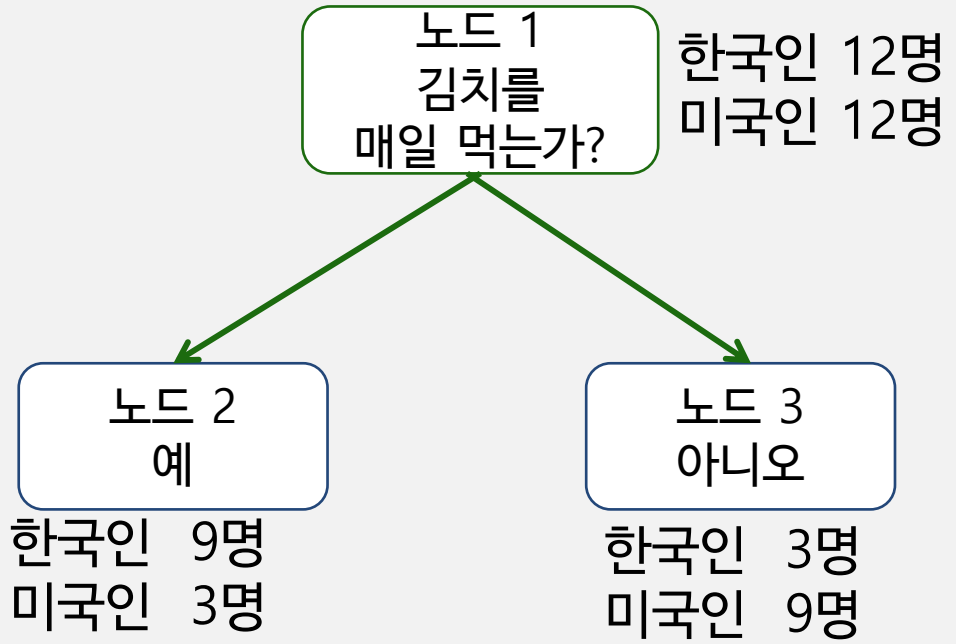
age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

‘buys_computer’에 대한 의사 결정 나무



분리(Splitting)

- 분류 나무에서 가장 핵심이 되는 분류 나무를 만드는 알고리즘
- 특정 노드가 분리 후 더욱 정확한 분류가 가능할 경우 분리가 일어나게 되며, 이를 불순도(Impurity)가 낮아졌다고 함



분리(Splitting)

- 엔트로피(Entropy)
  - 각 노드의 불순도(Impurity)를 측정
  - $Entropy(P) = \sum_{i=1}^{c} -p_i \log_2 p_i$, $Entropy(S) = \sum_{i=1}^{n} w_i Entropy(P_i)$
- 정보 획득(Information Gain)
  - 분리 전 엔트로피 – 분리 후 엔트로피
  - $InfoGain(F) = Entropy(S_1) - Entropy(S_2)$
  - 의사결정나무는 불순도(Impurity)가 감소하는 방향으로 데이터를 분리해가는 모델을 생성 함

분리(Splitting)

- 엔트로피(Entropy)
  - 노드 1 (분리 전) 엔트로피
    - $-\frac{12}{24}log_2\left(\frac{12}{24}\right)-\frac{12}{24}log_2\left(\frac{12}{24}\right)=1$
  - 노드2, 노드3 (분리 후) 엔트로피
    - $-\frac{12}{24}\left(-\frac{9}{12}log_2\left(\frac{9}{12}\right)-\frac{3}{12}log_2\left(\frac{3}{12}\right)\right)+$

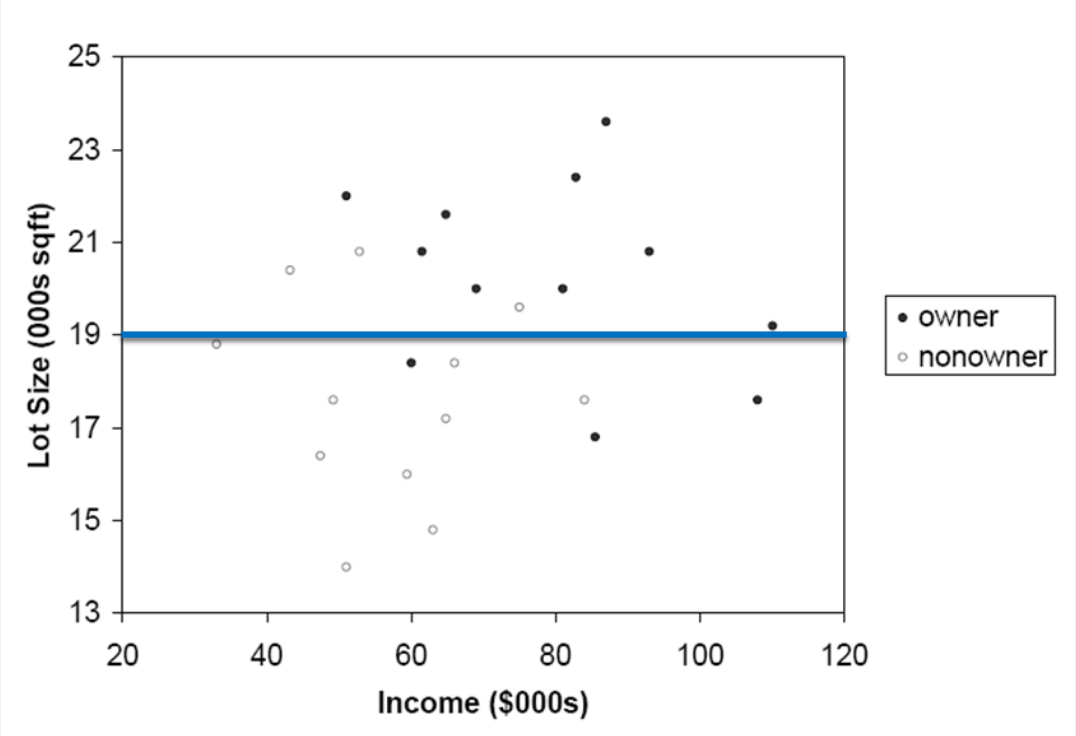
      $-\frac{12}{24}\left(-\frac{3}{12}log_2\left(\frac{3}{12}\right)-\frac{9}{12}log_2\left(\frac{9}{12}\right)\right)$ = 0.81
  - 노드 분리 후 정보획득
    - 1 – 0.81 = 0.19

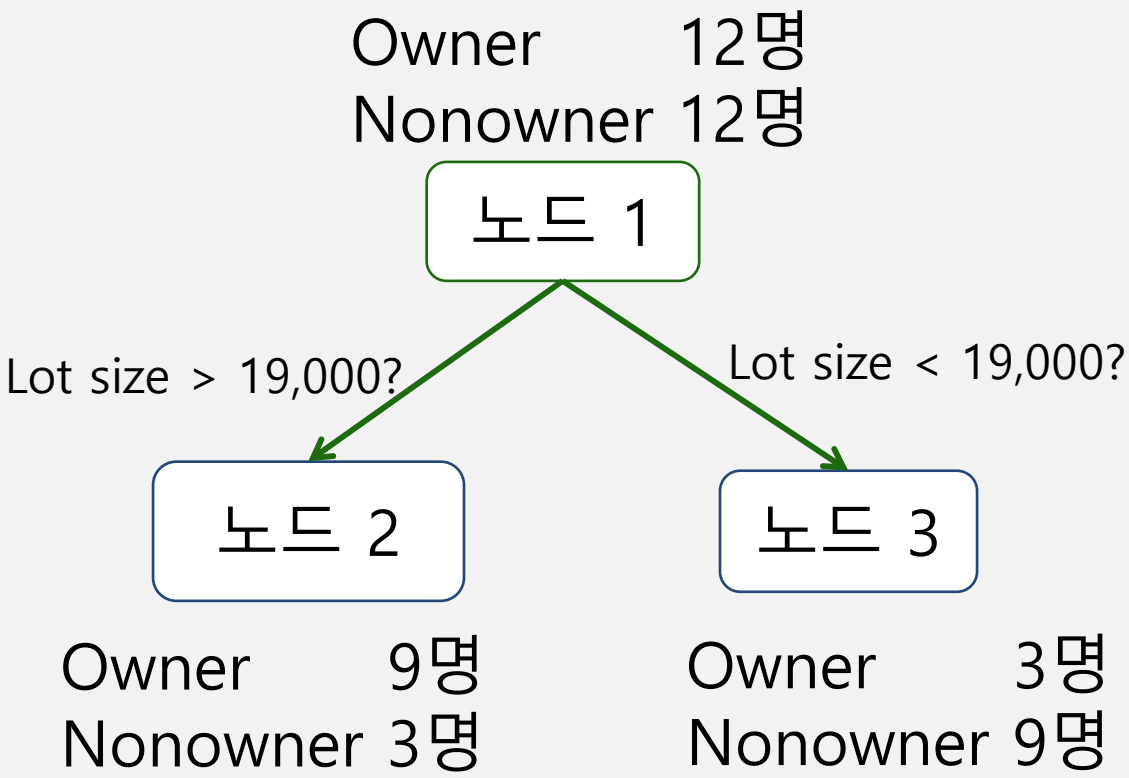
재귀적 분기(Recursive Partitioning)

- 각 노드 별 주어진 자료를 불순도(Impurity)에 기반한 최적의 분리 규칙을 찾아서 분리해가며 의사결정나무를 성장시키는 과정
- 각 노드 별 분리 방법
  - 불순도를 가장 낮출 수 있는 독립변수를 선택
  - 불순도를 가장 낮출 수 있는 독립변수의 값을 선택
  - 선택된 독립변수 및 독립변수의 값으로 분리 전/후 불순도 계산
  - 다른 독립변수, 독립변수의 값으로 위의 단계 반복 수행
  - 가장 불순도가 낮아지는 독립변수와 독립변수 값을 선택하여 분리 수행

재귀적 분기(Recursive Partitioning)

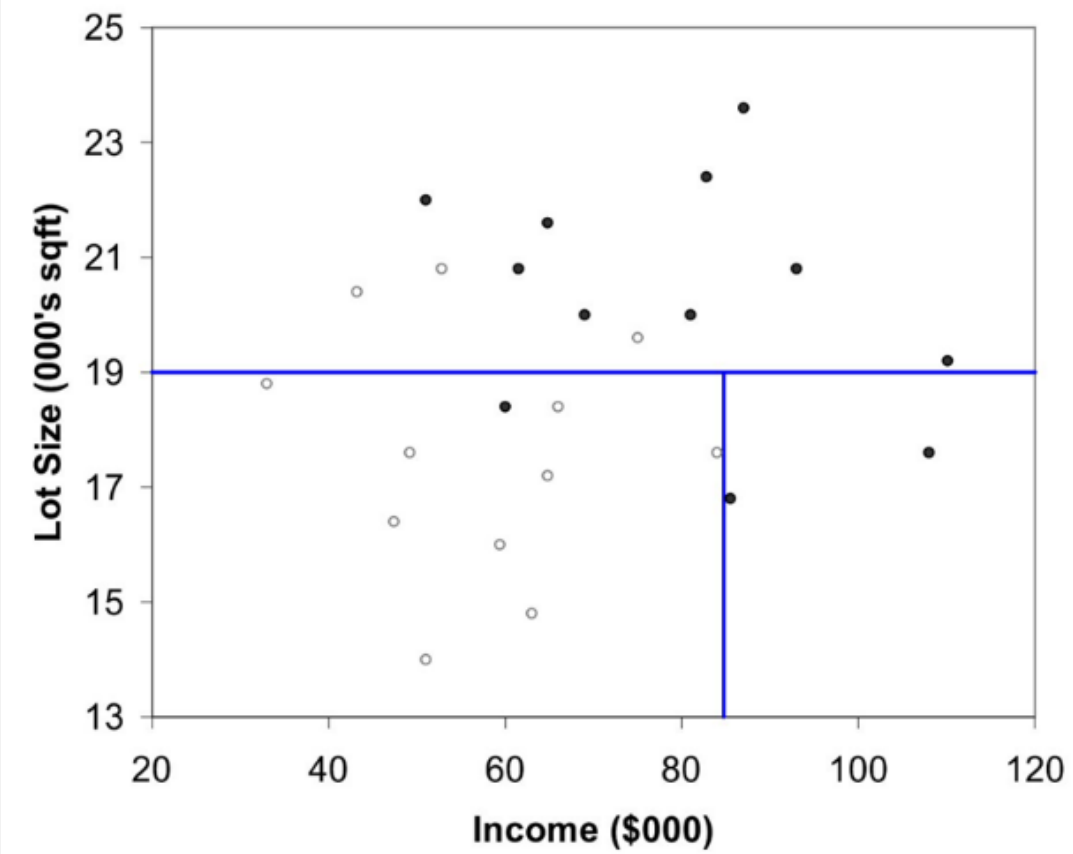
- 첫 번째 분리 수행

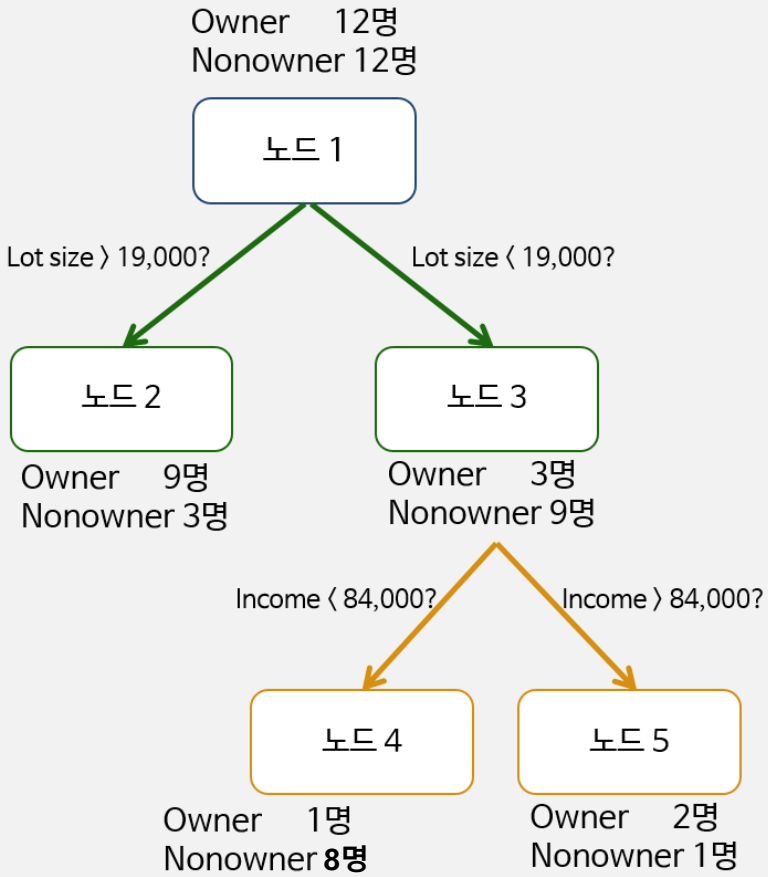




재귀적 분기(Recursive Partitioning)

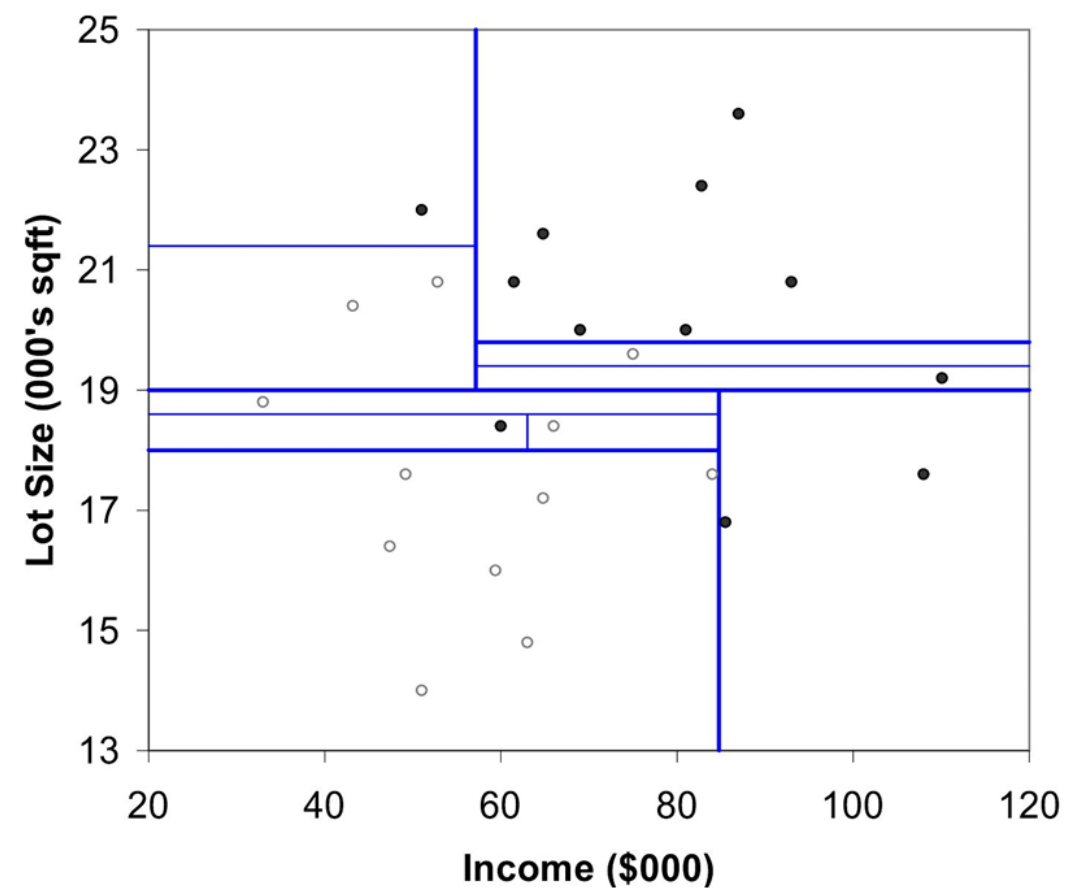
- 두 번째 분리 수행





재귀적 분기(Recursive Partitioning)

- 지속적 분리 수행

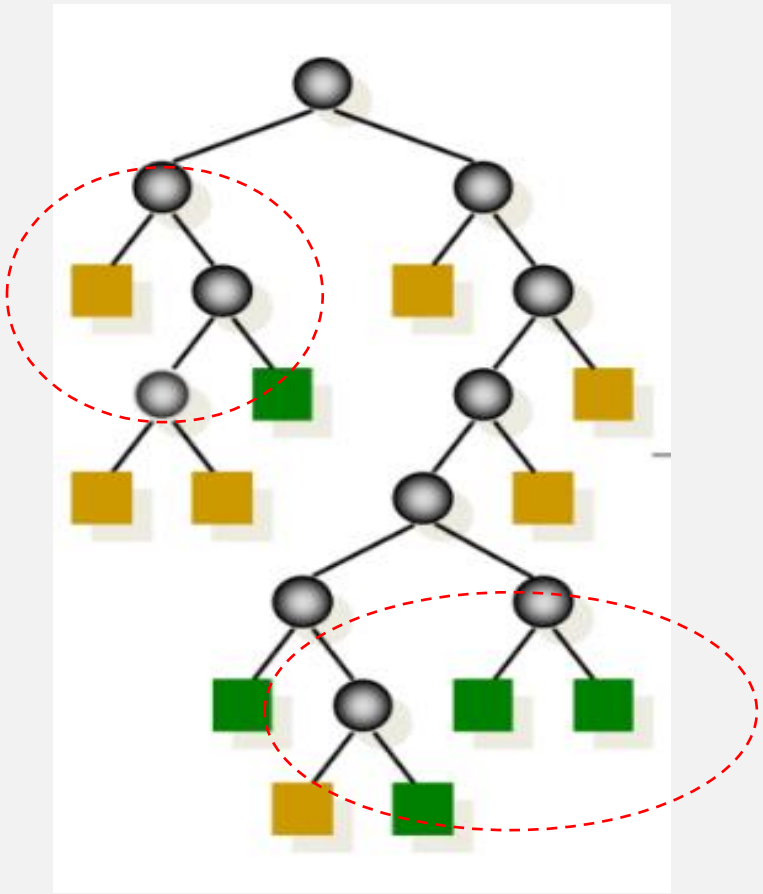


가지치기(Pruning)

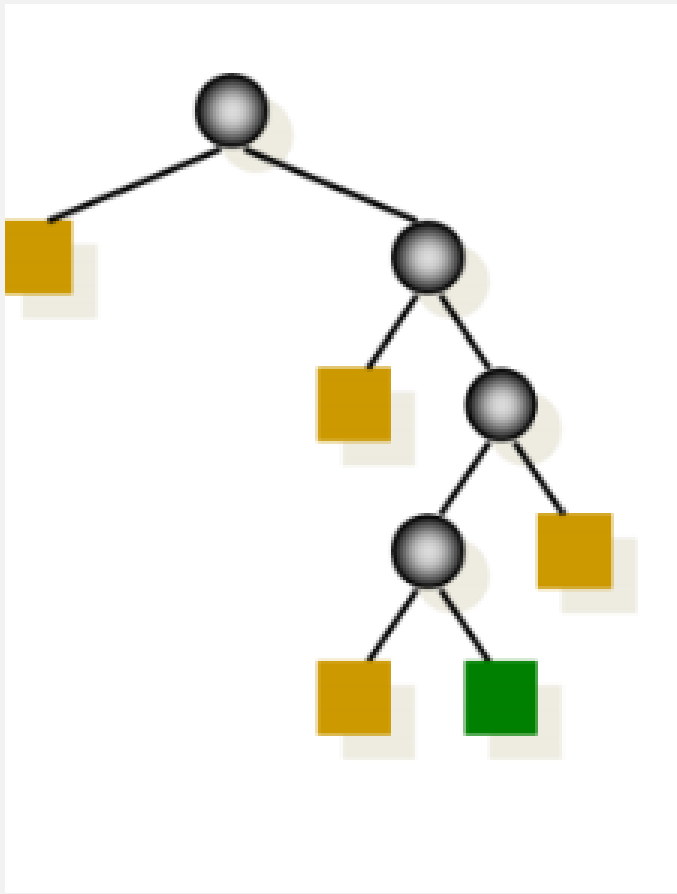
- 의사결정나무가 더 클 수록 좋은 모델일까?
  - 과적합의 위험 존재
- 의사결정나무는 불순도를 줄이는 방향으로 성장하기 때문에 모든 학습 데이터가 완벽하게 분류될 수 있도록 복잡하고 큰 나무로 성장 가능
- 학습 데이터 기반으로 복잡하고 큰 나무로 성장할 경우, 학습 데이터가 아닌 모델이 보지 못한 테스트 데이터에 나무 모델을 적용할 경우 그 성능이 저하될 수 있음 (과적합)

가지치기(Pruning)

가지치기 전

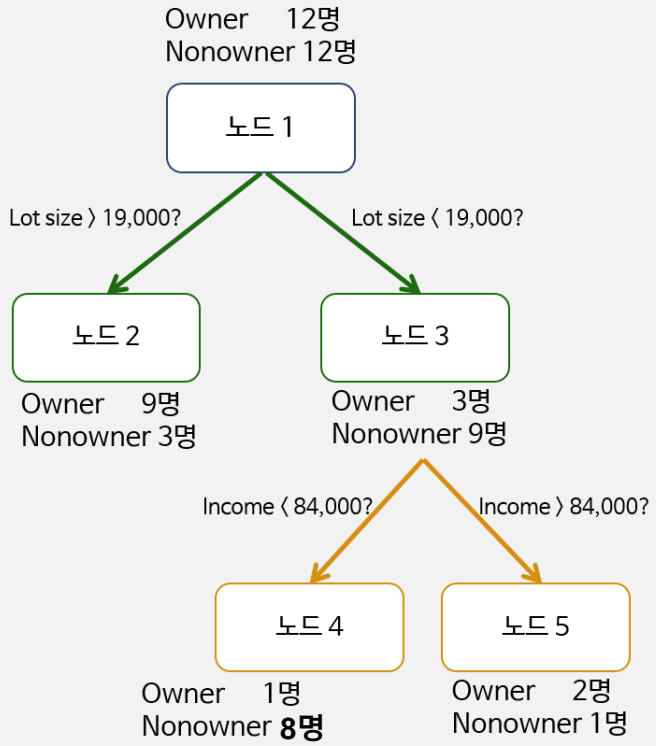


가지치기 후



범주 부여(Labeling)

- 종말 노드(Terminal)의 다수를 이루는 범주를 해당 노드의 최종 범주로 부여하고 규칙을 생성함



(종말노드 범주 부여 예시)
노드 2: Owner
노드 4: Non-owner
노드 5: Owner

(규칙 생성 예시)
노드4:
IF Lot size < 19,000 AND
Income < 84,000
THEN "Non-owner"

나누어 정복하기

- 뿌리 노드에서 시작하여 가장 예측 가능성이 높은 속성을 선택
- 이러한 트리 생성 알고리즘은 멈춤 조건에 이를 때까지 반복
- 멈춤 조건
  - 나누어진 각각의 노드에서 다음을 확인
    - 모든 예제가 같은 범주이다.
    - 예제들을 구별 할 속성이 남아있지 않다.
    - 미리 정의한 크기의 범위에 도달하였다.

특징

- 장점
  - 이해하고 해석하기 쉬움
  - 전문가에 의해서 새로운 통찰력을 얻을 수 있음
  - 새로운 시나리오를 추가 가능
  - 다양한 시나리오에 대해 최고, 최악, 기대값을 알 수 있음
- 단점
  - 분류되는 데이터의 크기가 다를 때, 분류 크기가 많은 것에 더 영향을 받음
  - 값이 많을 때는 계산이 복잡해 짐
  - 정확도가 상대적으로 떨어짐

iris Data

- 데이터 준비

```
data(iris)
data(iris, package = 'datasets')

library(help = 'datasets')
ls('package:datasets')

str(iris)
```

- 학습 데이터와 테스트 데이터로 나눔

```
set.seed(1234)
ind <- sample(2, nrow(iris), replace = TRUE, prob = c(0.7, 0.3))
train.data <- iris[ind ==1,]
test.data <- iris[ind == 2,]
```

Build a ctree

- ctree : Conditional Inference Trees

```
library(party)

myFormula <- Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width
iris_ctree <- ctree(myFormula, data = train.data)

table(predict(iris_ctree), train.data$Species)
```

Result and Predict ctree

- Print ctree

```
print(iris_ctree)

plot(iris_ctree)
plot(iris_ctree, type = 'simple')
```

- Predict ctree

```
testPred <- predict(iris_ctree, newdata = test.data)
table(testPred, test.data$Species)
```

The bodyfat Dataset

```
data('bodyfat', package = 'TH.data')
dim(bodyfat)
head(bodyfat, 5)
```

Train a Decision Tree with rpart

- rpart : Recursive Partitioning and Regression Trees

```
set.seed(1234)
ind <- sample(2, nrow(bodyfat), replace=T, prob = c(0.7, 0.3))
bodyfat.train <- bodyfat[ind == 1,]
bodyfat.test <- bodyfat[ind == 2,]

library(rpart)
myFormula <- DEXfat ~ age + waistcirc + hipcirc + elbowbreadth + kneebreadth
bodyfat_rpart <- rpart(myFormula, data = bodyfat.train, control = rpart.control(minsplit = 10))
```

Result rpart & Pruning

- Result rpart

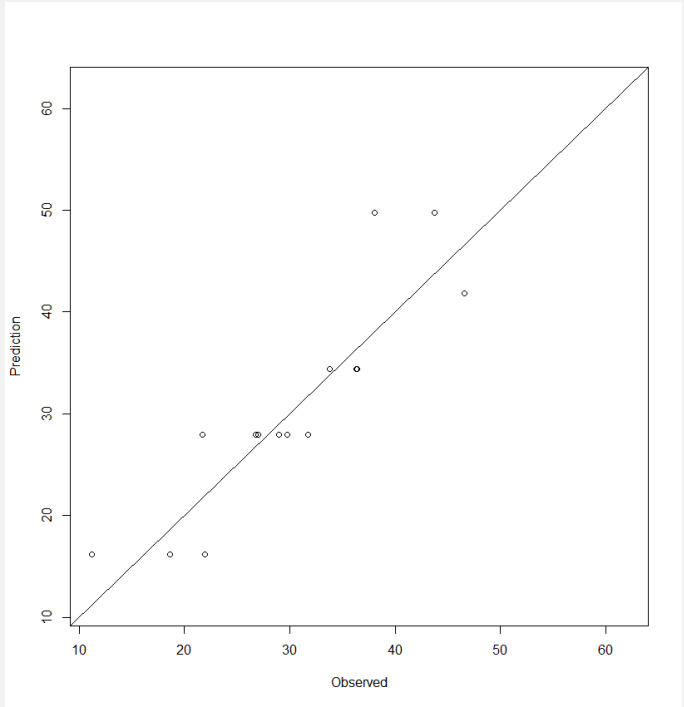
```
print(bodyfat_rpart)

plot(bodyfat_rpart)
text(bodyfat_rpart, use.n = T)
```

- pruning

```
opt <- which.min(bodyfat_rpart$cptable[, 'xerror'])
cp <- bodyfat_rpart$cptable[opt, 'CP']

bodyfat_prune <- prune(bodyfat_rpart, cp = cp)

plot(bodyfat_prune)
text(bodyfat_prune, use.n = T)
```

Model Evaluation

```
DEXfat_pred <- predict(bodyfat_prune, newdata = bodyfat.test)
xlim <- range(bodyfat$DEXfat)
plot(DEXfat_pred ~ DEXfat, data = bodyfat.test,
      xlab = 'Observed', ylab = 'prediction', ylim = xlim, xlim = xlim)
abline(a = 0, b = 1)
```

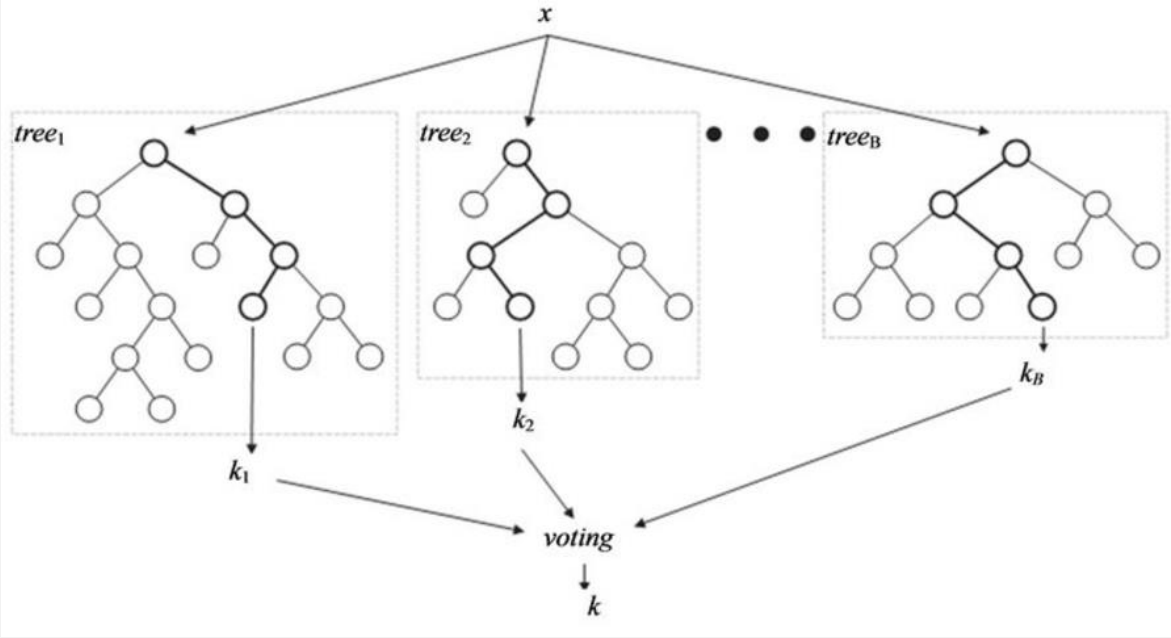


Random Forest (랜덤 포레스트)

- 앙상블(Ensemble) 학습법
  - 머신 러닝에서 더 좋은 예측 성능을 얻기 위해 다수의 학습 알고리즘을 함께 사용하는 방법
- 랜덤포레스트(Random Forest)
  - 앙상블(Ensemble) 지도 학습 방법의 일종으로, 학습 과정에서 구성한 다수의 의사결정나무로부터 분류 또는 회귀 분석을 수행

Random Forest (랜덤 포레스트)

- 랜덤포레스트(Random Forest)
  - 다수의 의사결정나무가 예측하는 결과를 종합하여 다수의 의사결정나무가 예측하는 결과 (투표)를 최종 예측결과로 선정



랜덤포레스트 예시

의사결정나무	예측
A	가입
B	가입
C	미 가입
D	미 가입
E	가입

가입 3 vs 미 가입 2로 최종 '가입'으로 예측

Random Forest (랜덤 포레스트)

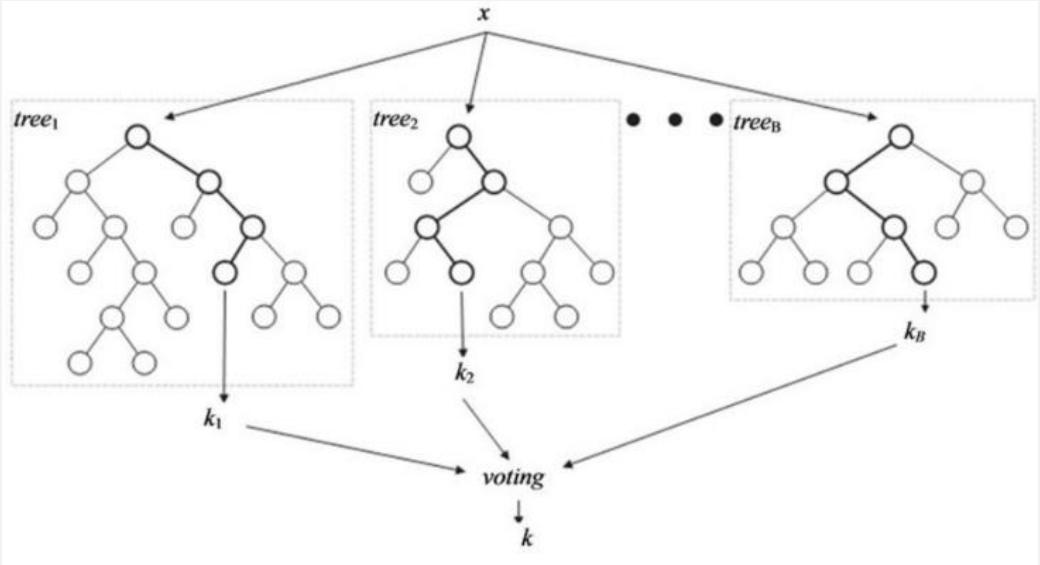
- 장점
  - 임의성(Randomness)을 부여하여 보다 높은 정확도 및 낮은 과적합 경향
  - 변수 중요도 정보 제공
  - 의사결정나무의 주요 단점 (과적합 등)을 보완 가능
- 단점
  - 블랙박스 모델로써 모델 자체의 해석이 어려움
  - 의사결정나무는 해석이 용이한 모델

Random Forest (랜덤 포레스트)

- 임의성
  - 랜덤 포레스트의 핵심이 되는 개념
  - 랜덤 포레스트는 임의성을 활용해 모델의 정확도를 극대화하고 과적합을 줄일 수 있는 특징이 있음
  - 랜덤 포레스트는 임의성에 의해 서로 조금씩 다른 특성을 가지고 있는 여러 의사결정나무로 구성되어 있음
- 데이터의 임의성
  - 각 의사결정나무가 임의로 추출된 샘플로 부터 생성
- 변수의 임의성
  - 각 의사결정나무에서 분리 수행 시 독립변수가 임의로 선택되어 생성됨

Random Forest (랜덤 포레스트)

- 예측
  - 분류(Classification) : 각 나무 별로 예측한 결과를 투표(Voting) 하는 방법으로 최종 예측 범주를 결정하게 됨
  - 회귀(Regression) : 각 나무 별로 예측한 결과의 평균(Average)하는 방법으로 최종 예측 값을 결정하게 됨



Train a Random Forest

```
ind <- sample(2, nrow(iris), replace = TRUE, prob = c(0.7, 0.3))
train.data <- iris[ind == 1, ]
test.data <- iris[ind == 2, ]

library(randomForest)
rf <- randomForest(Species ~ ., data = train.data, ntree = 100, proximity = T)

table(predict(rf), train.data$Species)
```

Result of Random Forest

```
print(rf)

# Error Rate
plot(rf, matin = '')

# Vairable Importance
importance(rf)

varImpPlot(rf)
```

```
Call:
 randomForest(formula = Species ~ ., data = train.data, ntree = 100,
  proximity = T)
               Type of random forest: classification
                     Number of trees: 100
No. of variables tried at each split: 2

        OOB estimate of  error rate: 5.94%
Confusion matrix:
           setosa versicolor virginica class.error
setosa         37          0         0  0.00000000
versicolor      0         26         3  0.10344828
virginica       0          3        32  0.08571429
```

Predictions of Random Forest

```
irisPred <- predict(rf, newdata = test.data)
table(irisPred, test.data$Species)
```

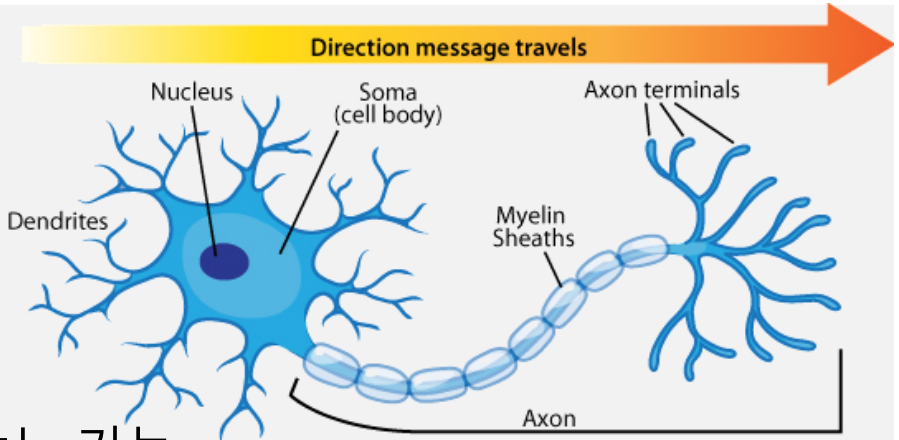
Contents

- 분류의 개념
- Decision Trees
- Neural Networks
- SVM (Support Vector Machine)

Neural Networks

- 신경회로망
  - 인간의 두뇌 작용을 신경세포(neuron, nerve cell)들간의 연결 관계로 모형
  - 연결주의(connectionism) 혹은 신연결주의(neo-connectionism)라 불림
  - 신경세포의 구조 및 기능을 단순화 하여 수학적 모형(model)으로 표시하고 이런 소자들을 상호 연결하여 신경회로망을 구성
  - 음성 인식, 영상 인식, 감성 인식, 추론 학습 기능 등의 문제 해결
  - 학습(learning)과 재생(recall)라는 두 단계의 과정
    - 학습(learning): 패턴 부류에 따라 신경망의 연결가중치 조정
    - 재생(recall): 학습된 가중치와 입력벡터와의 거리 계산하여 가장 가까운 클래스로 분류

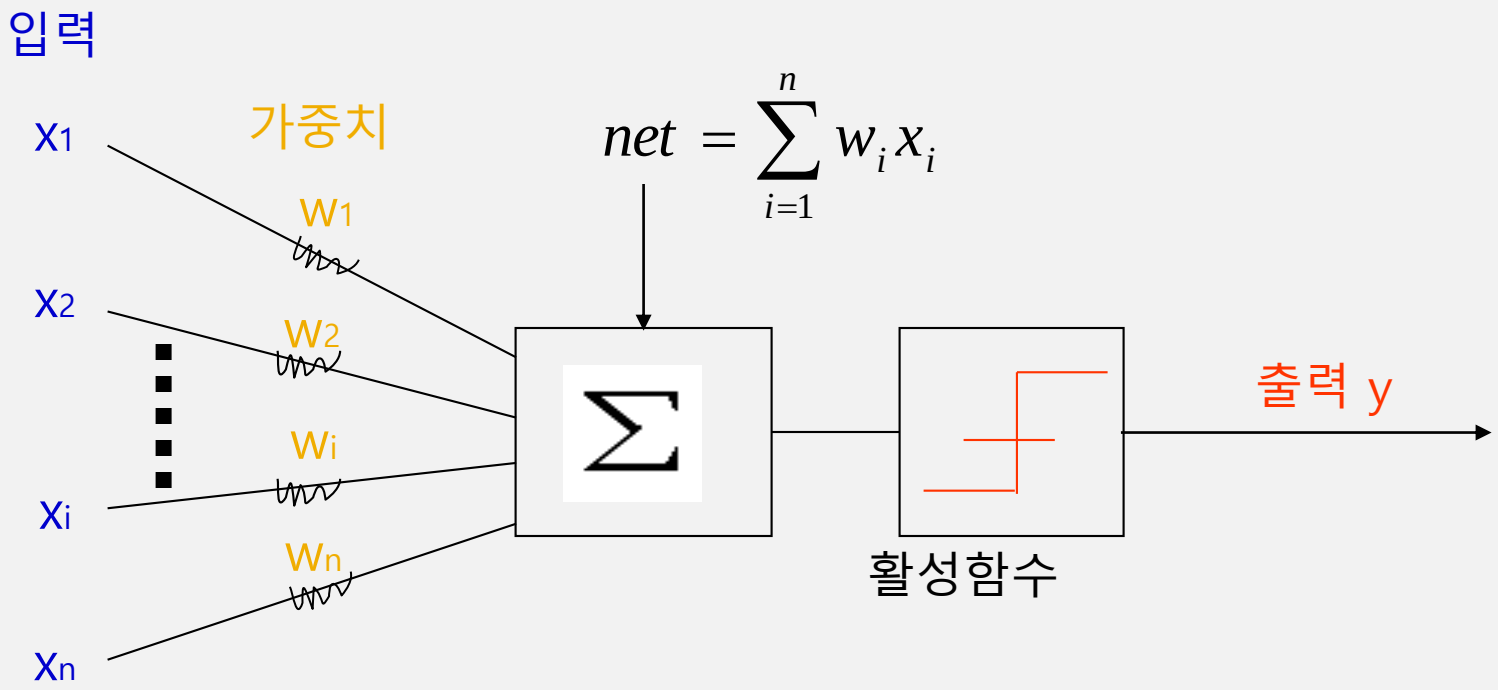
Neural Networks



- 신경세포의 구성 요소
  - 수상돌기(dendrite)
    - 다른 신경세포의 전기화학적 신호를 신경망칩을 통해 받는 기능
  - 신경연접(synapse)
    - 받아들인 자극을 강도에 따라서 활성화(excite) 또는 억제(inhibit)
  - 세포체(cell body, soma)
    - 활성적(active) 입력신호와 억제적(inhibitory) 입력신호들을 합함
  - 축색돌기(axon)
    - 세포체의 점화에 의해 발생되는 전기화학적 에너지를 다른 신경세포까지 연결 전달
  - 신경세포 간의 정보교환은 모두 신경연접를 통하여 행하여 지며 신경 정보의 전달은 편 지향적

Neural Networks

- 인공 신경회로망의 처리소자(Processing Unit: PU) 구조



Neural Networks의 특징

- 예제를 통한 학습 가능성
  - 많은 신경망은 예제들로 이루어진 훈련세트(training set)로 학습 가능
  - 지도학습(supervised learning)의 시나리오를 통해 입력 패턴과 원하는 목표 출력과의 차이를 줄여나감
  - 입력값들이 주어지면 비슷한 것들이 모아지는 비지도학습(unsupervised learning)도 있음
  - 보통(비학습) 컴퓨터 프로그램과 차이점
    - → 프로그램에 의해 미리 정해진 순서를 따라 수행

Neural Networks의 특징

- 결함 허용성
  - 분산 저장 방법의 결과로 정보의 중복된(redundancy) 저장
  - 시스템이 일부 파손되더라도 대부분 작동 가능한 파손내구성(fault tolerant)
- 일반화
  - 신경망은 분류 작업을 수행하는데 있어서 뛰어난 성능을 발휘
  - 이전의 예들로부터 집단의 특징들을 일반화
    - → 예가 많을 수록 좋은 일반화 가능
- 연상기억(Associative Memory: AM)
  - 패턴의 일부나 약간 다른 패턴으로부터 전체를 얻을 수 있는 연상기억
  - 궤환(feed back)회로의 존재로 서로 영향을 주며 해에 수렴하는 동적(dynamic) 작동원리

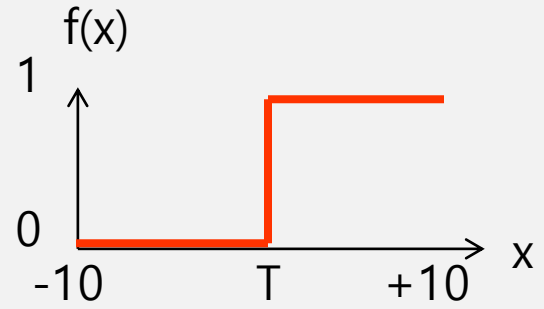
Neural Networks의 요소

- 신경회로망의 구조
  - 처리 소자들간의 연결 및 작용 방향 등을 정의
- 노드 특성
  - 각각의 노드에서의 처리 특성
- 학습 규칙
  - 신경회로망이 원하는 결과를 만들어 내도록, 가중치 조절 과정을 통하여 입출력 관계를 근사하게, 하는 학습 알고리즘
- 신경회로망의 구성
  - 입력층: 외부의 입력을 받아들이는 층
  - 출력층: 신경회로망의 처리 결과를 외부로 내보내는 층
  - 가중치: 각 처리소자간의 연결성을 표현하는 연결가중치 (connection weight)

Neural Networks의 요소

- 신경회로망의 구성
  - 활성(activation) 함수: 가중치합을 입력으로 하여 (보통) 비선형함수를 사용하여 결과값을 출력값으로 변환
    - 스텝함수: 입력이 임계치(threshold)보다 작으면 0을 출력, 임계치보다 크면 1을 출력하는 함수

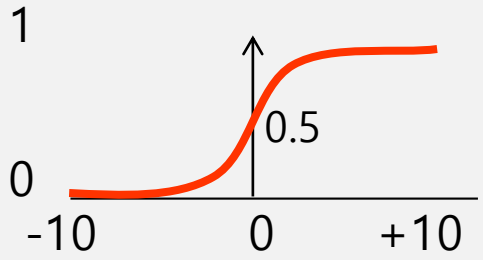
$$F(x) = \begin{cases} 1 & \text{if } x > T \\ 0 & \text{otherwise} \end{cases}$$



Neural Networks의 요소

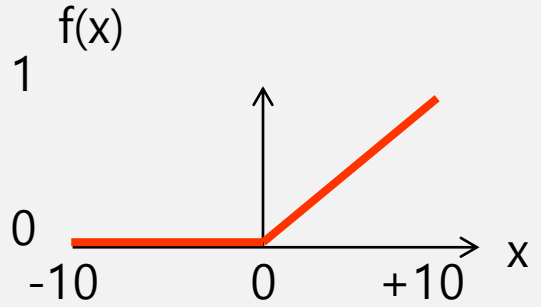
- 신경회로망의 구성
  - 활성(activation) 함수
    - 시그모이드 함수: 넓은 범위의 입력에 대한 적절한 이득제어를 제공

$$F(X) = \frac{1}{1+e^{-x}}$$

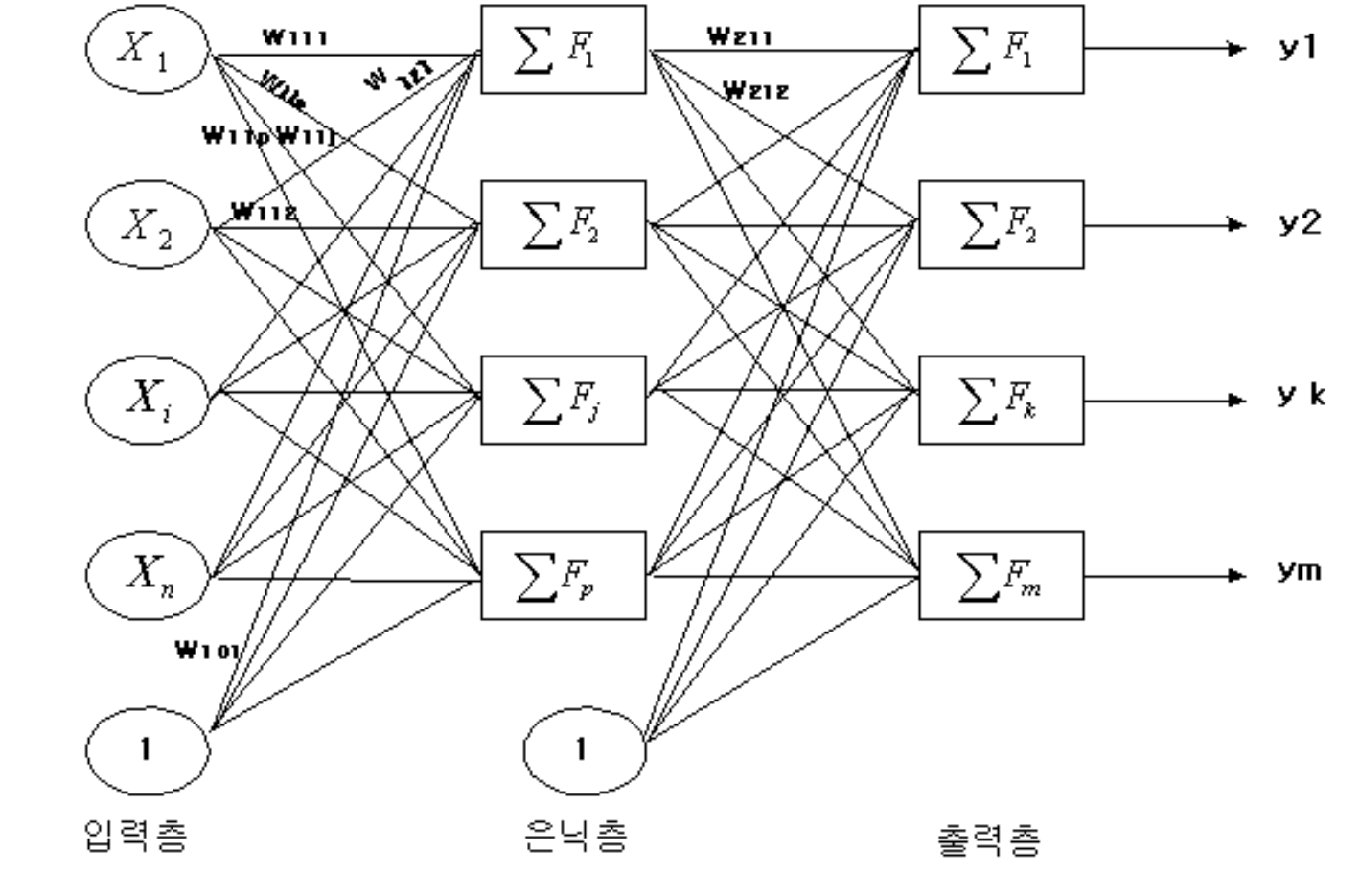


    - ReLU (Rectified Linear Unit) 함수 : 네트워크가 많아도 학습이 이루어지게 하는 함수

$$F(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$$



Multi Layer Perceptron



Neural Networks의 장단점

- 장점
  - 다양한 문제에 적용 가능: 인식, 분류, 예측, 시각화, 제어
  - 우수한 일반화 성능
  - 잡음 처리 능력 우수
  - 범주 및 연속형 변수 처리 가능
  - 다양한 S/W 패키지 사용 가능
  - 복잡한 문제에서도 쓸만한 결과를 도출
- 단점
  - 모든 입력값과 출력값의 디지털화 필요
  - 신경망의 학습 결과를 설명하는 것이 어려움
  - 입력 변수의 수에 비례하여 필요한 자료의 수 증가
  - 지역적 해에 수렴하는 학습 경향

Train by neuralnet

- **infert dataset:** 자연 유산과 인공 유산 후 불임

```
data(infert, package = 'datasets')

ind <- sample(2, nrow(infert), replace = T, prob = c(0.7, 0.3))
train.infert <- infert[ind == 1,]
test.infert <- infert[ind == 2,]

library(neuralnet)

net.infert <- neuralnet(case~parity+induced+spontaneous, train.infert, hidden = 2, err.fct="ce",
linear.output=FALSE, likelihood=TRUE)

print(net.infert)

plot(net.infert)
```

Train by nnet (1/3)

```
library(nnet)

infert.nn <- nnet(case ~ parity+induced+spontaneous, data=train.infert, size = 2)

inf.train.result <- predict(infert.nn, train.infert)
inf.train.result <- ifelse(inf.train.result > 0.5, 1, 0)

table(train.infert$case, inf.train.result)

inf.test.result <- predict(infert.nn, test.infert)
inf.test.result <- ifelse(inf.test.result > 0.5, 1, 0)

table(test.infert$case, inf.test.result)
```

Train by nnet (2/3)

```
iris.targets <- class.ind(iris$Species)
iris.new <- cbind(iris[,1:4], iris.targets)

ind <- sample(2, nrow(iris.new), replace = T, prob = c(0.7, 0.3))
train.iris <- iris.new[ind == 1,]
test.iris <- iris.new[ind == 2,]

iris.nn <- nnet(train.iris[,1:4], train.iris[5:7], size = 2, rang = 0.1, decay = 5e-4, maxit = 200)

test.cl <- function(true, pred) {
  true <- max.col(true)
  cres <- max.col(pred)
  table(true, cres)
}

test.cl(train.iris[,5:7], predict(iris.nn, train.iris[,1:4]))

test.cl(test.iris[,5:7], predict(iris.nn, test.iris[,1:4]))
```

Train by nnet (3/3)

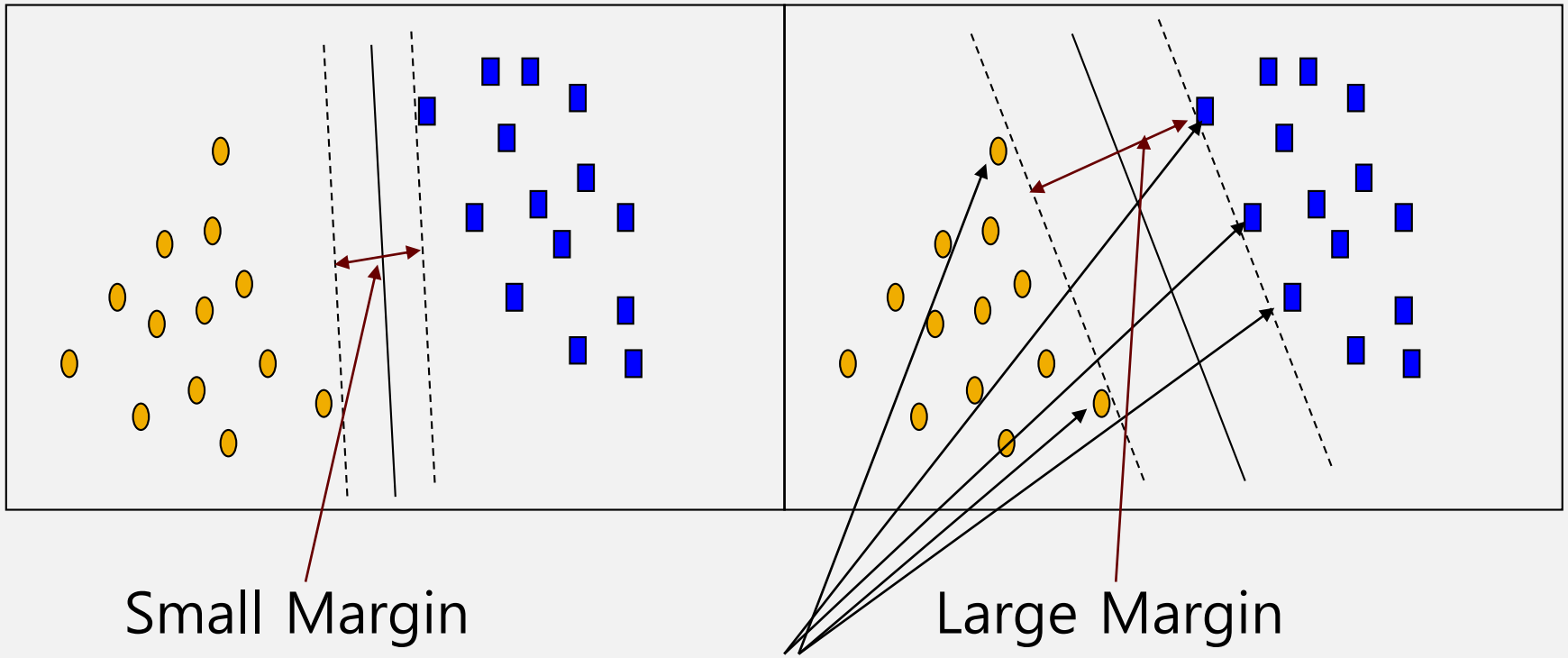
```
train.iris2 <- iris[ind == 1,]
test.iris2 <- iris[ind == 2,]

iris.nn2 <- nnet(Species ~ ., data = train.iris2, size = 2, rang = 0.1, decay = 5e-4, maxit = 200)

table(train.iris2$Species, predict(iris.nn2, train.iris2, type = "class"))
table(test.iris2$Species, predict(iris.nn2, test.iris2, type = "class"))
```

Contents

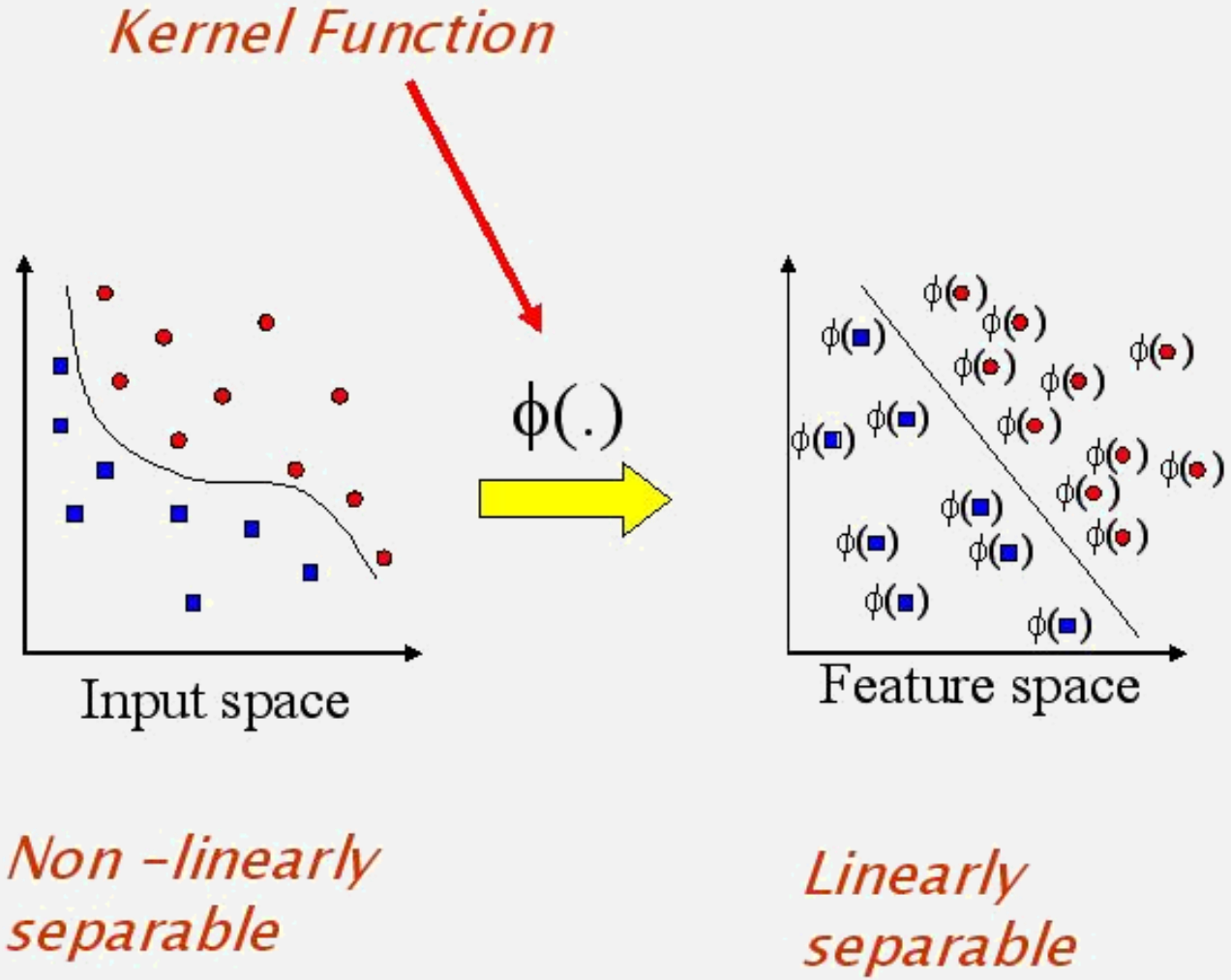
SVM

- 선형 및 비선형 데이터에 대한 새로운 분류 방법
- 비선형 매핑을 사용하여 원본 학습 데이터를 더 높은 차원으로 변환
- 새로운 차원에서 선형 최적 분리 초평면 (즉, "결정 경계")을 찾음
- 충분히 높은 차원으로의 적절한 **비선형 매핑**을 사용하면 두 클래스의 데이터를 항상 초평면으로 분리 가능
- SVM은 Support Vector (지원 벡터 : "필수" 훈련 튜플) 및 margins (여백 : 지원 벡터에 의해 정의 됨)을 사용하여 이 초평면을 찾음

SVM



SVM


Kernel Function
ϕ(.)
Input space
Feature space
Non -linearly separable
Linearly separable

Train by SVM (1/2)

```
library(e1071)

ind <- sample(2, nrow(iris), replace = T, prob = c(0.7, 0.3))
train.iris <- iris[ind == 1,]
test.iris <- iris[ind == 2,]

model <- svm(Species ~ ., data = train.iris)

print(model)

summary(model)
```

Train by SVM (2/2)

```
plot(cmdscale(dist(train.iris[,-5])),
     col = as.integer(train.iris[,5]),
     pch = c("o","+")[1:150 %in% model$index + 1])

pred <- predict(model, train.iris[,1:4])
table(pred, train.iris[,5])

pred2 <- predict(model, test.iris[,1:4])
table(pred2, test.iris[,5])
```

2. 현장사례

1. 회사 소개
BC Card(주)는 4개의 시중은행 (조흥, 제일, 한미, 하나), 3개의 특수은행 (농협, 기업, 국민), 3개의 지방은행 (대구, 부산, 경남)과 1개의 카드 전업사(우리신용카드)의 신용카드 업무를 지원하는 회사로 주요 내용은 아래와 같다.

2. Data Mining 도입 배경
90년대 중반 이후 정부의 신용카드 사용 장려 정책으로 인하여 신용카드 가입자의 급격한 증가는 카드회사의 매출뿐만 아니라 회원 관리에 대한 비용증가를 가져왔다. 특히 카드 분실에 따른 부정 사용으로 인한 손실은 카드사와 고객의 과실 책임을 가리는 사후적 방법만 존재하는 시점에서 문제의 보다 근원적인 해결 방법의 필요성이 대두되었다.
만약 고객이 카드를 분실하였을 경우, 분실신고 60일 이전까지의 부정사용액에 대해서 고객의 책임은 면제되고 자연적으로 그 손실액은 카드사가 부담하게 된다. 이는 고객 증가와 더불어 분실에 따른 부정사용을 미연에 방지할 수 있는 방법으로 Data Mining 기법의 도입을 결정하였다.

3. Data Mining 대상이 된 문제 또는 자료의 특성
분실카드 부정 사용을 방지하기 위해서 과거 5년간 부정 사용된 적이 있는 카드의 자료(이용시간, 금액, 장소)등을 통해 고객의 거래 정보와 회원특성에 대한 분석을 시도하였다.
위와 같은 분석 작업을 통해 정상적인 사용과 부정적인 사용을 구분 짓는 특성을 변수들을 선정하여 고객의 구매 당시 이전과 패턴에서 크게 벗어날 경우 거래 승인은 잠시 보류되고 카드 소지자와의 전화 통화를 통해서 본인 확인 후 거래가 가능하게 하였다.

4. 적용된 Data Mining 방법
BC Card 사에서 카드 부정사용 적발 (Fraud Detection)을 위해 기본적으로 적용한 유형은 예측(Prediction)이다. 다양한 입력 변수 (거래금액, 사용시간, 장소, 거래간격 등)들 속에서 이 거래가 정상적인 사용인지 부정한 사용인지 예측하는 것이다.
이를 보다 세부적을 살펴보면 신경망 모형 (Neural Networks)을 사용하여 기존의 데이터로부터 반복적인 학습을 통해서 고객의 구매 패턴을 찾아내고 이를 일반화시켜 각 거래에 대해서 부정사용 여부를 예측하게 된다.

5. Data Mining 결과로 인한 경영개선의 내용
카드 부정사용에 대한 예방적 대책이 없던 시점과 비교하여 98년 4월 이 시스템의 도입 이후 부정 사용으로 인한 손실액의 40% 가량이 감소 하였다.

3. 예제 또는 실습

- http://global.oup.com/us/companion.websites/fdscontent/uscompanion/us/static/companion.websites/9780195089653/Spreadsheets/mushroom.csv
- 카네기 멜론 대학에서 제공하는 8124개의 버섯 표본과 23개의 주름 버섯으로 이루어진 데이터
- 분류 알고리즘을 적용하여 식용 가능 버섯인지 독버섯인지 분류

4. 학습진단 평가문제

- 다음 중 의사 결정 나무를 만드는 함수로 알맞지 않은 것은?

① ctree　　② randomForest

③ nnet　　④ rpart

- 다음 중 의미가 다른 하나는?

```
> colnames(iris)
 [1] "Sepal.Length" "Sepal.Width" "Petal.Length" "Petal.Width" "Species"
```

① Iris[5,]　　② iris$Species

③ iris[,-c(1,2,3,4)]　　④ iris[,5]

- 다음 중 신경세포의 구성 요소가 아닌 것은?

① dendrite　② synapse　③ soma　④ dendrogram

회귀 분석 (Regression)

◎ 학습지식 개요/요점

- 수치 데이터를 예측 하기 위한 회귀 분석의 개념을 이해한다.
- 가장 단순한 회귀 분석인 선형 회귀 분석 (Linear Regression)의 개념을 살펴보고, 실행해 본다.
- 선형 회귀 분석의 일반화 버전의 개념을 살펴보고, 실행해 본다.

Contents

회귀 분석 (regression)

- 회귀 분석
  - 독립변인 (input)이 종속변인 (target)에 영향을 미치는지 알아보고자 할 때 실시하는 분석방법
  - 연속형 자료에 따른 연속형 자료의 영향력을 검증할 때 사용

영향을 주는 변수 input	영향을 받는 변수 target	통계분석 방법
범주형 자료	범주형 자료	카이제곱 검정
	연속형 자료	T검정 분산분석
연속형 자료	연속형 자료	회귀분석 구조방정식
	범주형 자료	로지스틱회귀분석

회귀 분석 예시

- 데이터
  - 연속형 변수 : 커피 맛, 가게 인테리어, 직원 친절도 (7점 척도)
  - 목표 : 고객 만족도에 영향을 미치는 변수 파악

  - 독립변수 (input) : 연속형 자료 – 커피의 맛, 가게 인테리어, 직원 친절도
  - 종속변수 (target) : 연속형 자료 – 만족도

회귀 분석의 종료

- 단순 회귀 분석
  - 영향을 주는 변수가 1개
- 다중 회귀 분석
  - 영향을 주는 변수가 2개 이상
- 분석 도구에서는 큰 차이 없음

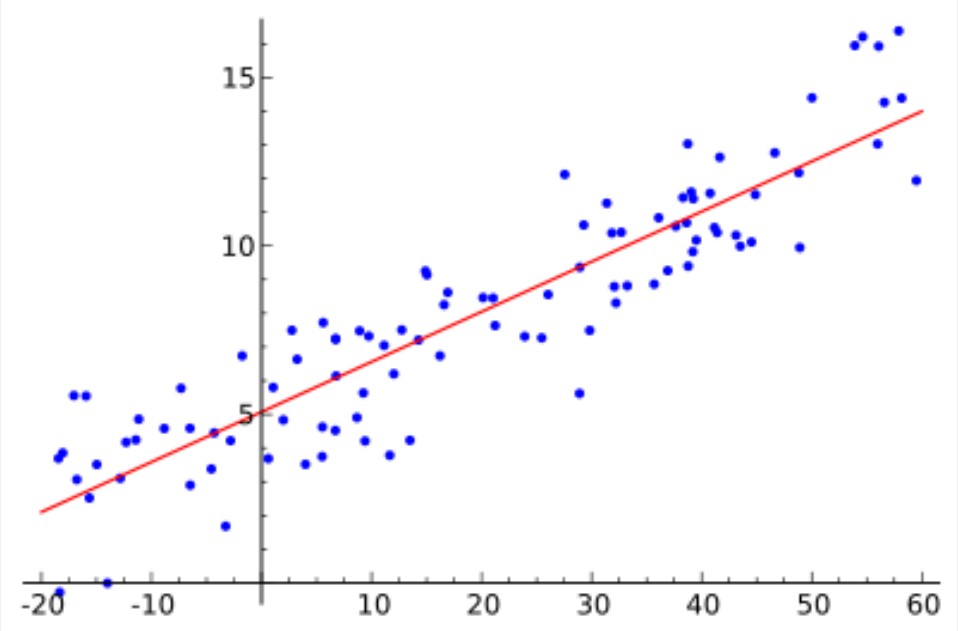
회귀 분석의 결과 분석

- R제곱 (R-Squared)
  - 식의 설명력
  - 독립 변수가 종속 변수를 얼마나 설명하느냐를 판단
- F-Statistics
  - 모형 적합도
  - p-Value가 0.05보다 작으면 이 모형이 적합함

Contents

Linear Regression

- Linear Regression은 다음과 같은 선형 방정식을 이용하여 목표 값을 예측하는 것이다.
- $y = c_0 + c_1 x_1 + c_2 x_2 + \cdots + c_k x_k$
- Target(y)와 Input($x_1, x_2, \ldots, x_k$)을 이용하여 $c_0, c_1, c_2, \ldots, c_k$을 찾아내는 것이 목표



The CPI Data

- Australian CPI (Consumer Price Index) data

```
year <- rep(2008:2010, each = 4)
quarter <- rep(1:4, 3)
cpi <- c(162.2,164.6,166.5,166,166.2,167,168.6,169.5,171,172.1,173.3,174)
plot(cpi, xaxt = "n", ylab = "CPI", xlab = "")
axis(1, labels = paste(year, quarter, sep = "Q"), at = 1:12, las = 3)
```

Train by Linear Regression (1/2)

```
cor(year, cpi)
cor(quarter, cpi)
fit <- lm(cpi ~ year + quarter)
fit

cpi2011 <- fit$coefficients[[1]]+fit$coefficients[[2]]*2011 + fit$coefficients[[3]]*(1:4)
cpi2011
```

$$cpi = -7644.487500 + 3.887500 * year + 1.166667 * quarter$$

Train by Linear Regression (2/2)

attributes(fit)

fit$coefficients

residuals(fit)

summary(fit)

Prediction of CPIs in 2011

```
data2011 <- data.frame(year = 2011, quarter = 1:4)
cpi2011 <- predict(fit, newdata = data2011)
style <- c(rep(1,12), rep(2,4))
plot(c(cpi, cpi2011), xaxt = 'n', ylab = 'CPI', xlab = '', pch = style, col = style)
axis(1, at = 1:16, las = 3,
     labels = c(paste(year, quarter, sep='Q'), '2011Q1', '2011Q2', '2011Q3','2011Q4'))
```

Contents

Generalized Linear Model (GLM)

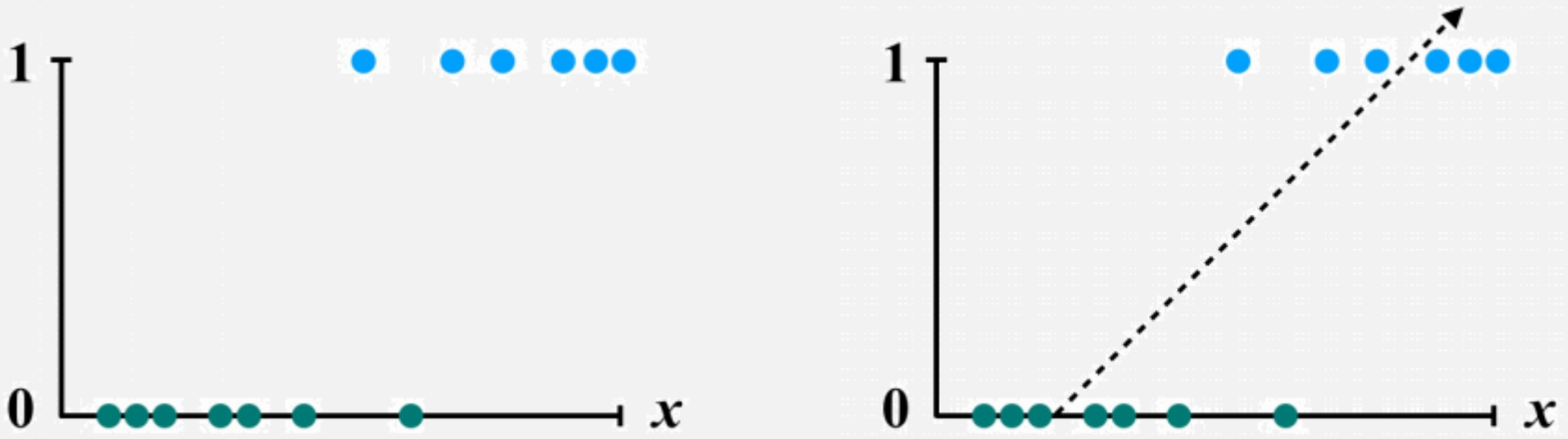
- GLM
  - Linear Regression: 종속변수(target)의 정규 분포와 분산의 동등성 가정
  - GLM: 자료의 독립성 가정
  - 광범위한 비정규분포 자료의 사용 허용
  - Input과 target의 관계가 선형 및 비선형인 경우에도 연결 함수 (link function)을 이용하여 모형의 선형성 충족

Generalized Linear Model

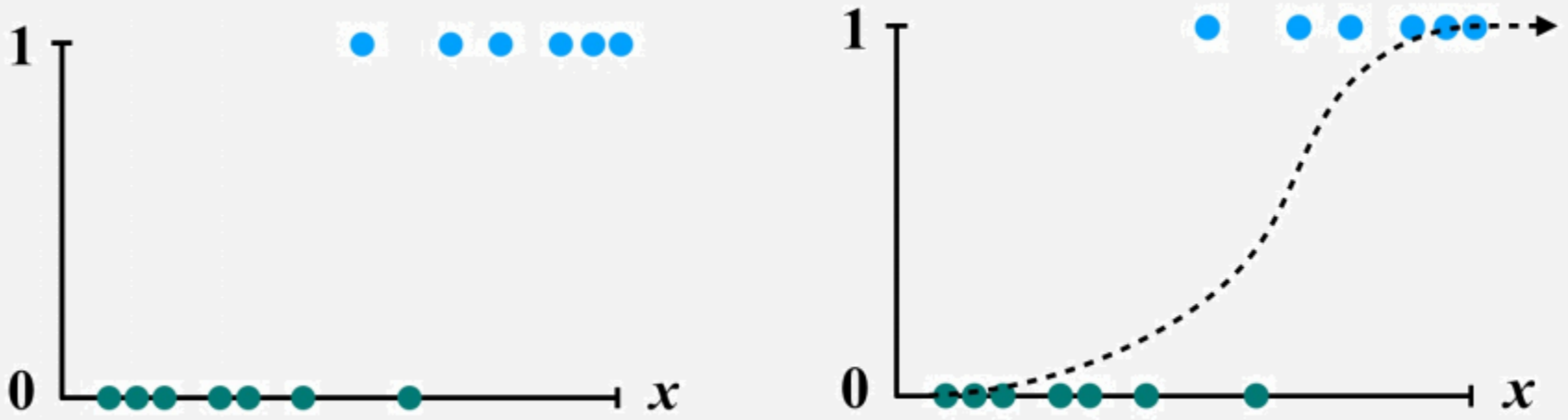
```
data('bodyfat', package='TH.data')
myFormula <- DEXfat ~ age + waistcirc + hipcirc + elbowbreadth + kneebreadth
bodyfat.glm <- glm(myFormula, family = gaussian('log'), data = bodyfat)
summary(bodyfat.glm)

pred <- predict(bodyfat.glm, type = 'response')
plot(bodyfat$DEXfat, pred, xlab = 'Observed', ylab = 'Prediction')
abline(a = 0, b = 1, col = 'red', lwd = 2)
```

Contents

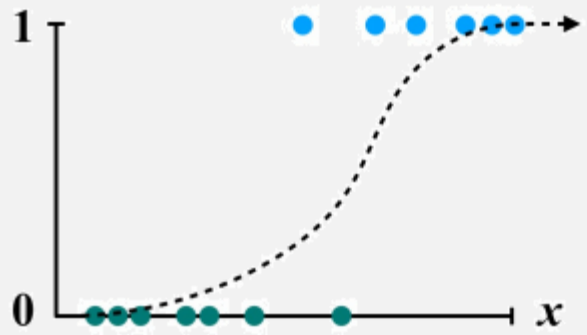
Regression for binary classification



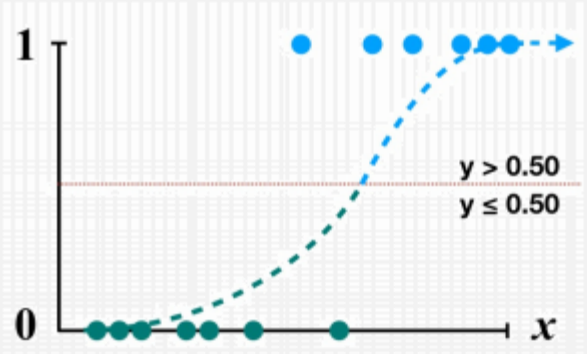
Introducing logistic regression



Making prediction with logistic regression




```
m <- glm(y ~ x1 + x2 + x3,
         data = my_dataset,
         family = "binomial")
```




```
prob <- predict(m, test_dataset,
          type = "response")
```

```
pred <- ifelse(prob > 0.50, 1, 0)
```

Build Logistic Regression Model

```
# Examine the dataset to identify potential independent variables
str(donors)

# Explore the dependent variable
table(donors$donated)

# Build the donation model
donation_model <- glm(donated ~ bad_address + interest_religion + interest_veterans,
                      data = donors, family = "binomial")

# Summarize the model results
summary(donation_model)
```

Binary Prediction

```
# Estimate the donation probability
donors$donation_prob <- predict(donation_model, type = "response")

# Find the donation probability of the average prospect
mean(donors$donated)

# Predict a donation if probability of donation is greater than average (0.0504)
donors$donation_pred <- ifelse(donors$donation_prob > 0.0504, 1, 0)

# Calculate the model's accuracy
mean(donors$donated == donors$donation_pred)
```

2. 현장사례

- 경제성장률, 1인당 소득변화 등에 따른 수요 변화 정도 측정
- 양극화가 자동차 수요에 미치는 영향 정도 파악
- 글로벌 경제위기로 인해 향후 세계경제 전망이 명확하지 않은 경우, 몇 가지 시나리오에 대한 영향을 선제적으로 파악
- 품목별 소득-수요의 관계를 회귀분석으로 추정
- 인터넷 사용자와 1인당 GDP, 기온과 음료수 판매량, 세계경제 성장이 제품 수요에 미치는 영향 등

3. 예제 또는 실습

- https://archive.ics.uci.edu/ml/machine-learning-databases/concrete/compressive/Concrete_Data.xls
- 콘크리트 데이터 셋
- 1,033개의 콘크리트 예제를 포함
- 섞을 때 사용하는 구성 요소를 나타내는 8개의 속성으로 최종 압축 내구력과 관계가 있음

4. 학습진단 평가문제

- 다음 중 다른 것과 속성이 다른 것은?

① 카이제곱 검정　　② 분산 분석

③ 회귀 분석　　④ 구조 방정식

- Random number를 일정하게 유지하기 위해 사용하는 함수는?

① kmeans()　　② set.seed()

③ Set.Seed()　　④ plyr()

- Binary 데이터를 분류할 수 있는 함수는?

① rpart()　　② table()

③ glm()　　④ lm()

군집화 (Clustering)

◎ 학습지식 개요/요점

- 컴퓨터 임의로 데이터를 구분하는 군집화의 개념을 확인한다.
- Top-down 방법인 Partitioning Clustering의 개념을 살펴보고, 실행해 본다.
- Bottom-up 방법인 Hierarchical Clustering의 개념을 살펴보고, 실행해 본다.

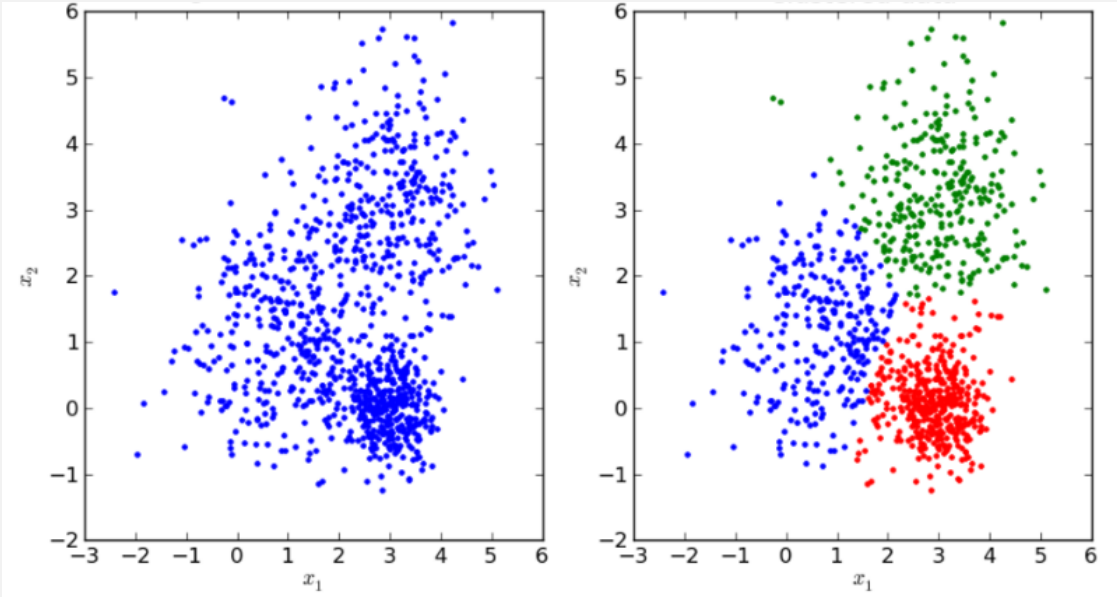
Contents

비 지도학습이란?

- 학습을 할 때 입력 값(독립변수)만 제공하고, 입력 값 사이의 관계나 패턴을 학습하는 모델을 생성하여 입력 값에 정답을 부여하는 방법
- 대표적으로 주어진 데이터 입력 값의 패턴으로 여러 개의 클러스터를 생성 후, 입력 값이 속하는 클러스터(정답) 번호를 부여하는 클러스터링 방법이 있음
- 예시
  - 고객 성향 세그멘테이션
  - 공정 이상치 탐지
  - 영화 컨텐츠 카테고리 분류

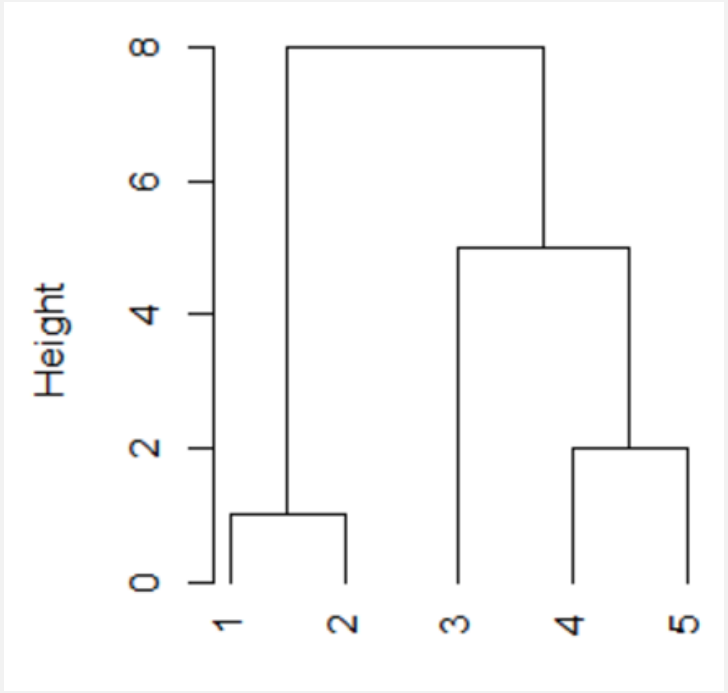
군집분석(Clustering analysis) 의미

- 개체 간 유사성(Similarity)을 계산하여 유사성이 높은 개체를 군집으로 생성하는 비지도 학습 분석 방법
- 동일한 군집에 속하는 개체들은 유사성이 높고, 동일하지 않은 군집에 속하는 개체는 유사성이 낮은 특징을 보이게 됨
- 지도학습과는 다르게 개체에 대한 정답이 존재하지 않는 상황에서 적용 가능한 방법



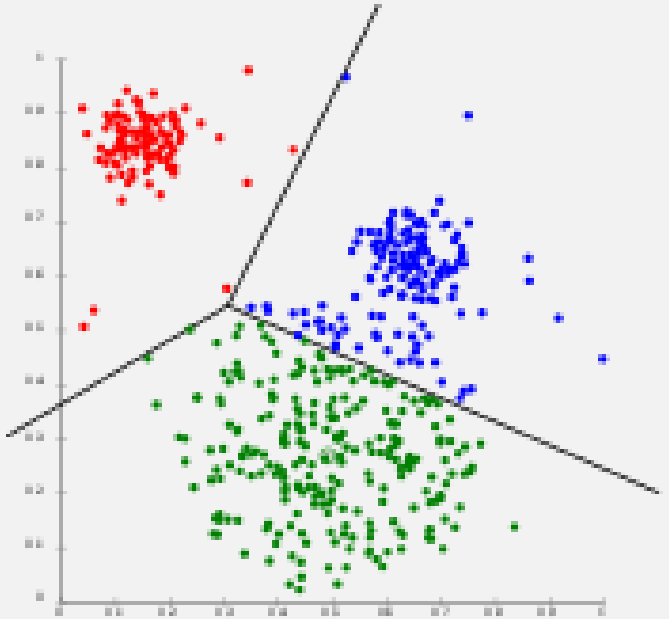
군집분석(Clustering analysis) 종류

- 계층적 군집분석(Hierarchical methods)
  - 각 개체마다 하나의 군집이 부여되어 시작되며, 단계별로 기준에 따라 유사한 군집 간의 통합을 반복 수행하는 군집 분석 방법



군집분석(Clustering analysis) 종류

- Partitioning Clustering (Centroid-based clustering)
  - 클러스터는 중앙 벡터로 표현되며, 클러스터 수를 k로 고정하면, k 클러스터 중심을 찾아 가장 가까운 클러스터 중심에 객체를 할당



Contents

K-means clustering

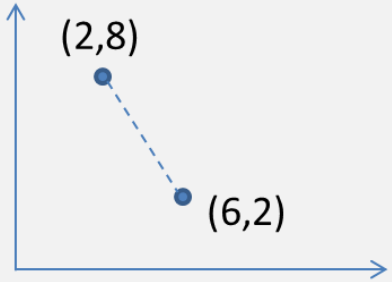
- K-MEANS 군집분석(K-MEANS CLUSTERING) 의미
  - 비계층적 군집분석의 대표적인 군집 분석 방법
  - 주어진 데이터를 K개의 군집으로 묶는 군집 분석으로, 각 군집 내의 데이터들의 거리를 최소화, 각 군집간 데이터들의 거리를 최대화 시키는 방법으로 군집화를 실시함
  - 레이블(정답)이 없는 데이터를 대상으로 수행하며, 분석 결과 각 데이터가 속하는 군집이 부여됨

K-means clustering

- K-MEANS 군집분석 기본 용어
  - 중심점(Centroid)
    - 각 군집의 중심이 되는 점
  - 거리(Distance)
    - 데이터 간 또는 데이터와 군집의 중심 간의 거리
    - 대표적으로 유클리디언 거리(Euclidean distance)가 많이 쓰임
  - K
    - 사전에 정의되는 값
    - 군집 분석 후 최종 할당되는 군집의 수를 의미

K-means clustering

- 유클리디언 거리(Euclidean distance)
  - 두 데이터 사이의 거리를 계산할 수 있는 대표적인 방법
    - $\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$ n :각 점이 가지고 있는 속성 (변수)를 의미
  - 예제



$(x_1, x_2) = (2,8)$
$(y_1, y_2) = (6,2)$

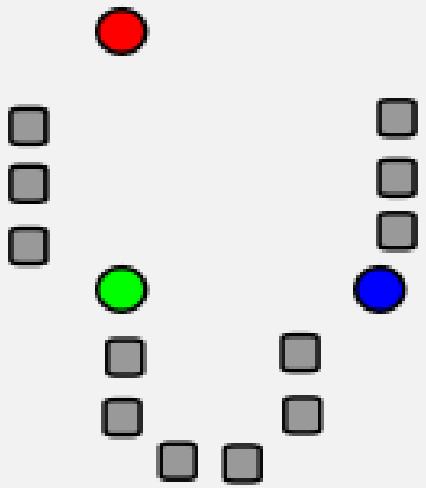
Distance = $\sqrt{(2-6)^2+(8-2)^2} = 7.21$

$(x_1, x_2, x_3, x_4,) = (2,8,1,7)$
$(y_1, y_2, y_3, y_4) = (6,2,3,5)$

Distance = $\sqrt{(2-6)^2+(8-2)^2+(1-3)^2+(7-5)^2} = 7.75$

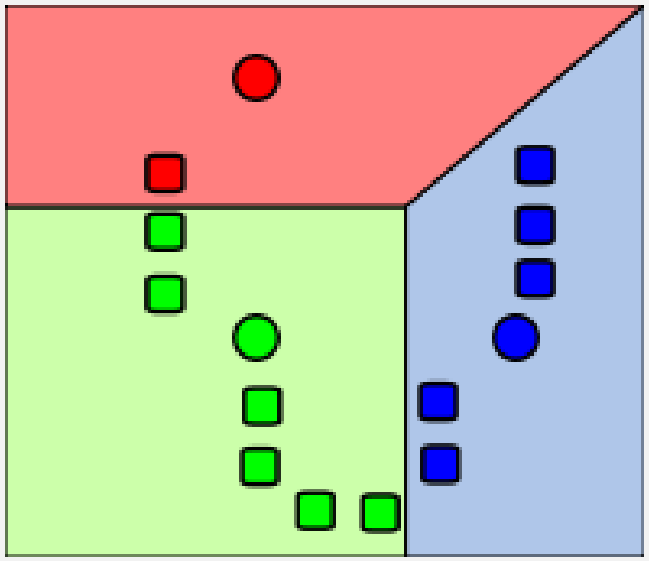
K-means clustering

- K-means clustering 알고리즘 상세
  - Step 1: 데이터 중 임의로 중심점(Centroid)를 3개 만큼 선정한다. (k=3)



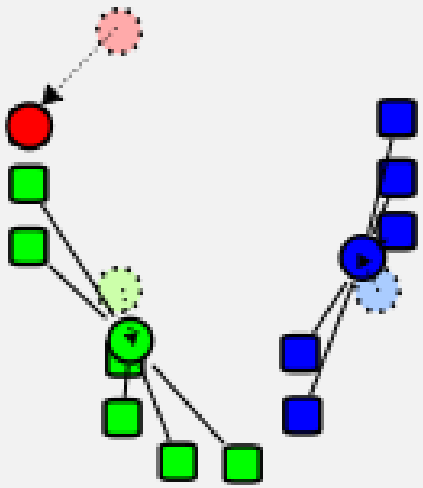
K-means clustering

- K-means clustering 알고리즘 상세
  - Step 2: 각 데이터를 가장 가까운 중심점의 군집에 할당



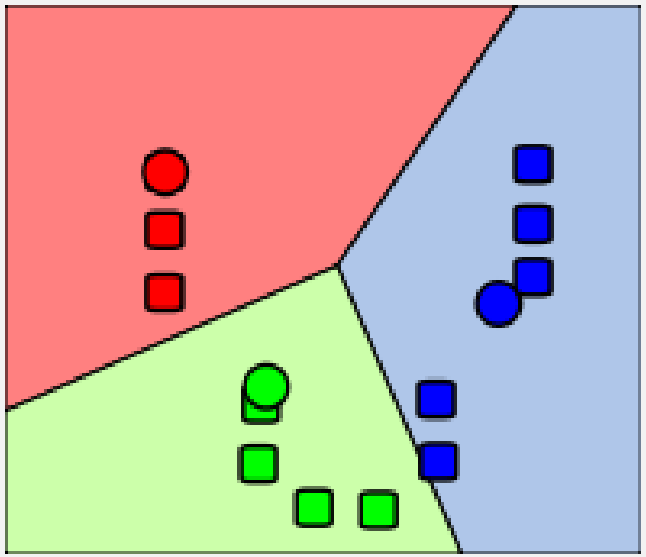
K-means clustering

- K-means clustering 알고리즘 상세
  - Step 3: 3개의 군집에 속한 데이터들의 중심점을 재 조정한다.



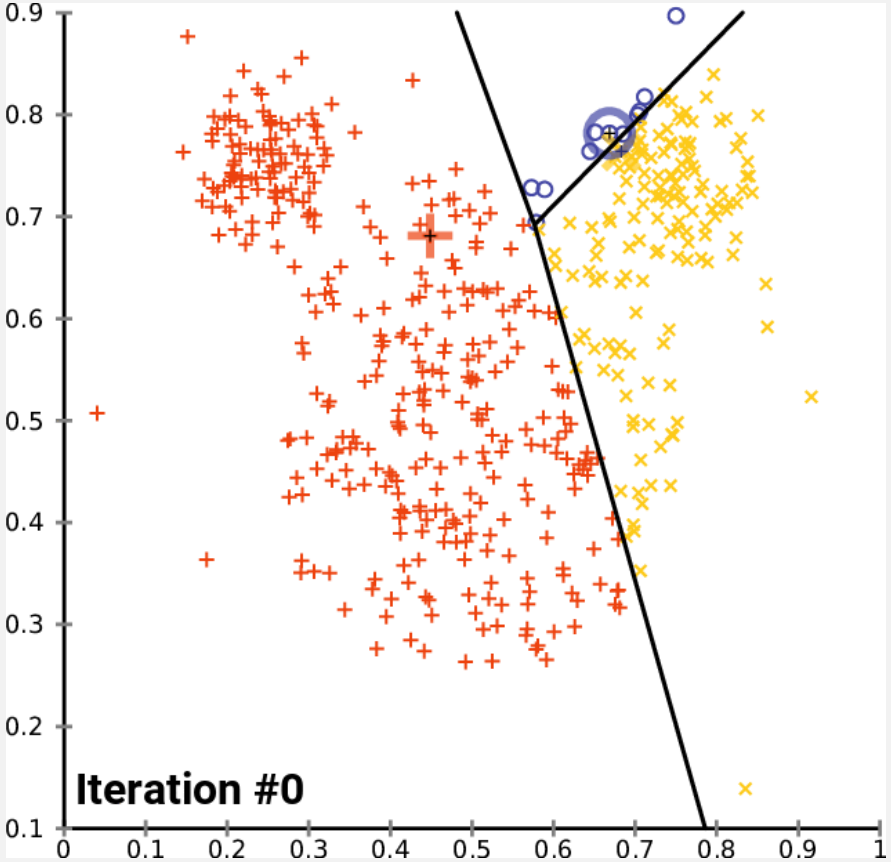
K-means clustering

- K-means clustering 알고리즘 상세
  - Step 4: 각 군집에 속하는 데이터들이 변하지 않을 때까지 Step2, Step3 을 반복



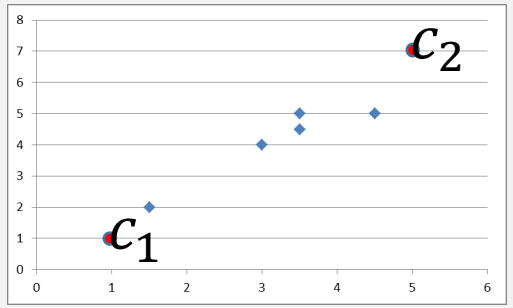
K-means clustering

- K-means clustering 알고리즘



K-means clustering

- 12개의 데이터에 대한 K-MEANS 군집 분석 실시 (K=2)



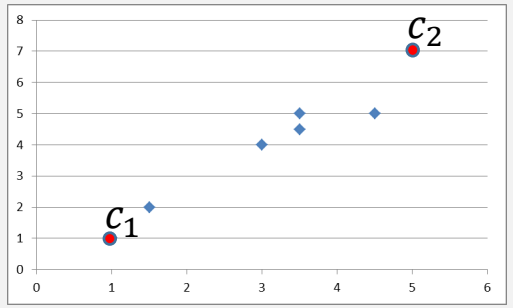
데이터	A	B
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

Step 1

$$c_1 = (1,1), c_2 = (5,7)$$

K-means clustering

- 12개의 데이터에 대한 K-MEANS 군집 분석 실시 (K=2)



데이터	A	B
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

Step 2,3

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j$$

	군집 1		군집 2	
Step	데이터	C1	데이터	C2
1	1	(1.0, 1.0)	4	(5.0, 7.0)
2	1, 2	(1.2, 1.5)	4	(5.0, 7.0)
3	1, 2, 3	(1.8, 2.3)	4	(5.0, 7.0)
4	1, 2, 3	(1.8, 2.3)	4	(5.0, 7.0)
5	1, 2, 3	(1.8, 2.3)	4, 5	(4.2, 6.0)
6	1, 2, 3	(1.8, 2.3)	4, 5, 6	(4.3, 5.7)
7	1, 2, 3	(1.8, 2.3)	4, 5, 6, 7	(4.1, 5.4)

	데이터	중심점
군집1	1, 2, 3	(1.8, 2.3)
군집2	4, 5, 6, 7	(4.1, 5.4)

K-means clustering

- 12개의 데이터에 대한 K-MEANS 군집 분석 실시 (K=2)

Step 2,3 반복

데이터	군집	군집1 중심점과의 거리	군집2 중심점과의 거리
1	군집1	1.5	5.4
2	군집1	0.4	4.3
3	군집1	2.1	1.8
4	군집2	5.7	1.8
5	군집2	3.2	0.7
6	군집2	3.8	0.6
7	군집2	2.8	1.1

	데이터	중심점
군집1	1, 2, 3	(1.8, 2.3)
군집2	4, 5, 6, 7	(4.1, 5.4)

K-means clustering

- 12개의 데이터에 대한 K-MEANS 군집 분석 실시 (K=2)

Step 2,3 반복

데이터	군집	군집1 중심점과의 거리	군집2 중심점과의 거리
1	군집1	1.5	5.4
2	군집1	0.4	4.3
3	군집1 -> 군집2	2.1	1.8
4	군집2	5.7	1.8
5	군집2	3.2	0.7
6	군집2	3.8	0.6
7	군집2	2.8	1.1

	데이터	중심점
군집1	1, 2, 3	(1.8, 2.3)
군집2	4, 5, 6, 7	(4.1, 5.4)

K-means clustering

- 12개의 데이터에 대한 K-MEANS 군집 분석 실시 (K=2)

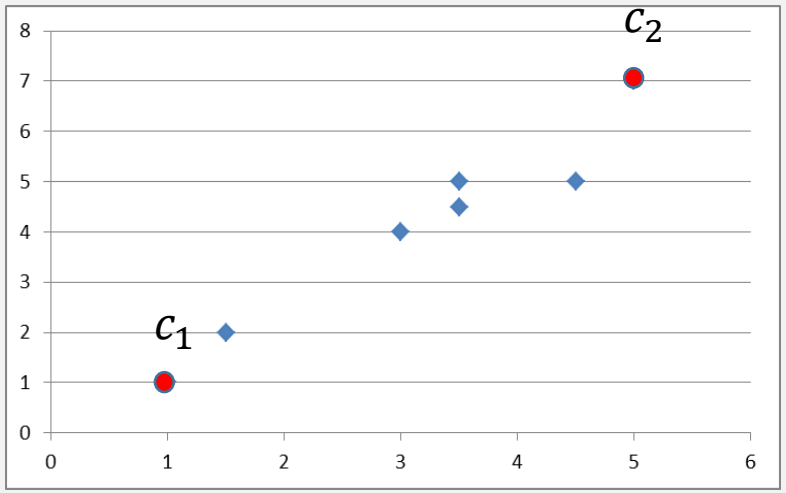
Step 2,3 반복

데이터	군집	군집1 중심점과의 거리	군집2 중심점과의 거리
1	군집1	0.6	5.0
2	군집1	0.5	3.9
3	군집2	3.0	1.4
4	군집2	6.6	2.2
5	군집2	4.1	0.4
6	군집2	4.7	0.6
7	군집2	3.7	0.7

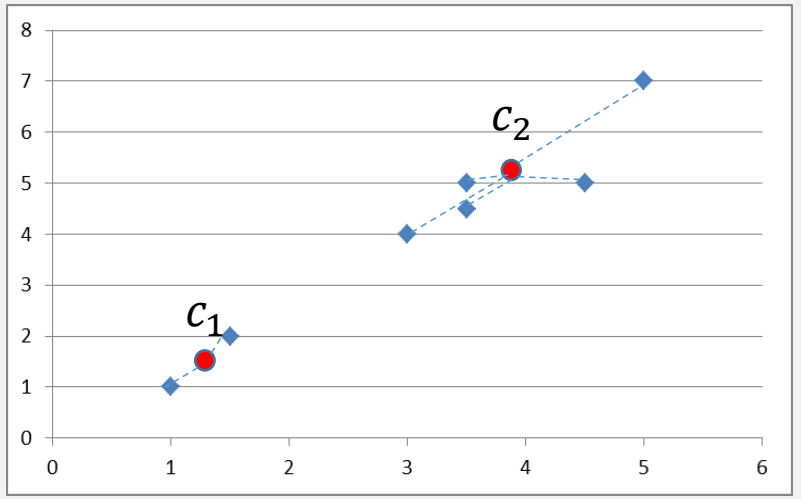
	데이터	중심점
군집1	1, 2	(1.3, 1.5)
군집2	3,4, 5, 6, 7	(3.9, 5.1)

K-means clustering

- 12개의 데이터에 대한 K-MEANS 군집 분석 실시 (K=2)



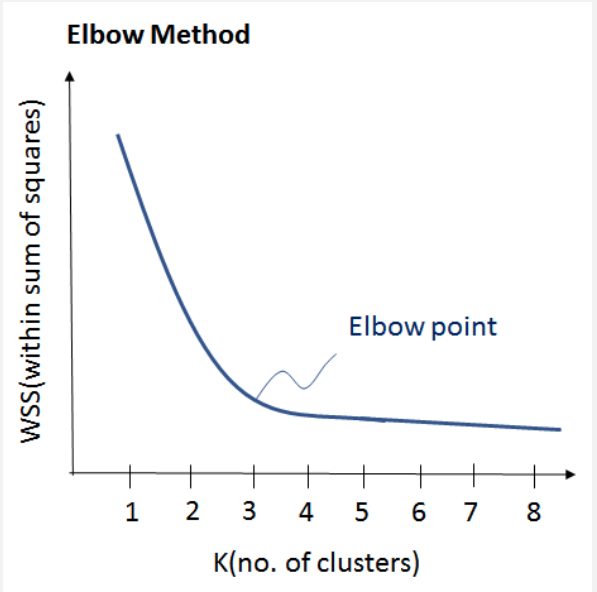
$c_1 = (1,1), c_2 = (5,7)$



$c_1 = (1.3, 1.5), c_2 = (3.9, 5.1)$

K-means clustering

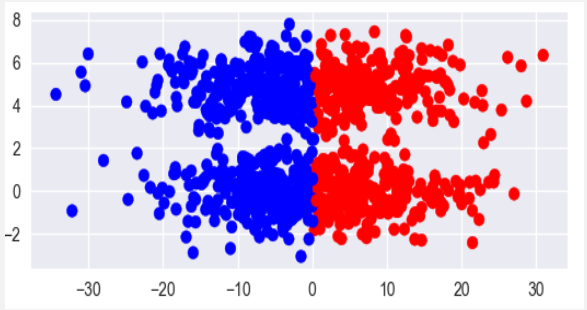
- K 값 결정
  - K 값은 사전에 정의되어야 하는 값으로써, K 값에 따라 각 데이터가 속하는 군집이 상이할 수 있음
  - K 값은 분석목적 및 K-MEANS 분석의 결과를 함께 반영하여 설정할 수 있으며, 분석결과만으로 판단하는 대표적인 방법으로는 팔꿈치 방법(Elbow method)이 있음
  - K 값을 다양하게 설정하여 군집 분석을 수행한 후, 각 군집 내 데이터들과 중심점 간의 거리 합계(WSS)가 더 이상 유의미하게 감소하지 않는 K 값을 선택하는 방법



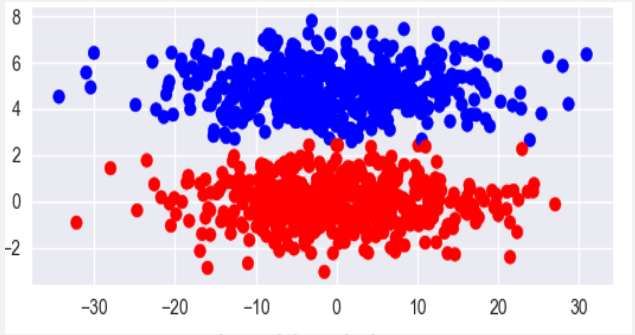
K-means clustering

- 데이터 정규화 (Data normalization)
  - 연속형 데이터가 다양한 단위로 존재할 경우, 데이터간 유클리디언 거리 계산시 의도하지 못한 왜곡이 발생할 수 있음
  - 사전에 데이터 정규화를 통해 데이터를 동일 단위로 설정한 후 군집화를 실시

ex 나이 (0 ~ 10세), 키(40cm ~ 120cm)로 이루어진 데이터의 경우 바로 유클리디언 거리를 구할 경우 대부분 나이에 상관없이 키에 따라 거리가 결정되는 왜곡이 발생함



[정규화 이전]



[정규화 이후]

$$X_{normalized} = \frac{X - \mu}{\sigma}$$

$$X_{standardized} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

K-means clustering

- K-means clustering

```
set.seed(8953)
iris2 <- iris
iris2$Species <- NULL
(kmeans.result <- kmeans(iris2, 3))
```

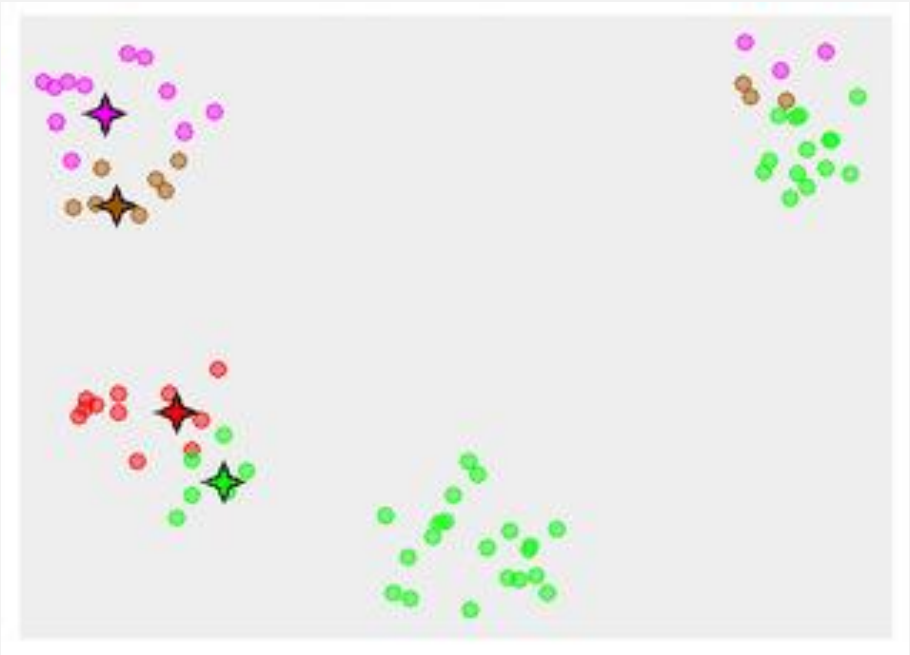
- Result

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

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)
```

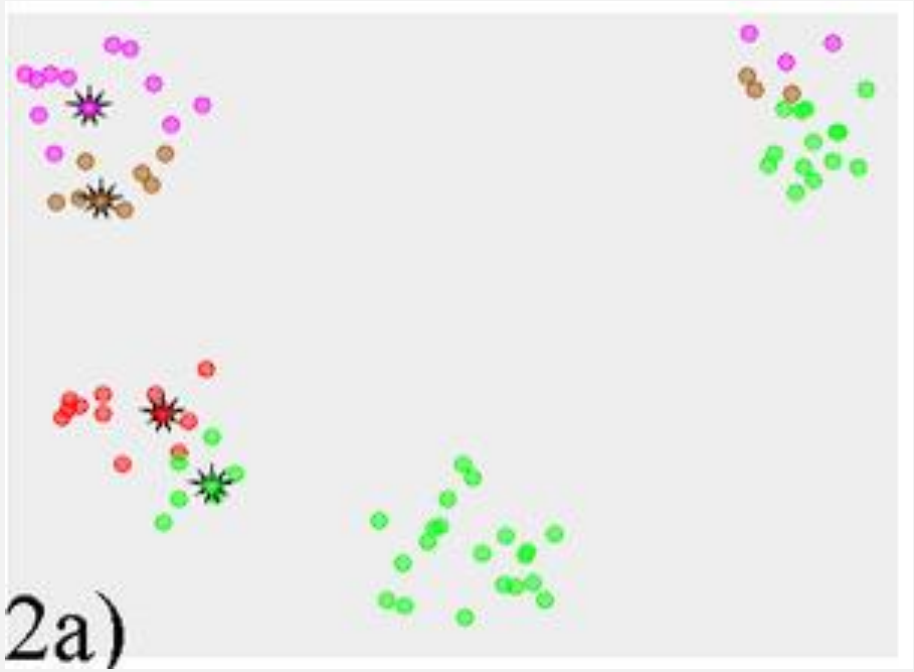
K-Medoids Clustering

- K-means와의 차이점 : 클러스터는 K-means 알고리즘에서 중심을 나타내지 만 K-medoids 클러스터링에서는 클러스터의 중심에 가장 가까운 객체
- 이상 치의 존재 하에서 K-means 보다 더 견고
- PAM (Partitioning Around Medoids)은 K-medoids 클러스터링을 위한 고전적인 알고리즘
- 단점 :
  - 느림 : O(k * pow((n-k), 2))
  - K-means : O(tkn) ~ O(n)

k-medoids vs. k-means

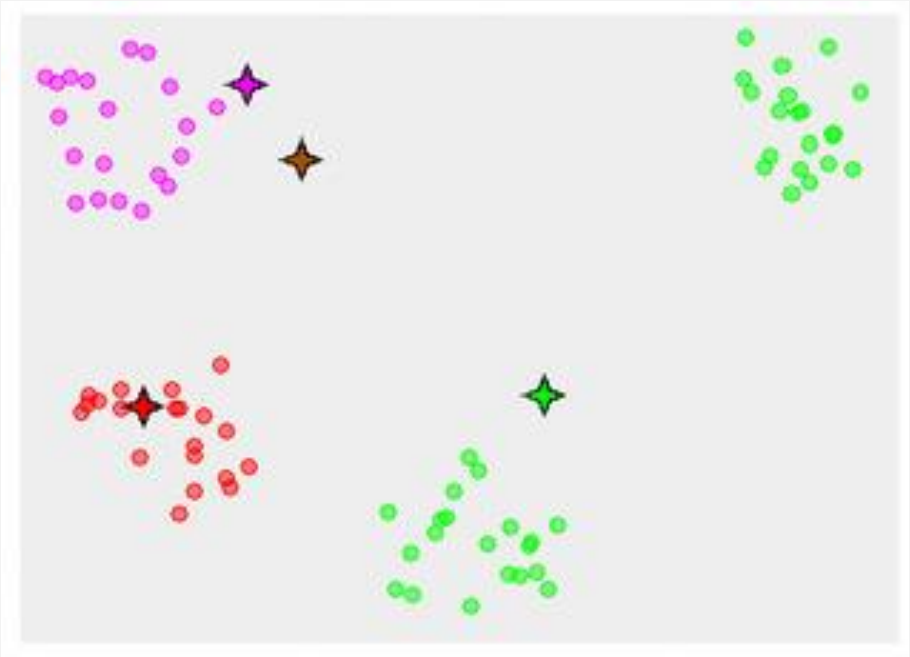


K-Means : iter 1

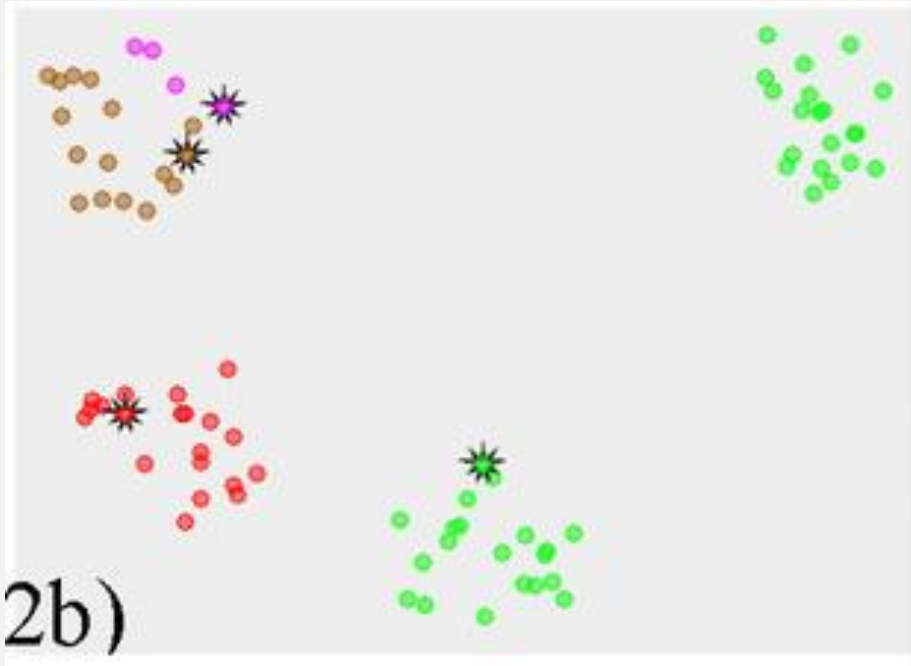


K-Medoids : iter 1

k-medoids vs. k-means

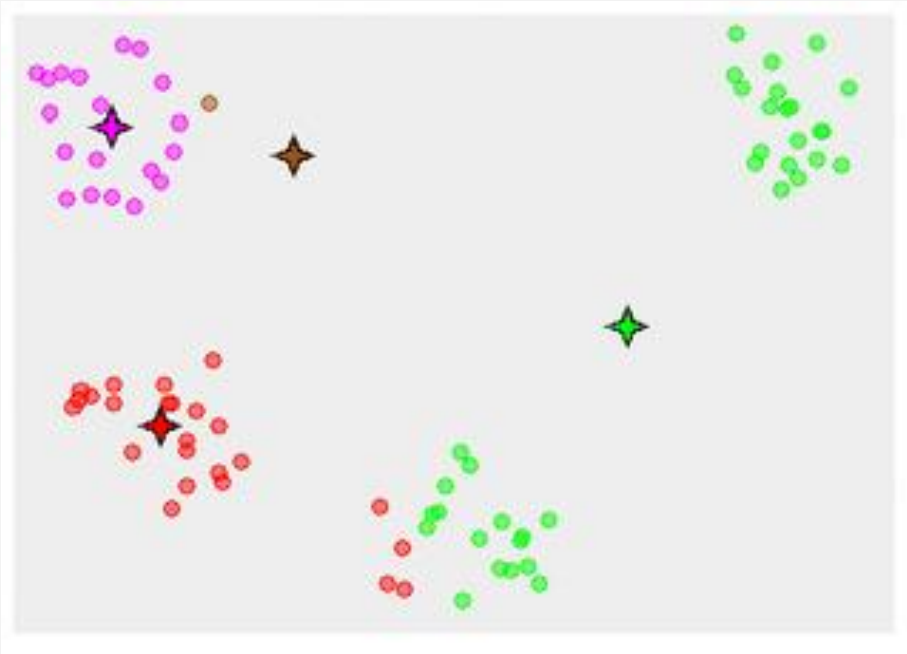


K-Means : iter 2

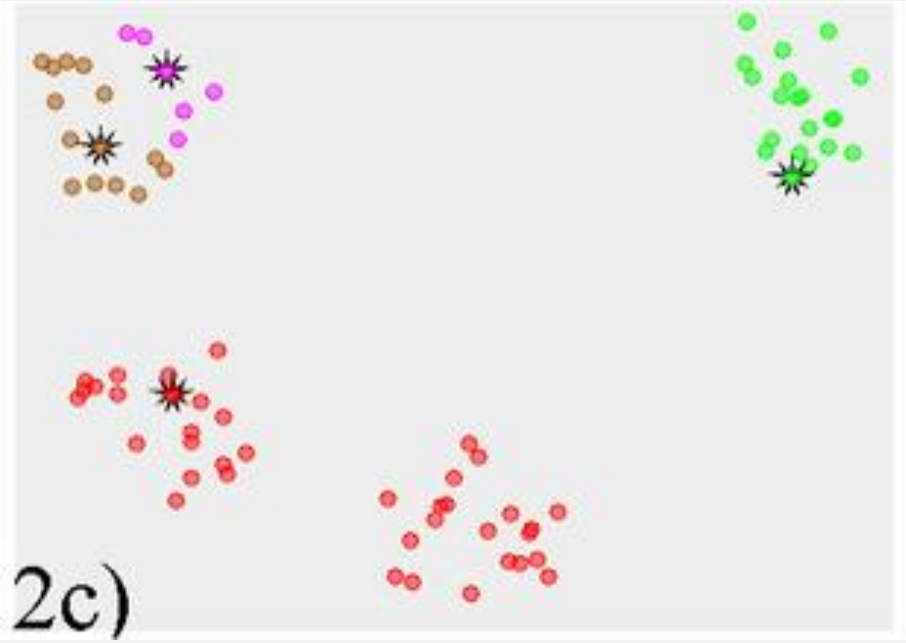


K-Medoids : iter 2

k-medoids vs. k-means

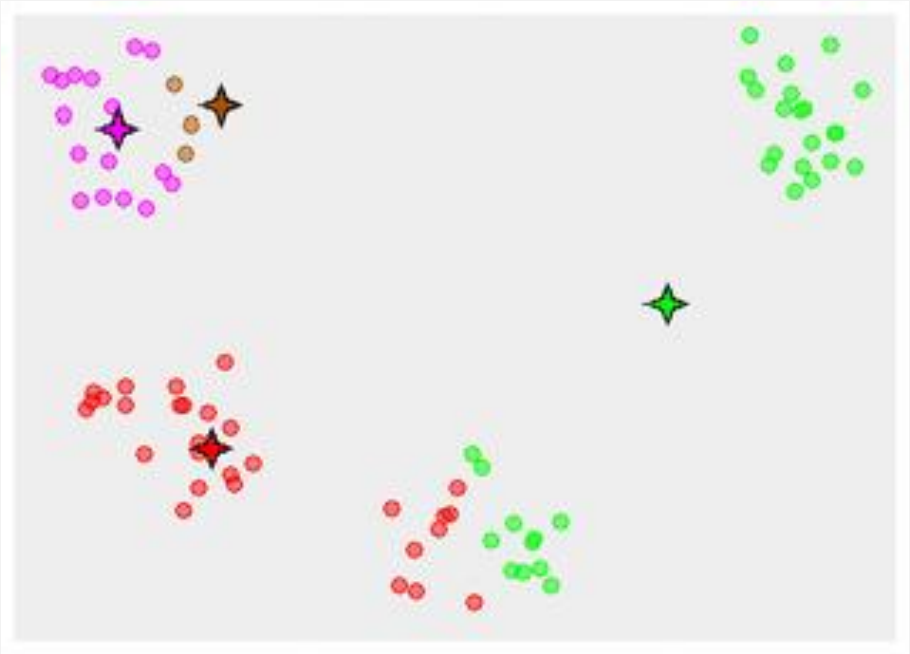


K-Means : iter 3

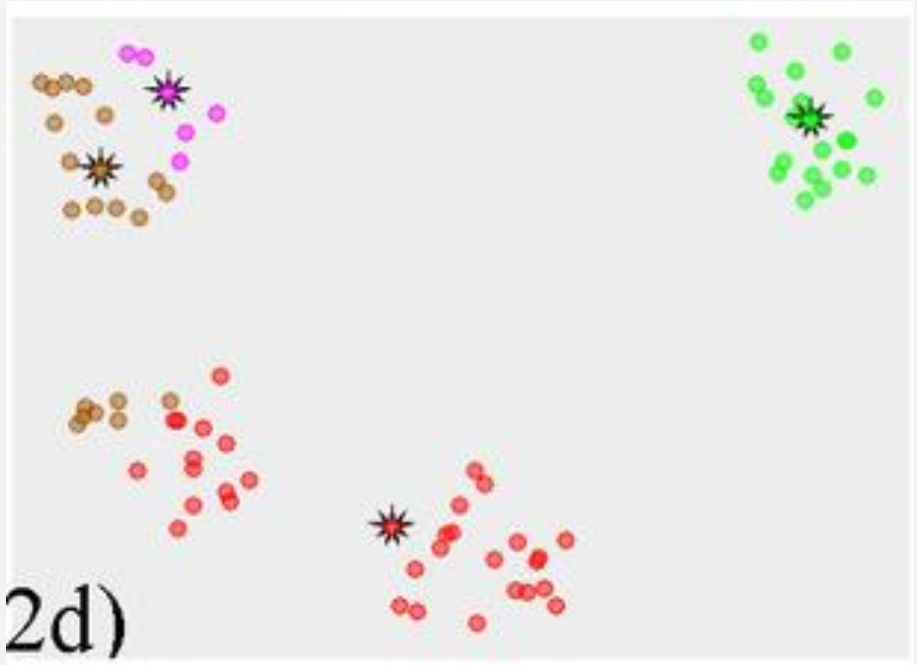


K-Medoids : iter 3

k-medoids vs. k-means

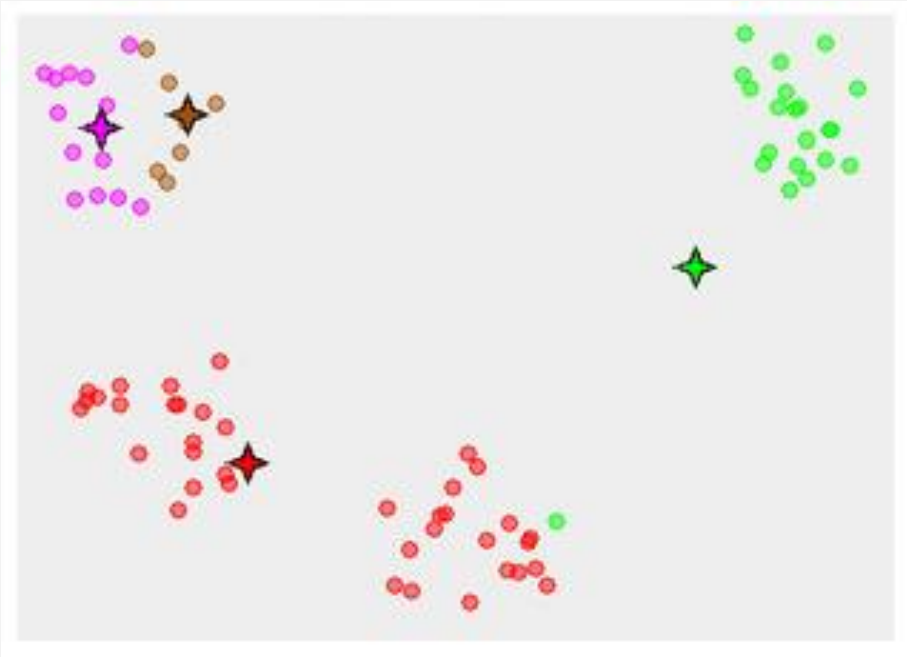


K-Means : iter 4

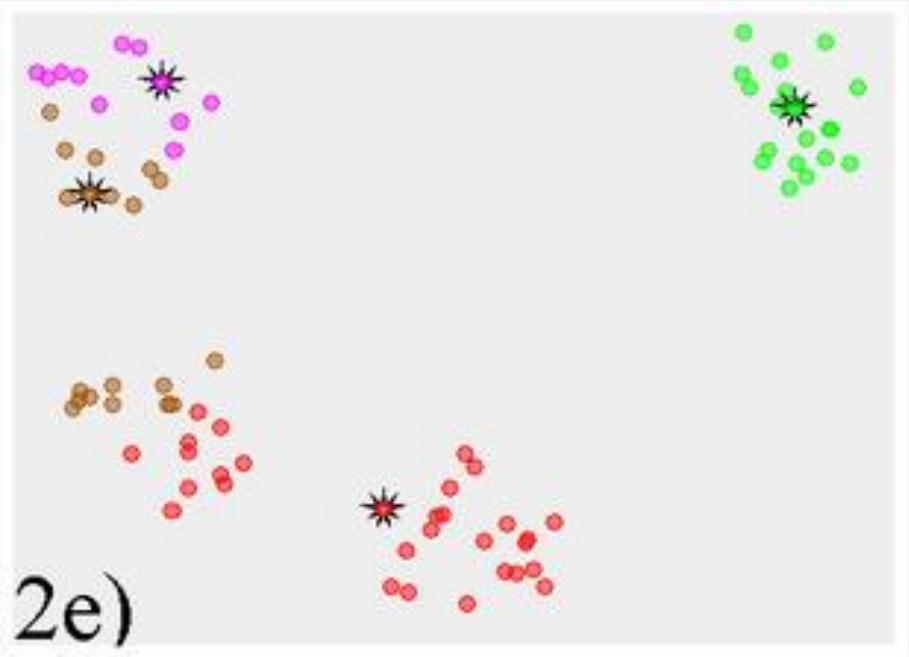


K-Medoids : iter 4

k-medoids vs. k-means

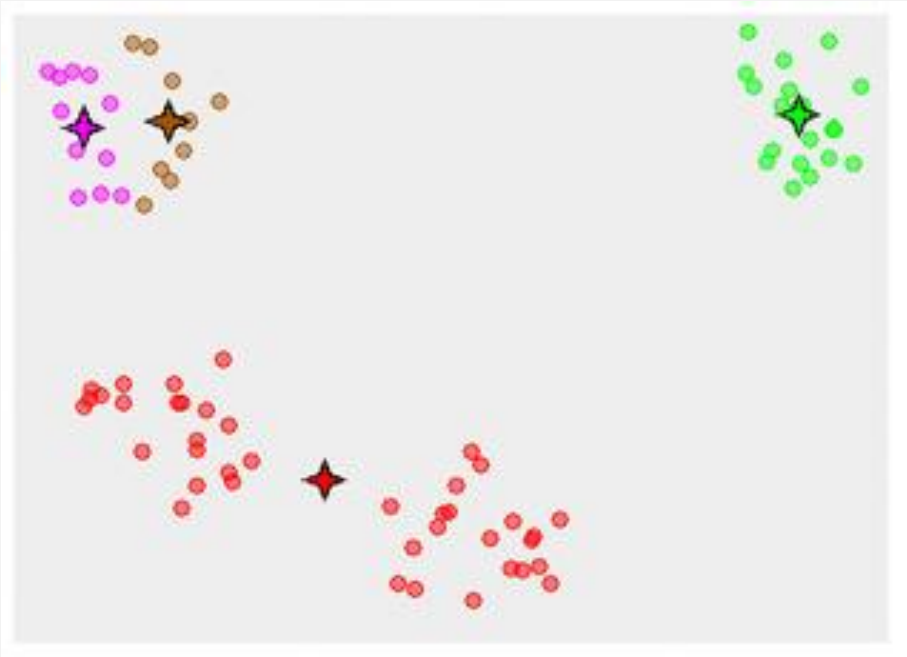


K-Means : iter 5

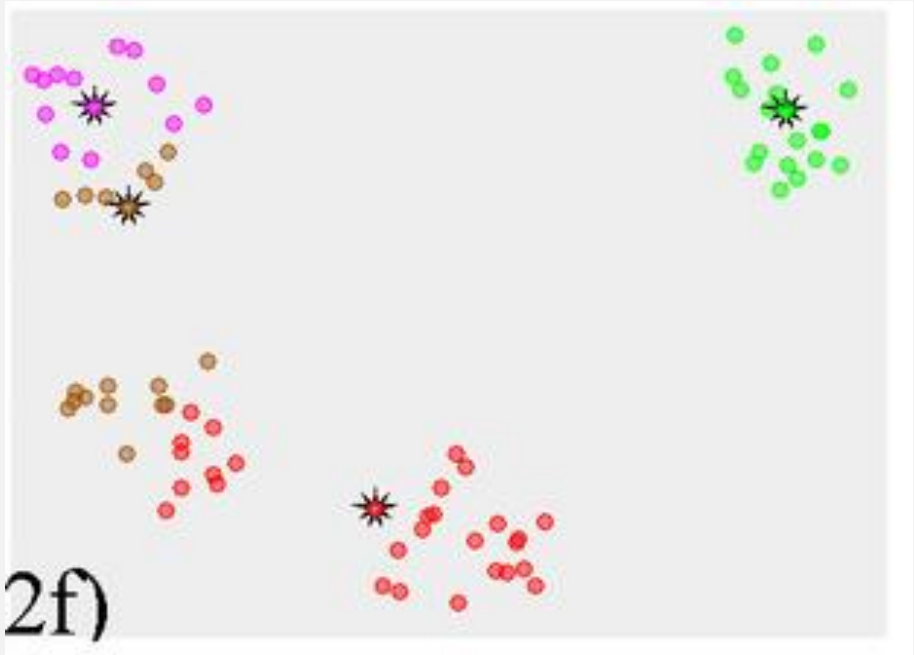


K-Medoids : iter 5

k-medoids vs. k-means

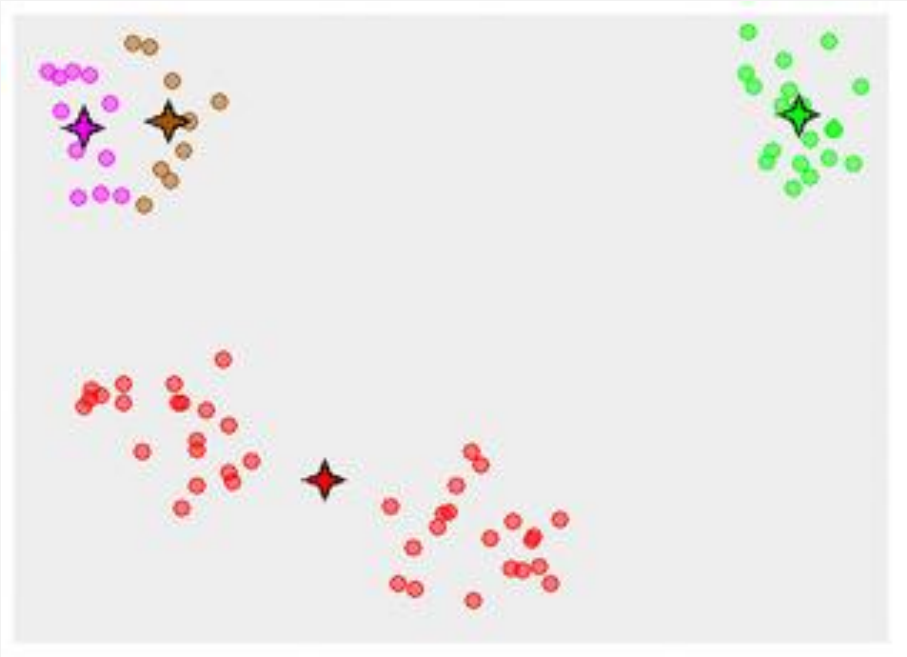


K-Means : iter 6

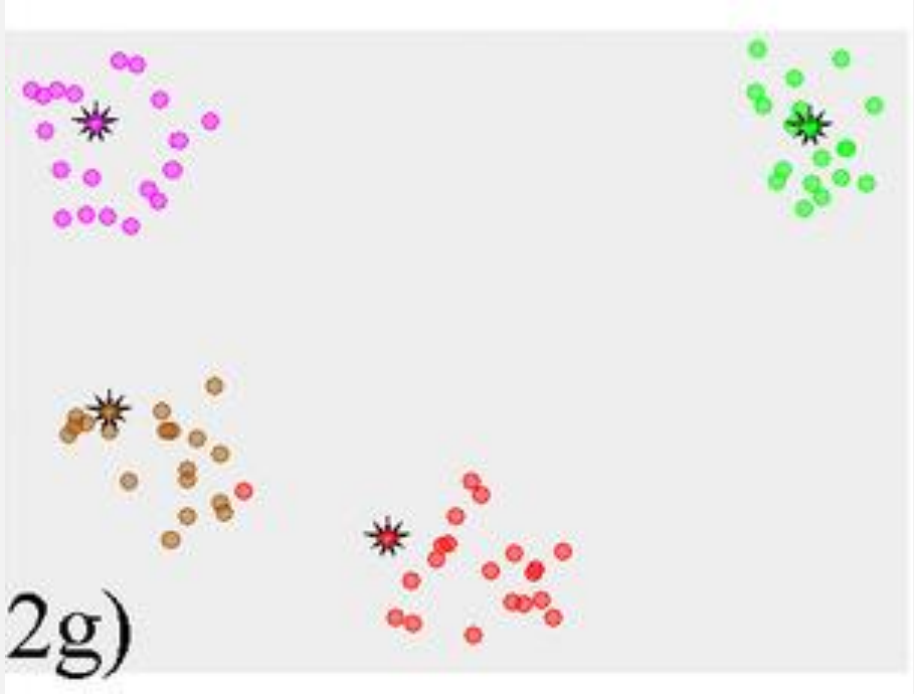


K-Medoids : iter 6

k-medoids vs. k-means

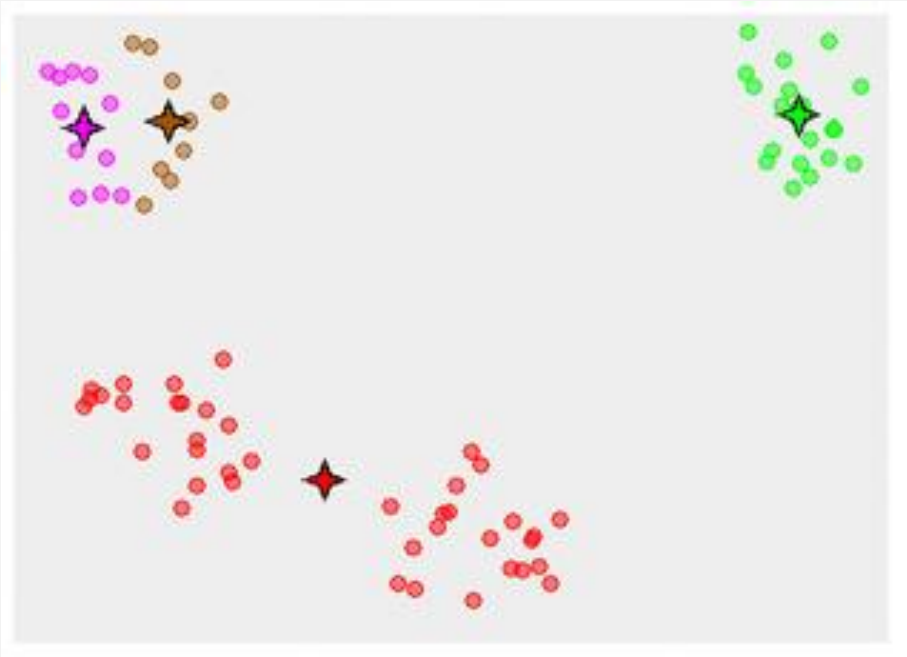


K-Means : iter 6

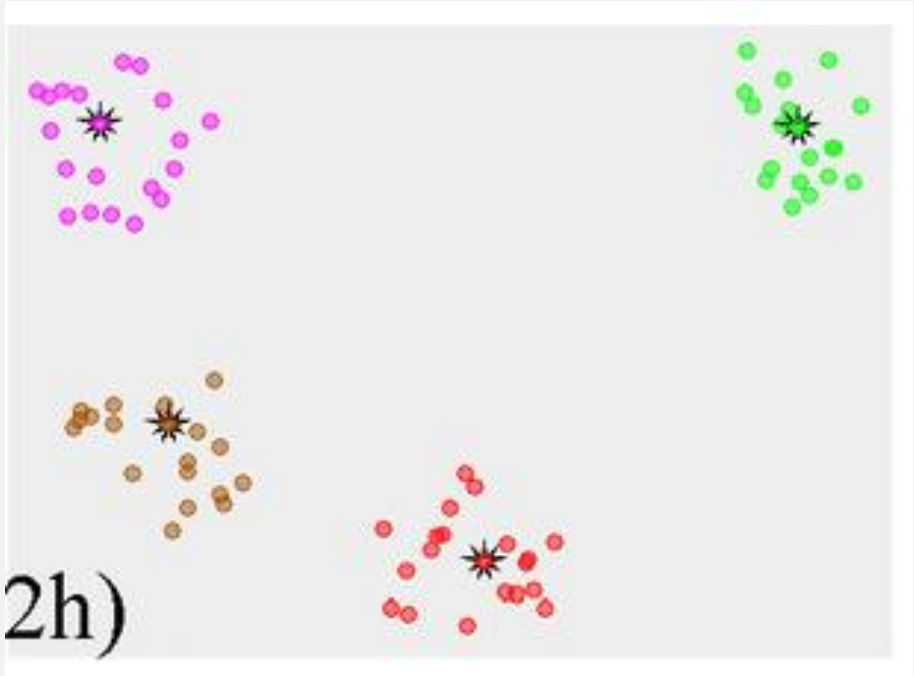


K-Medoids : iter 7

k-medoids vs. k-means



K-Means : iter 6



K-Medoids : iter 8

Clustering with pamk()

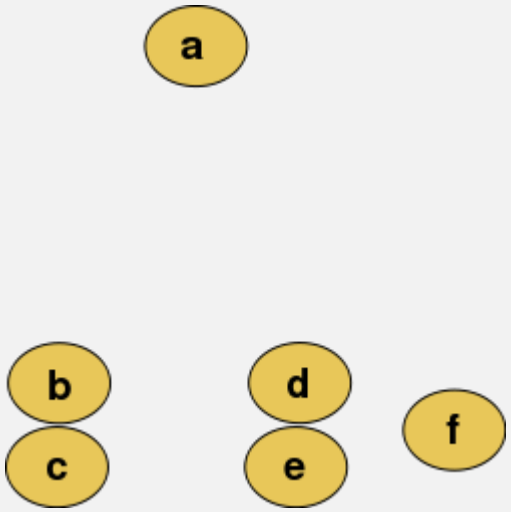
```
library(fpc)
pamk.result <- pamk(iris2)

pamk.result$nc

table(pamk.result$pamobject$clustering, iris$Species)

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

Clustering with pam()

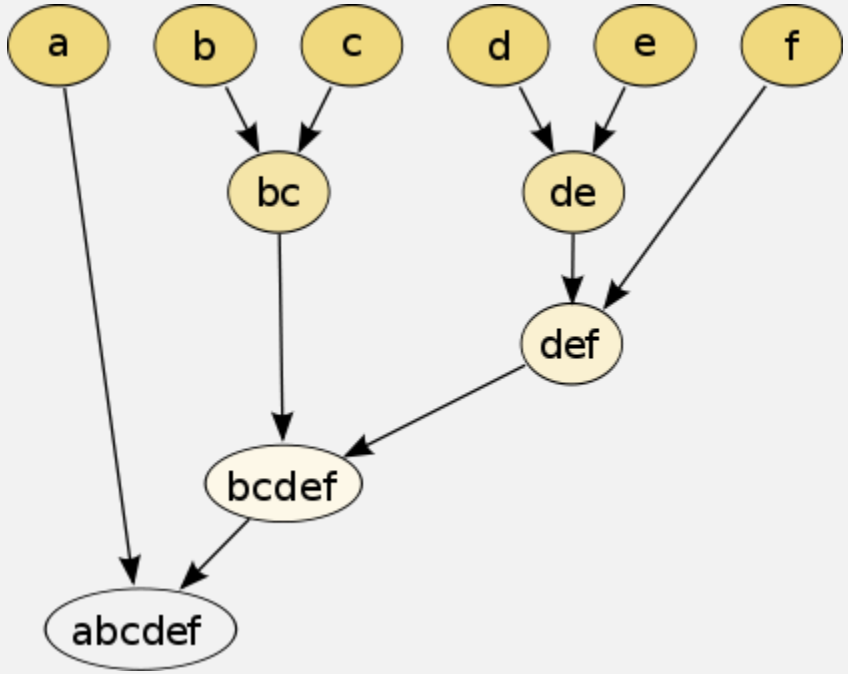
```
library(cluster)

pam.result <- pam(iris2, 3)
table(pam.result$clustering, iris$Species)

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

Contents

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'))

plot(hc, hang = -1, labels = iris$Species[idx])

rect.hclust(hc, k = 3)

(groups <- cutree(hc, k =3))
```

2. 현장사례

- K-means를 활용한 블로그 검색기법 연구
  - 블로그 숫자의 엄청난 증가와 함께, 많은 양의 중요한 정보가 블로그에 나타나게 되었다. 그러나 블로그 페이지들은 일반적인 웹 페이지들과 다루는 주제나 성향이 다르게 때문에 현재의 검색 엔진들이 적합한 검색 시스템을 제공하고 있지 못한 실정이다. 그러므로 많은 연구자들이 블로그들의 유용한 정보를 얻길 원하는 사람들을 위해 블로그만을 위한 검색체계를 연구하고 있다. 우리는 이 논문에서 KMeans를 사용하여 클러스터링 방법을 제안하고자 한다. 먼저 우리는 블로그가 무엇인지 살펴보고 현재 어떠한 방법들이 블로그 검색 엔진에 사용하는지 살펴볼 것이다. 둘째로 K-Means 알고리즘을 연구하여 블로그 제목에 적용하는 방법에 대하여 연구할 것이다. 마지막으로 우리의 알고리즘을 평가할 수 있는 프로토타입을 만들어 타당성을 판단하고 결론과 추후 작업사항에 대하여 제시하고자 한다.
- 클러스터링 기반 사례 기반 추론을 이용한 웹 개인화 추천 시스템
  - 최근, 추천시스템과 협업 필터링에 대한 연구가 학계와 업계에서 활발하게 이루어지고 있다. 하지만, 제품 아이템들은 다중값 속성을 가질 수 있음에도 불구하고, 기존의 연구들은 이러한 다중값 속성을 반영하지 못하고 있다. 이러한 한계를 극복하기 위하여, 본 연구에서는 추천시스템을 위한 새로운 방법론을 제시하고자 한다. 제안된 방법론은 제품 아이템에 대한 클러스터링 기법에 기반하여 다중값 속성을 활용하며, 정확한 추천을 위하여 협업 필터링을 적용한다. 즉, 사용자 간의 상관관계만이 아니라 아이템 간의 상관관계를 고려하기 위하여, 사용자 클러스터링에 기반한 사례기반추론과 아이템 속성 클러스터링에 기반한 사례기반추론 모두가 협업 필터링에 적용되는 것이다. 다중값 속성에 기반하여 아이템을 클러스터링 함으로써, 아이템의 특징이 명확하게 식별될 수 있다. MovieLens 데이터를 이용하여 실험을 하였으며, 제안된 방법론이 기존 방법론의 성능을 능가한다는 결과를 얻을 수 있었다.

3. 예제 또는 실습

- USArrests
- 미국 주 별 범죄율
- 범죄율을 기준으로 유사한 미국 주 확인

4. 학습진단 평가문제

- 다음 중 Clustering 알고리즘이 아닌 것은?

① K-Means　　② K-Medoids

③ K Nearest Neighbor　　④ Hierarchical Clustering

- ctree 함수가 있는 package는?

① ctree　　② iris

③ tree　　④ party

- SVM에서 데이터를 Linearly separable할 수 있게 변환시켜주는 함수는?

① Kernel Function　　② Random Function

③ Shadow Function　　④ Transform Function

참고문헌

- Jiawei Han, *데이터마이닝 (제2판) : 개념과 기법*, 사이플러스(2007), 1 ~ 717.
- 박창이, *R을 이용한 데이터마이닝*, 교우사(2011), 1 ~ 370.
- Brett Lantz, Machine Learning with R - Second Edition, PACKT(2015), 1 ~ 454.

과정요약

- 데이터 마이닝은 데이터로부터 숨겨진 지식을 발견하기 위한 일련의 과정이다.
- OSS R은 데이터 분석을 위한 언어이자 도구로 이를 사용하여 쉽게 데이터 마이닝을 수행할 수 있다.
- 학습 방법은 학습 결과에 대한 정답이 존재하는 지에 따라 교사 학습과 비교사 학습으로 나누어 진다.
- 교사 학습 방법은 분류, 회귀 분석이 있으며, 대표적인 알고리즘으로는 Neural Networks, Decision Tree, SVM, Linear Regression 등이 있다.
- 비교사 학습 방법은 군집화가 있으며, 대표적인 알고리즘으로는 K-Means, Hcluster 등이 있다.
- 소스코드는 https://goo.gl/5rnUe5 에서 다운로드 받을 수 있음

수강후 가이드

- 국가 공인 데이터 분석 전문가 – ADP : http://www.dbguide.net/da.db?cmd=snb_adp_1
- 국가 공인 데이터 분석 준전문가 – ADsP : http://www.dbguide.net/da.db?cmd=snb_adsp_1

판교 | 경기도 성남시 분당구 삼평동 대왕판교로 670길 유스페이스2 B동 8층 T. 070-5039-5805
가산 | 서울시 금천구 가산동 371-47 이노플렉스 1차 2층 T. 070-5039-5815
웹사이트 | http://edu.kosta.or.kr **팩스 |** 070-7614-3450