R을 활용한 데이터분석

호서대학교 빅데이터경영공학부 연규필 (kpyeon1@hoseo.edu)

분류모형

1. 로지스틱 회귀모형
2. 의사결정 나무모형
3. 신경망 모형
4. K-최근접 이웃 모형
5. 서포트 벡터 머신
6. 앙상블 모형
7. 예측모형의 평가

1. 로지스틱 회귀모형

■ 로지스틱 회귀분석(Logistic regression)

- 반응변수가 이항형 또는 다항형인 범주형 변수인 경우에 사용되는 예측모형

  - 이항형: 어떤 상품의 구입여부 (구입=1, 구입하지 않음=0)

  - 다항형: 고객의 신용등급(A=매우 좋음, B=좋음, C=좋지 않음, D=매우 좋지 않음)

- 반응변수가 갖는 각 범주의 확률을 추정해주며, 이를 통해 반응변수의 범주를 예측함

■ 모형

$$\log \frac{P(y=1|x_1,\cdots,x_p)}{1-p(y=1|x_1,\cdots,x_p)} = \alpha + \beta_1 x_1 + \cdots + \beta_p x_p$$

$$\Rightarrow \quad \hat{P}(y=1|x_1,\cdots,x_p) = \frac{\exp(a+b_1x_1+\cdots+b_px_p)}{1+\exp(a+b_1x_1+\cdots+b_px_p)}$$

- 통상적으로 사후확률 추정치가 0.5보다 크면 목표범주(y=1)로 분류함

1. 로지스틱 회귀모형

■ 오즈비(Odds ration)를 통한 회귀계수 해석 (Y: 0 또는 1, 목표범주는 1인 경우)

$$Odds\ Ratio = \frac{\exp\left[\alpha + \beta_1 x_1 + \cdots + \beta_i (x_i + 1) + \cdots \beta_p x_p\right]}{\exp\left[\alpha + \beta_1 x_1 + \cdots + \beta_i (x_i) + \cdots \beta_p x_p\right]} = \exp(\beta_i)$$

- 오즈(odd)= P(Y=1)/P(Y=0): Y가 0일 확률에 대한 Y가 1일 확률의 상대비
- 오즈비(odds ration)는 두 오즈의 상대비를 나타냄. 즉,

  odds_1 = P(Y=1|x)/P(Y=0|x), odds_2 = P(Y=1|x+1)/P(Y=0|x+1) 일 때

  오즈비 = odds_2/odds_1

- 회귀계수가 음수이면, 입력변수 x가 증가할 때 반응변수가 목표범주일 확률이 감소
- 회귀계수가 양수이면, 입력변수 x가 증가할 때 반응변수가 목표범주일 확률이 증가
- 회귀계수<0 ⇔ 오즈비 < 1
- 회귀계수>0 ⇔ 오즈비 > 1
- (예) 월수입 x(단위 100만원)를 입력변수로 하고 어떤 상품에 대한 구입여부(1=구입, 0=구입하지 않음) y를 반응변수라고 하자. 목표범주를 y=1(구입)로 하여 분석하는 경우에 b=3.73이라고 해보자. 이는 x가 1단위(백만원) 증가하면 구매하지 않을 확률에 대한 구매할 확률의 상대비가 exp(3.73)=42배 증가한다는 것을 의미한다.

1. 로지스틱 회귀모형

■ 오즈비(Odds ration)를 통한 회귀계수 해석 (Y: 0 또는 1, 목표범주는 1인 경우)

$$Odds\ Ratio = \frac{\exp\left[\alpha + \beta_1 x_1 + \cdots + \beta_i (x_i + 1) + \cdots \beta_p x_p\right]}{\exp\left[\alpha + \beta_1 x_1 + \cdots + \beta_i (x_i) + \cdots \beta_p x_p\right]} = \exp(\beta_i)$$

- 오즈(odd)= P(Y=1)/P(Y=0): Y가 0일 확률에 대한 Y가 1일 확률의 상대비
- 오즈비(odds ration)는 두 오즈의 상대비를 나타냄. 즉,

  odds_1 = P(Y=1|x)/P(Y=0|x), odds_2 = P(Y=1|x+1)/P(Y=0|x+1) 일 때

  오즈비 = odds_2/odds_1

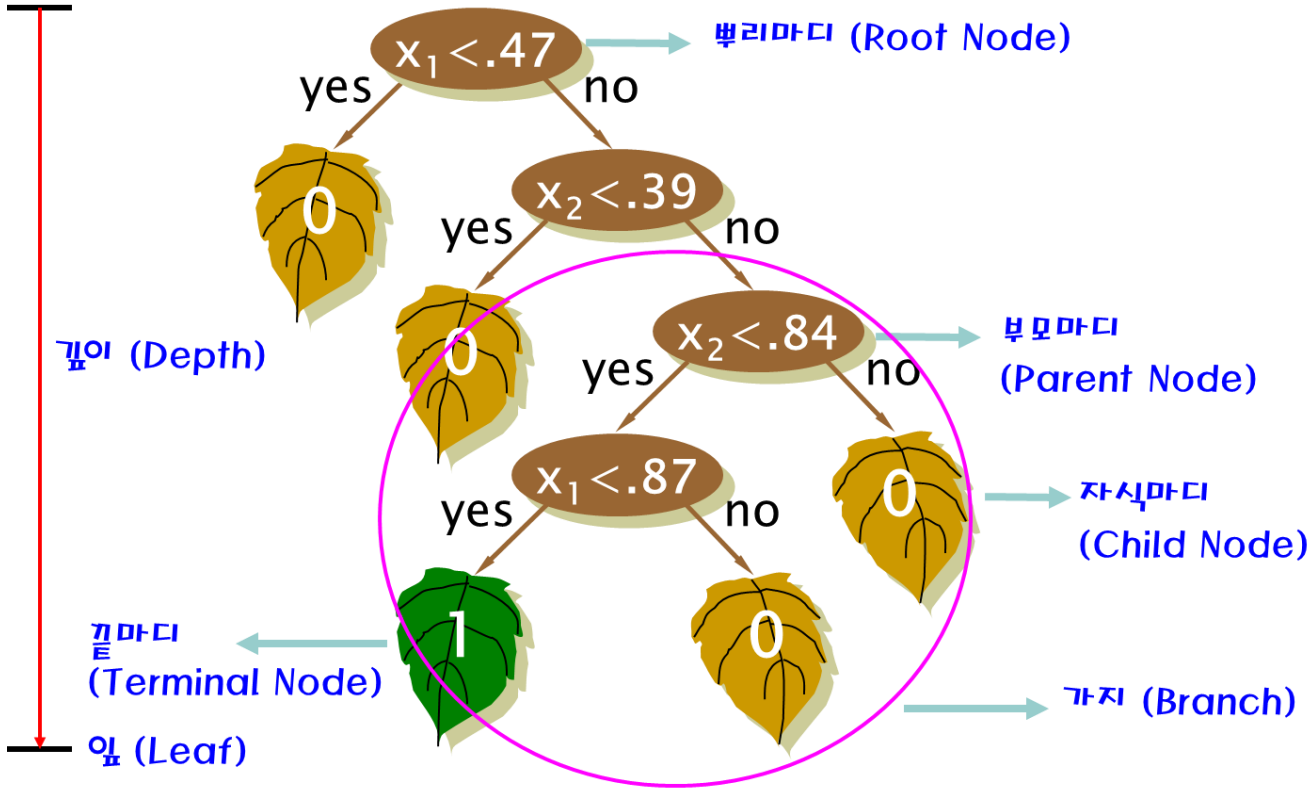
- 회귀계수가 음수이면, 입력변수 x가 증가할 때 반응변수가 목표범주일 확률이 감소
- 회귀계수가 양수이면, 입력변수 x가 증가할 때 반응변수가 목표범주일 확률이 증가
- 회귀계수<0 ⇔ 오즈비 < 1
- 회귀계수>0 ⇔ 오즈비 > 1
- (예) 월수입 x(단위 100만원)를 입력변수로 하고 어떤 상품에 대한 구입여부(1=구입, 0=구입하지 않음) y를 반응변수라고 하자. 목표범주를 y=1(구입)로 하여 분석하는 경우에 b=3.73이라고 해보자. 이는 x가 1 단위(백만원) 증가하면 구매하지 않을 확률에 대한 구매할 확률의 상대비가 exp(3.73)=42배 증가한다는 것을 의미한다.

2. 의사결정 나무모형

■ 의사결정나무모형(Decision Tree)

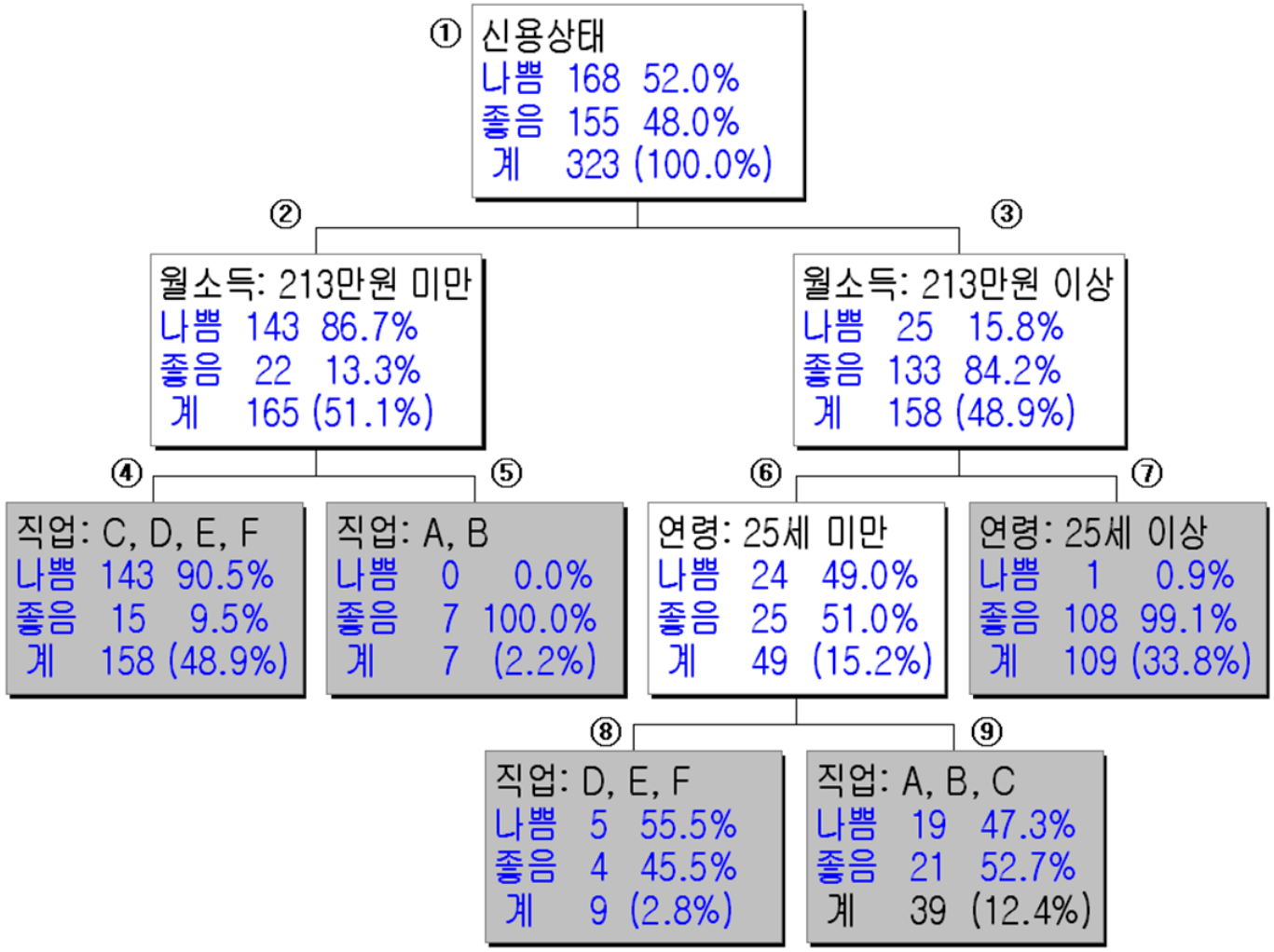
- 반응변수가 연속형일 때, 회귀나무(regression tree)
- 반응변수가 범주형일 때, 분류나무(classification tree)

■ 의사결정나무의 구성요소



2. 의사결정 나무모형

■ 사례: 신용평가



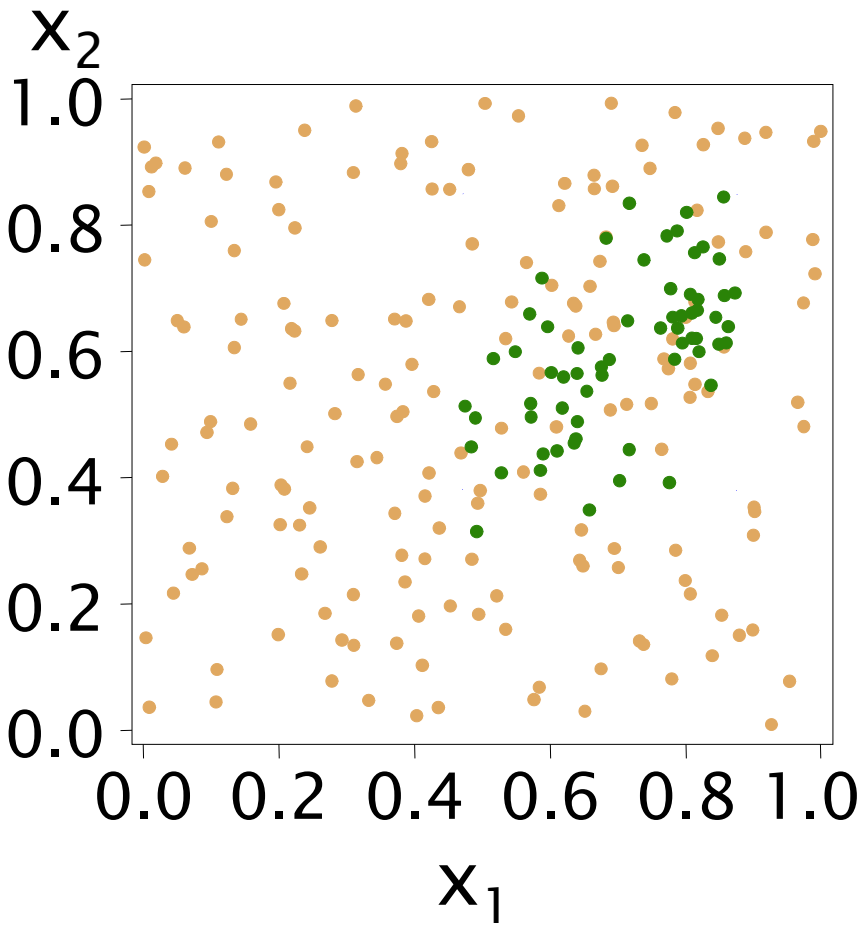
2. 의사결정 나무모형

■ 의사결정나무 형성과정

240
26%

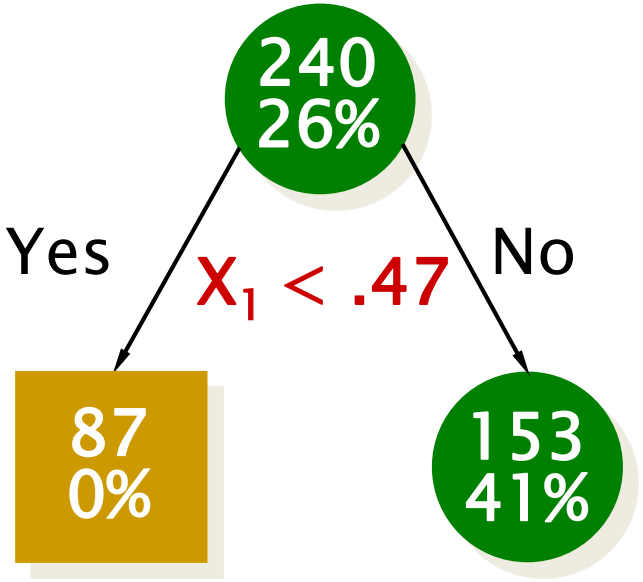
478 개의 분리조합이 존재

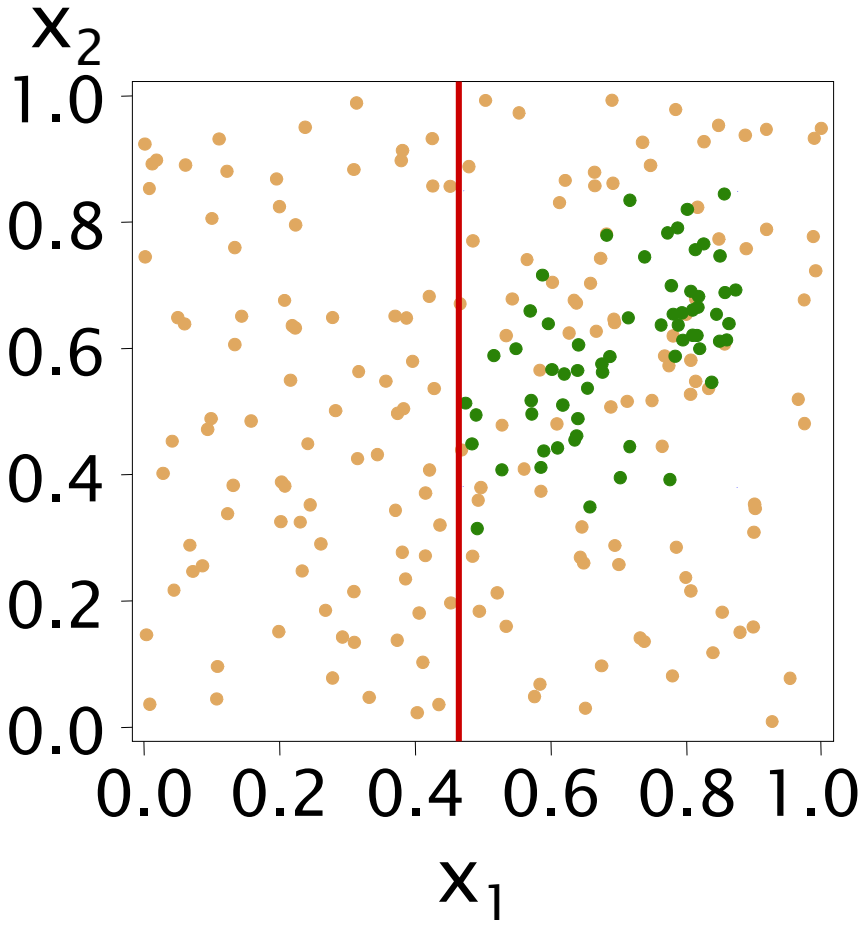
- x_1에 대해서 239개
  (x_1<.25, x_1<.26, etc.)
- x_2에 대해서 239개
  (x_2<.43, x_2<.86, etc.)



2. 의사결정 나무모형

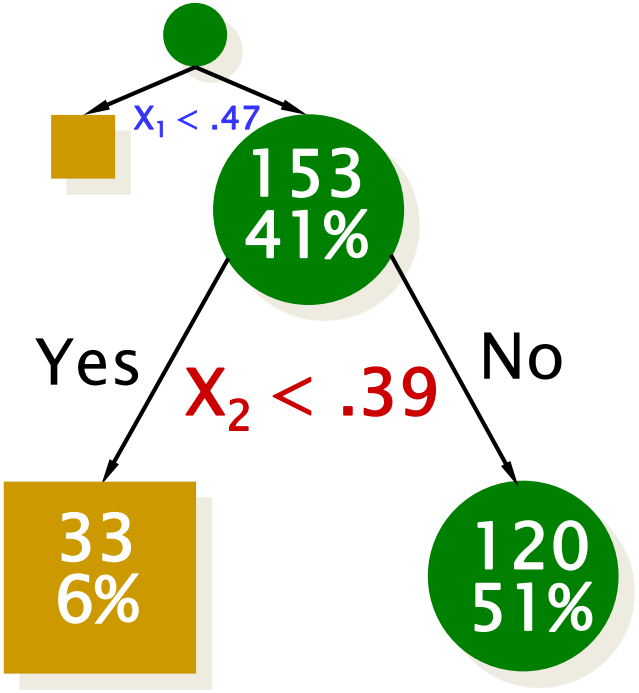
■ 의사결정나무 형성과정

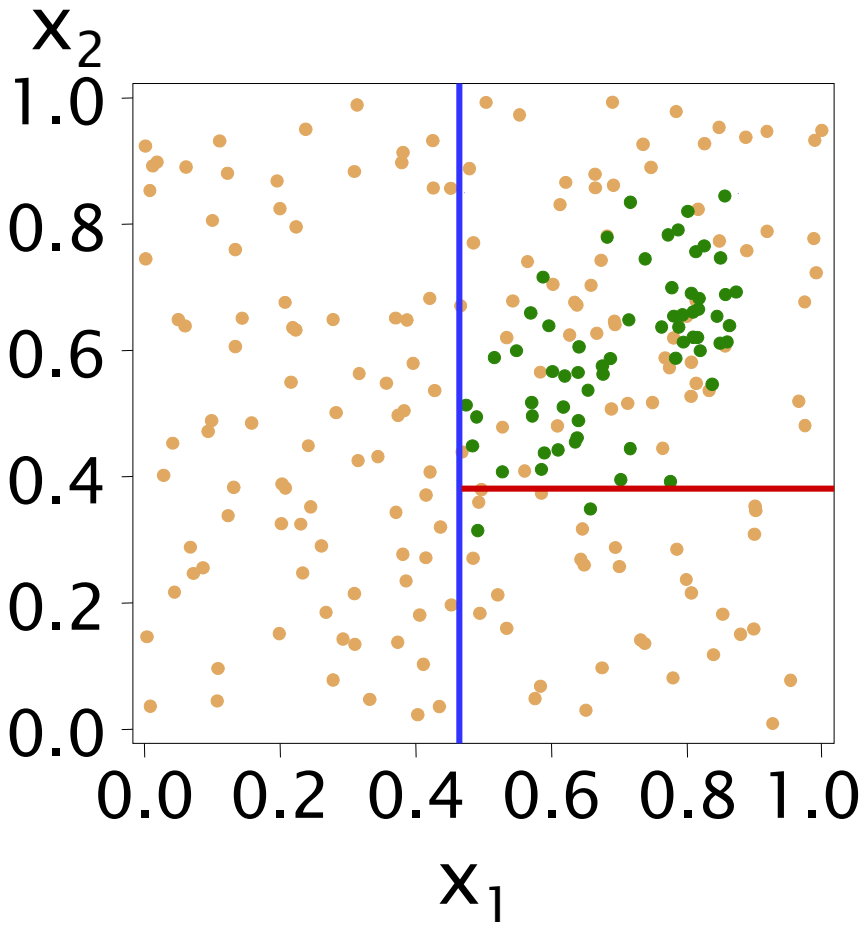




2. 의사결정 나무모형

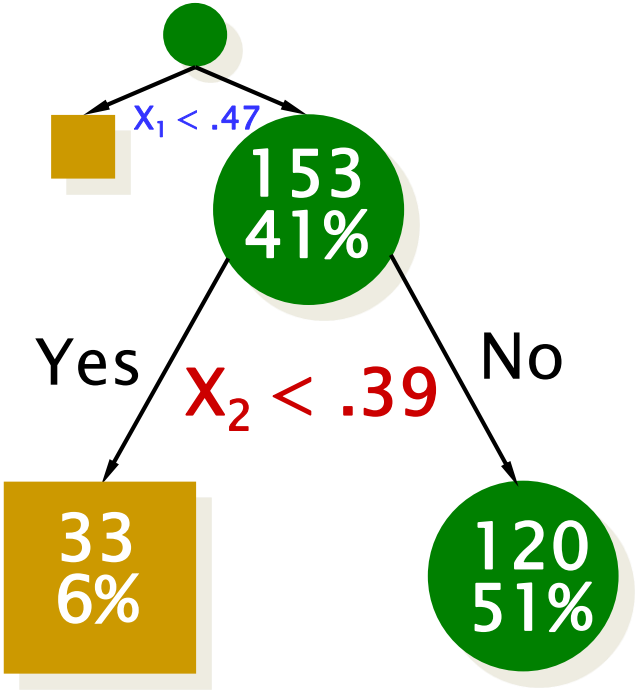
■ 의사결정나무 형성과정

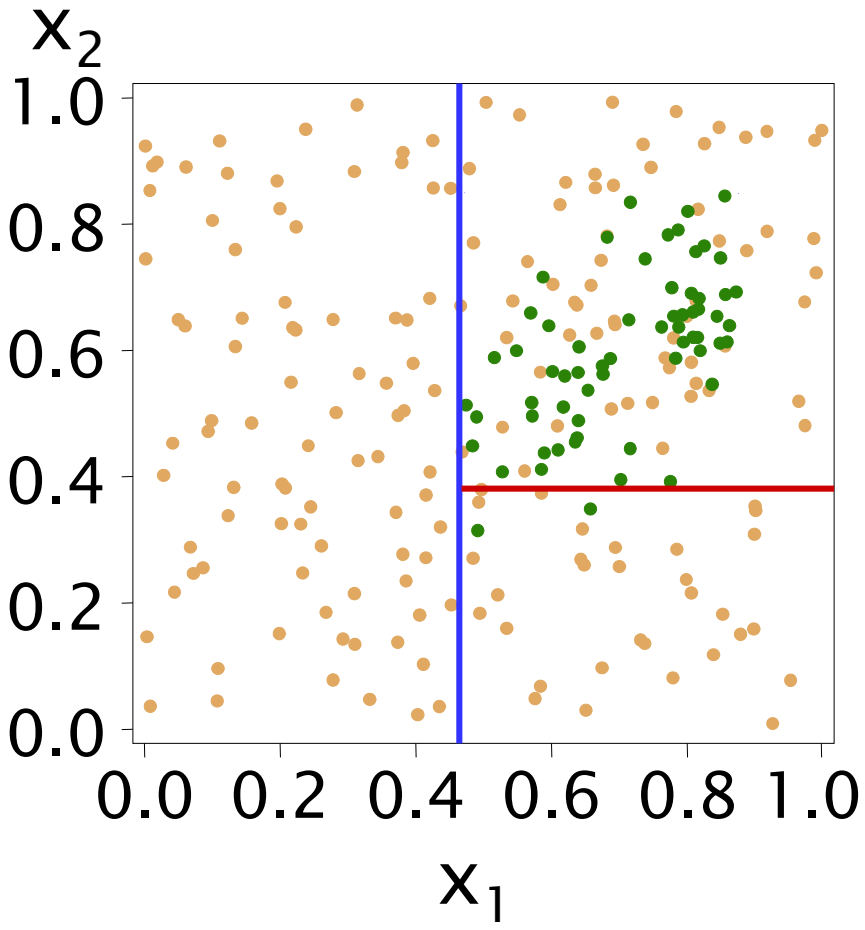




2. 의사결정 나무모형

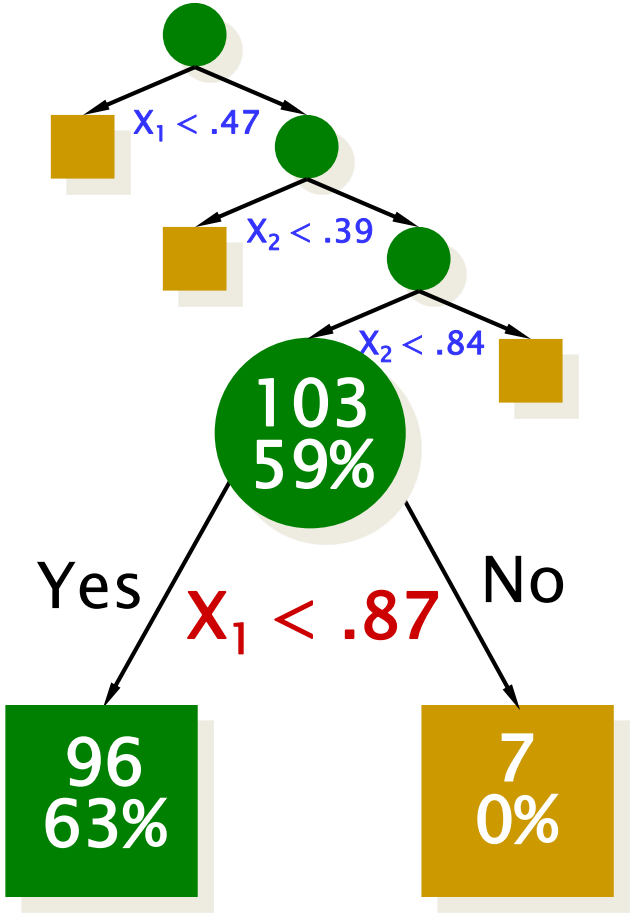
■ 의사결정나무 형성과정

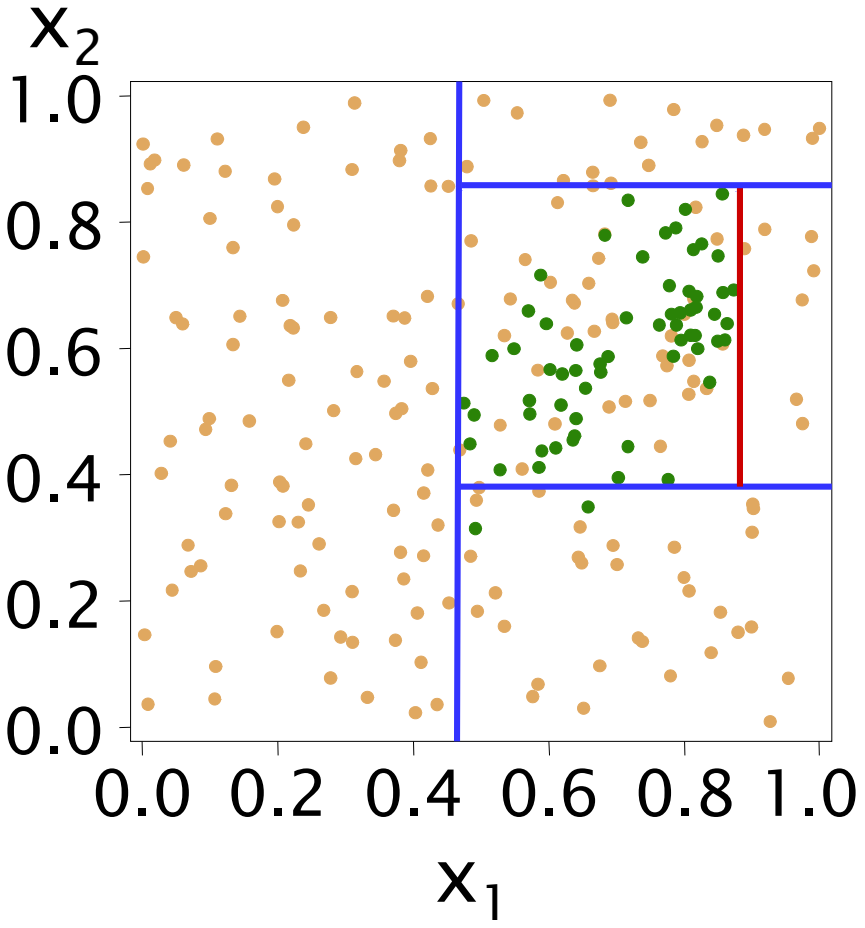




2. 의사결정 나무모형

■ 의사결정나무 형성과정





2. 의사결정 나무모형

■ 의사결정나무(분류나무)의 분리기준

- 카이제곱 통계량의 p-값 : CHAID, Kass(1980)

- 지니 지수 (Gini Index) : CART, BFOS(1984)

$$\sum\sum P(i)P(j) = \sum P(i)(1-P(j)) = 1 - \sum P(j)^2$$

$$= 1 - \sum (\frac{n_j}{n_0})^2$$

- 엔트로피 지수 (Entropy index) : C4.5, Quinlan(1993)

$$-\sum P(i)\log_2 P(i)$$

- 카이제곱 통계량이 지니 지수나 엔트로피 지수에 비해서 보다 단순한 형태의 나무구조를 가지게 하는 경향이 있음.

2. 의사결정 나무모형

■ 의사결정나무모형의 특징

- 해석의 용이성

- 나무구조의 모형이기에 사용자가 쉽게 이해할 수 있다.
- 새로운 개체에 대한 분류 또는 예측을 위해서 뿌리마디로부터 끝마디까지를 단순히 따라가면 되기에, 새로운 자료을 모형에 적합하기가 매우 쉽다.
- 나무구조로부터 어떤 입력변수가 목표변수를 설명하기 위해서 더 중요한지를 쉽게 파악할 수 있다.

- 상호작용 효과의 해석

- 두 개 이상의 변수가 결합하여 목표변수에 어떻게 영향을 주는지를 쉽게 알 수 있다.
- 의사결정나무는 유용한 입력변수나 상호작용(interaction)의 효과 또는 비선형성 (nonlinearity)을 자동적으로 찾아내는 알고리즘이라고 할 수 있다.

- 비모수적 모형

- 의사결정나무는 선형성(linearity)이나 정규성(normality) 또는 등분산성(equal variance) 등의 가정을 필요로 하지 않는 비모수적인(nonparametric) 방법이다.
- 의사결정나무에서는 순서형 또는 연속형 변수는 단지 순위(rank)만 분석에 영향을 주기 때문에 이상치(outlier)에 민감하지 않다는 장점을 가지고 있다.

2. 의사결정 나무모형

■ 의사결정나무모형의 특징

- 비연속성

- 의사결정나무에서는 연속형 변수를 비연속적인 값으로 취급하기 때문에 분리의 경계점 근방에서는 예측오류가 클 가능성이 있다.
- 최근에는 이러한 단점의 극복을 위해, 앞서 논의한 장점을 해치지 않고 모수적 모형이나 신경망 등을 의사결정나무와 결합하는 방법들이 연구되고 있다

- 선형성 또는 주효과의 결여

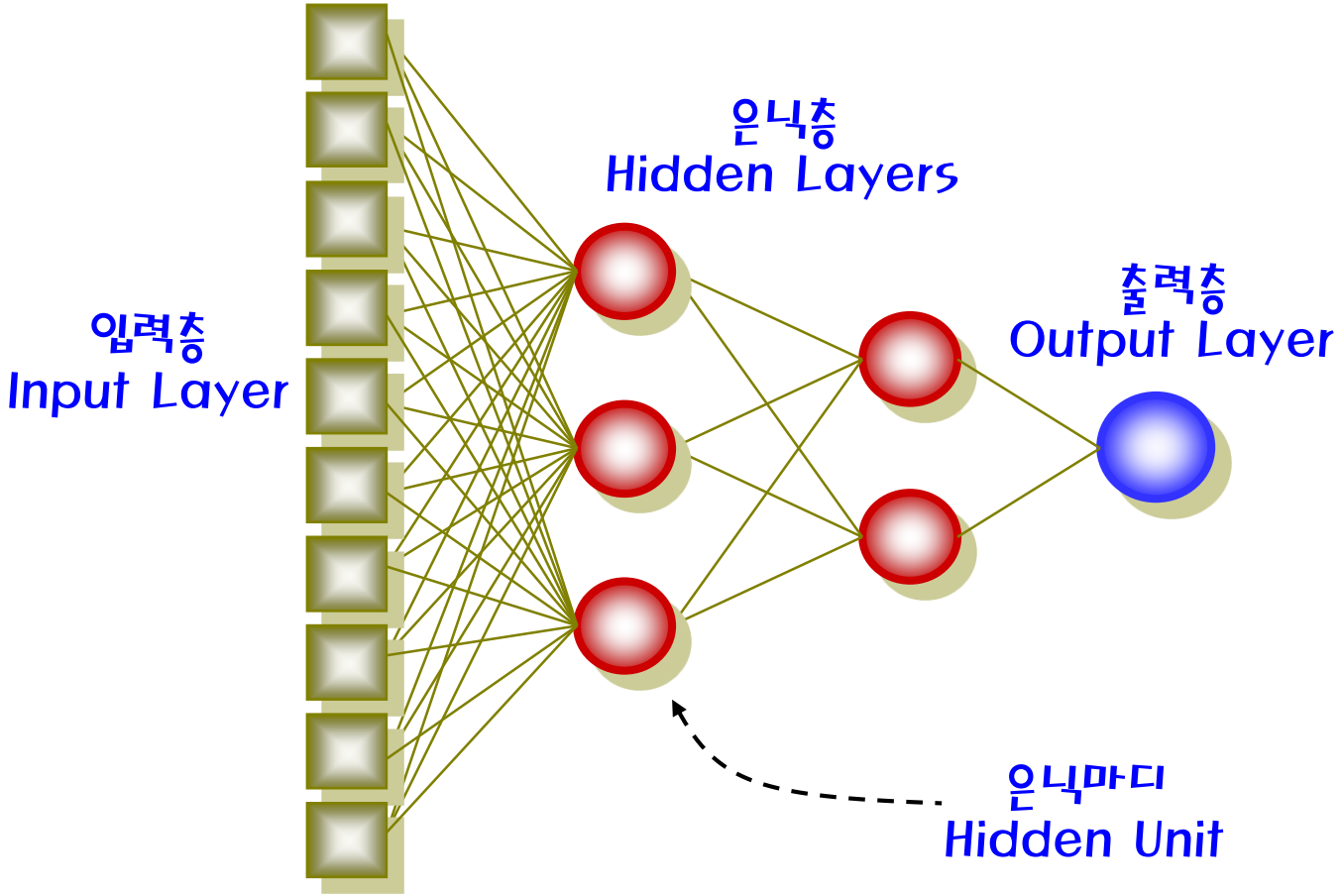
- 회귀모형에서는 회귀계수나 오즈비(odds ratio)를 이용하여 결과에 대한 유용한 해석을 얻을 수 있다. 즉, 선형모형(linear model)은 각 변수 고유의 영향력을 해석할 수 있다. 그러나, 의사결정나무는 선형(linear) 또는 주효과(main effect) 모형에서와 같은 결과를 얻을 수 없다는 한계점이 있다.

- 비안정성

- 분석용 자료(training data)에만 의존하기에 새로운 자료의 예측에서는 불안정(unstable)할 가능성이 높다. 특히 분석용 자료의 크기가 너무 작은 경우와 너무 많은 가지를 가지는 의사결정나무를 얻는 경우에 빈번히 발생한다.
- 따라서 검증용 자료(test data)에 의한 평가나 가지치기(Pruning)에 의해서 안정성 있는 의사결정나무를 얻는 것이 바람직하다.

3. 신경망 모형

■ MLP(multi-layer perceptron) 구조



■ MLP(multi-layer perceptron) 구조

- ### 입력층(Input Layer)

  각 입력변수에 대응되는 마디들로 구성되어 있다. 명목형(nominal) 변수에 대해서는 각 수준에 대응하는 입력마디를 가지게 되는데, 이는 통계적 선형모형에서 가변수(dummy variable)를 사용하는 것과 같다.

- ### 은닉층(Hidden Layer)

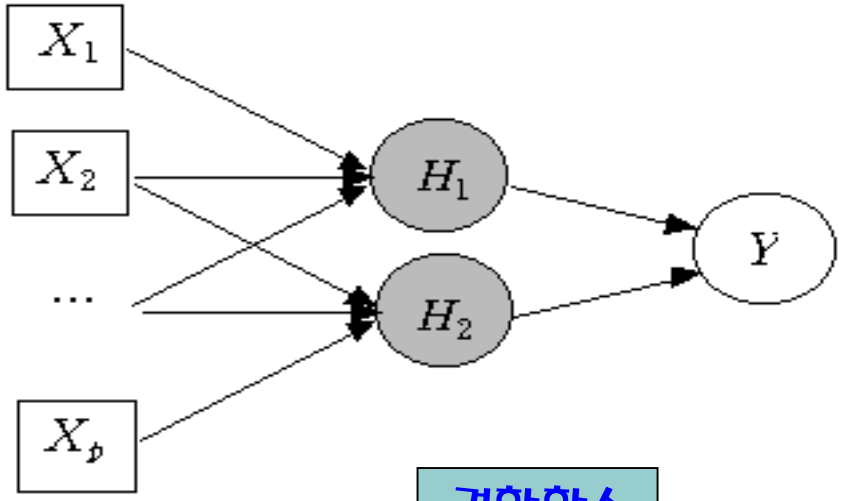
  여러 개의 은닉마디로 구성되어 있다. 입력층으로부터 전달되는 변수값들의 선형결합(linear combination)을 비선형함수(nonlinear function)로 처리하여 출력층 또는 다른 은닉층에 전달한다.

- ### 출력층(Output Layer)

  목표변수(target)에 대응하는 마디들을 갖는다. 여러 개의 목표변수 또는 세 개 이상의 수준을 가지는 명목형 목표변수가 있을 경우에는 여러 개의 출력마디들이 존재한다.

3. 신경망 모형

■ 결합함수와 활성함수



활성함수

결합함수

$$H_1 = f_1(b_1 + w_{11}X_1 + w_{21}X_2 + \cdots + w_{p1}X_p)$$

$$H_2 = f_2(b_2 + w_{12}X_1 + w_{22}X_2 + \cdots + w_{p2}X_p)$$

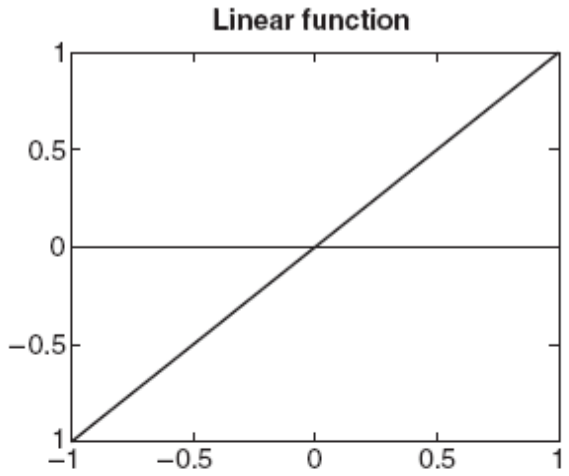
$$Y = g(b_0 + w_{10}H_1 + w_{20}H_2)$$

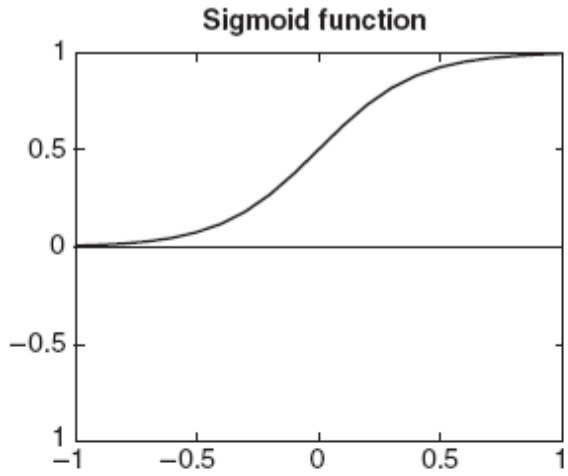
절편(bias)

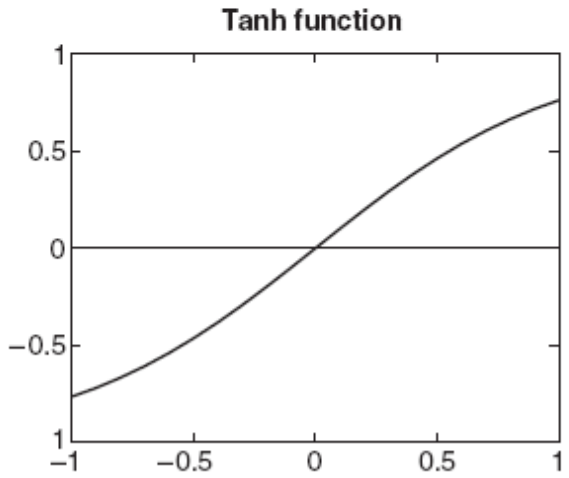
연결강도

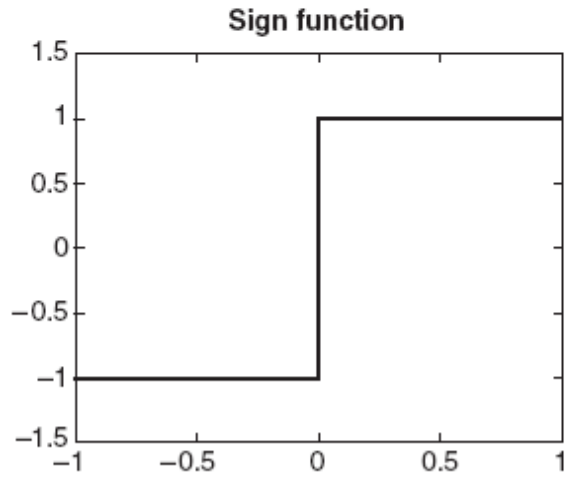
3. 신경망 모형

■ 활성함수의 예



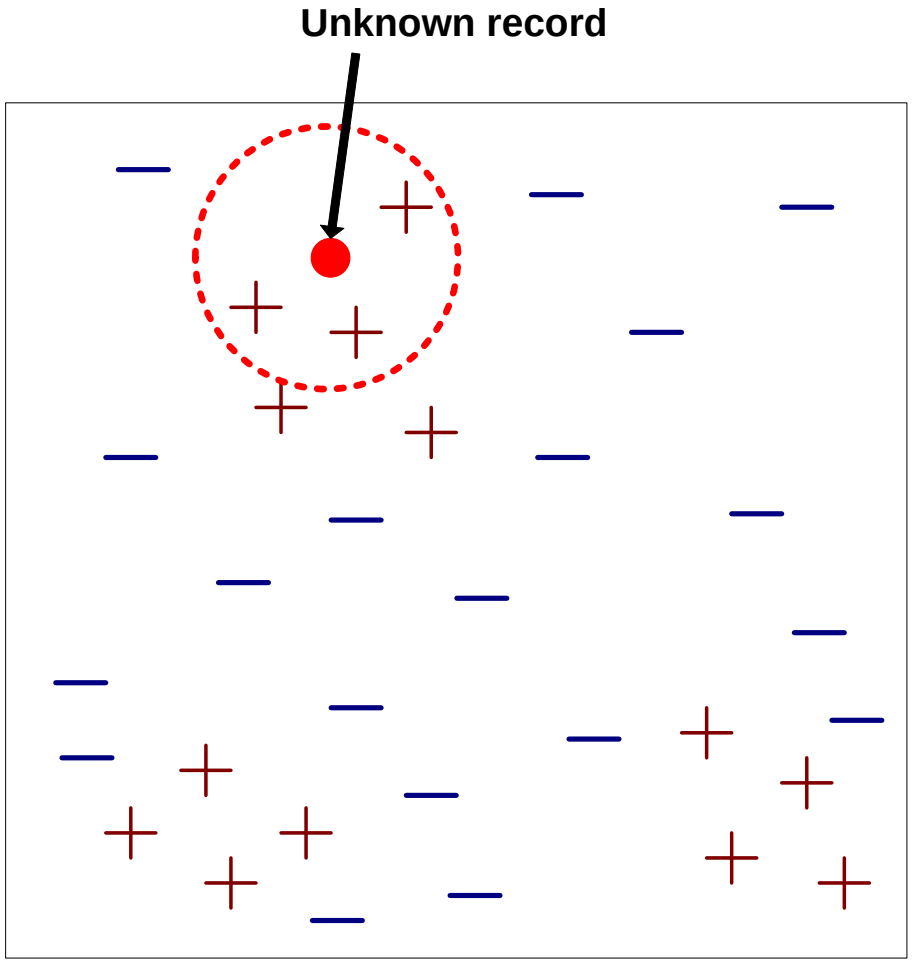






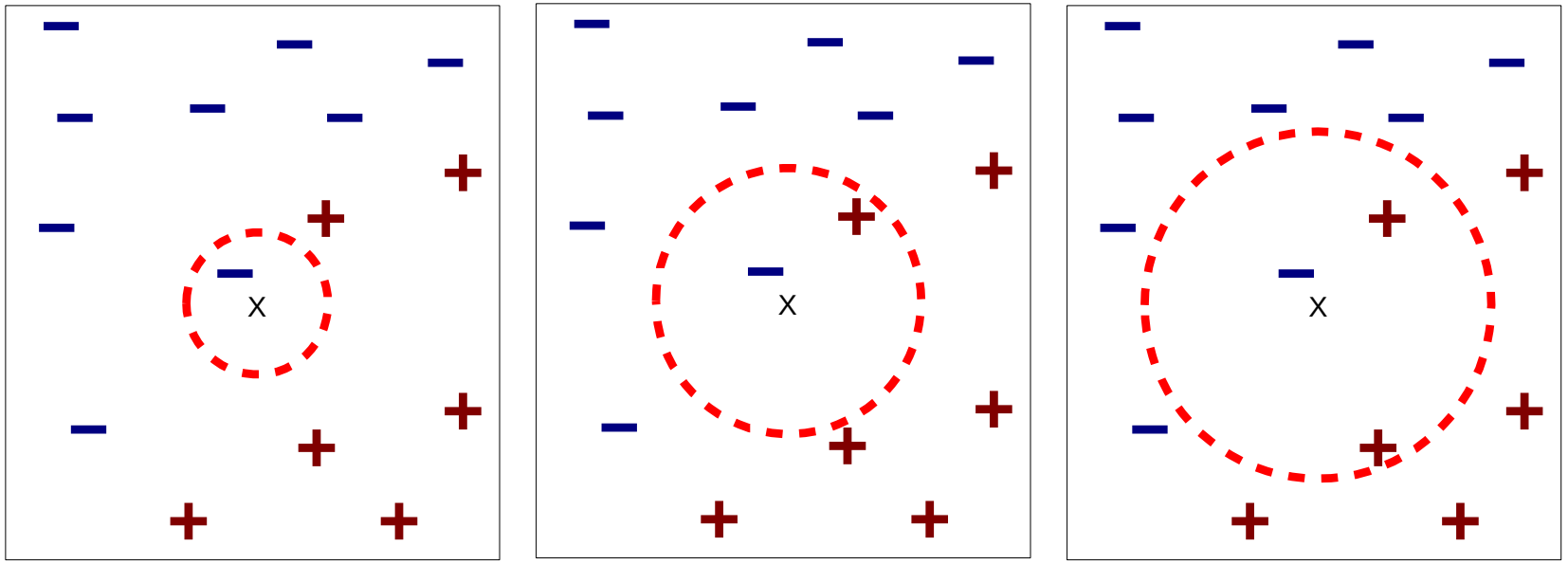
4. k-NN 모형

■ K-Nearest Neighbor 모형



4. k-NN 모형

■ K-Nearest Neighbor 모형



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

¶ 다수결을 통해 class를 결정하거나 다수결에 거리를 반영 (weight factor, $w \propto 1/d^2$)

■ K-NN에서의 고려사항

- Scaling

경우에 따라서 (예, 단위가 다른 경우) 변수들을 표준화한 후 거리를 계산한다.

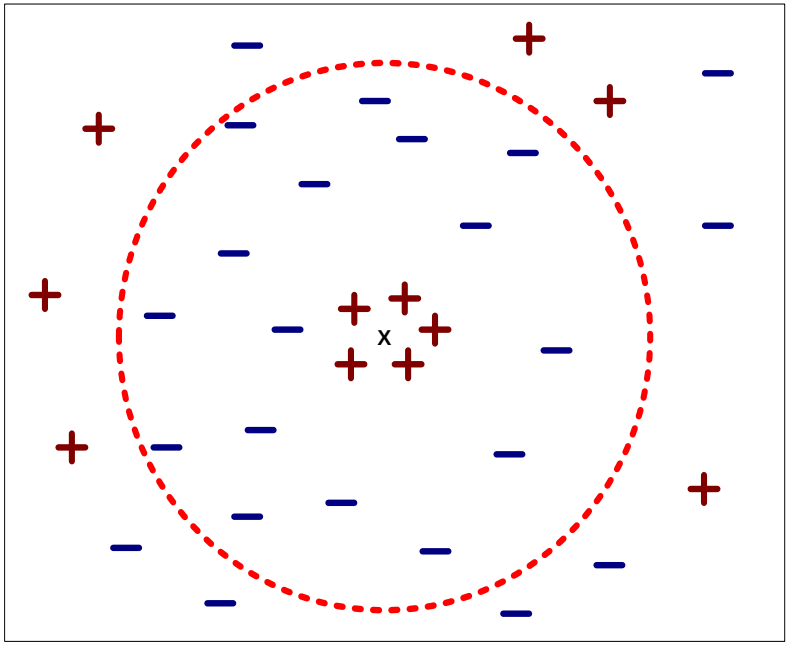
그렇지 않은 경우 변동폭이 큰 변수에 의해 무조건 결정이 되버린다.

  - Example:
    - (성인) 사람의 키: 1.4m ~ 2.3m
    - 사람의 몸무게: 40kg ~ 300kg
    - 사람의 수입: 0원 ~ 30,000,000,000원

4. k-NN 모형

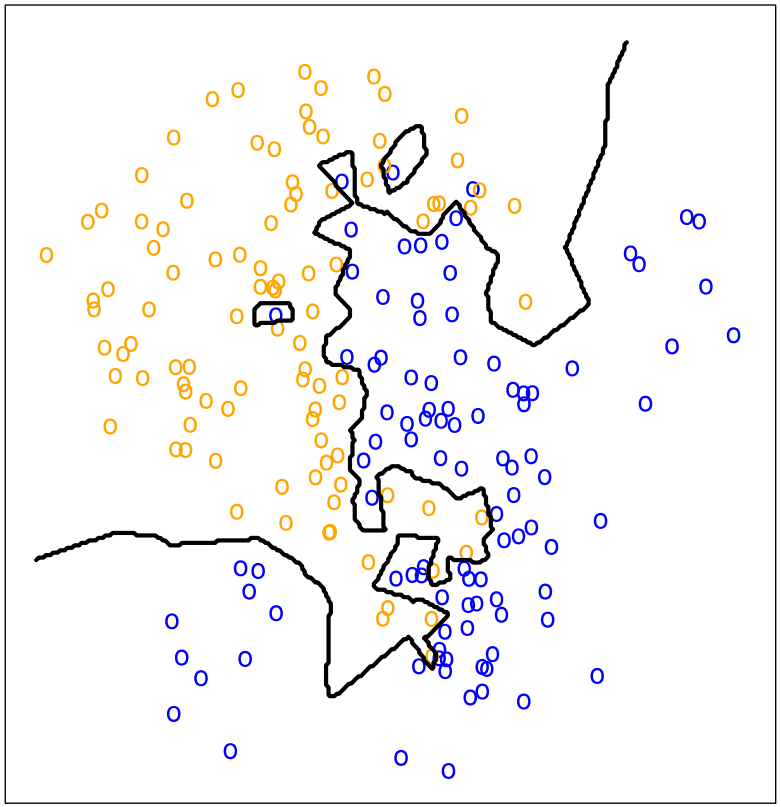
■ K-NN에서의 고려사항

- Choosing the value of k:
  - 너무 작은 k: noise에 취약
  - 너무 큰 k: 멀리 떨어진 곳에 위치한 점들까지 고려

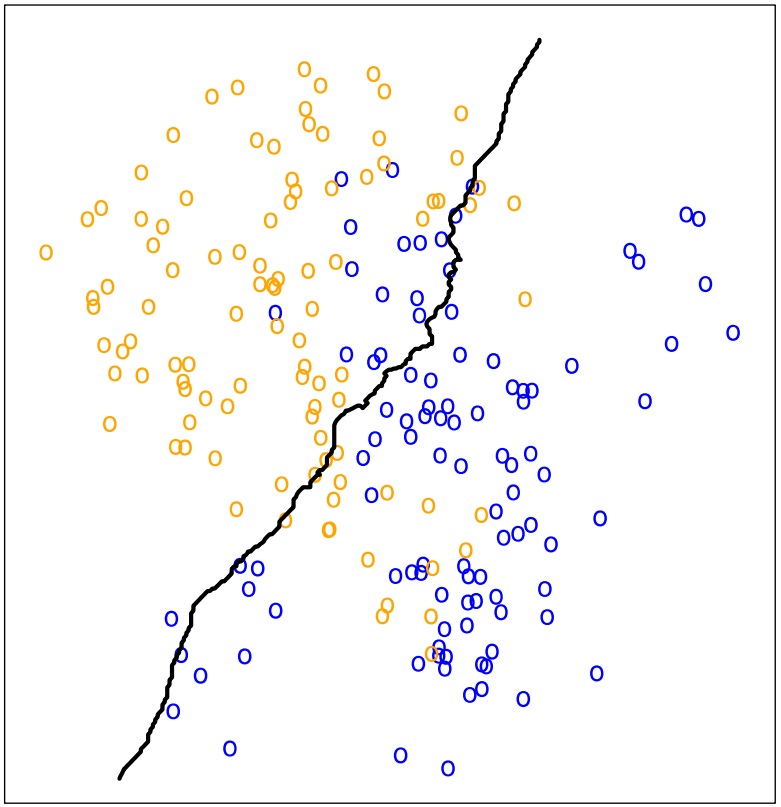


4. k-NN 모형

KNN: K=1

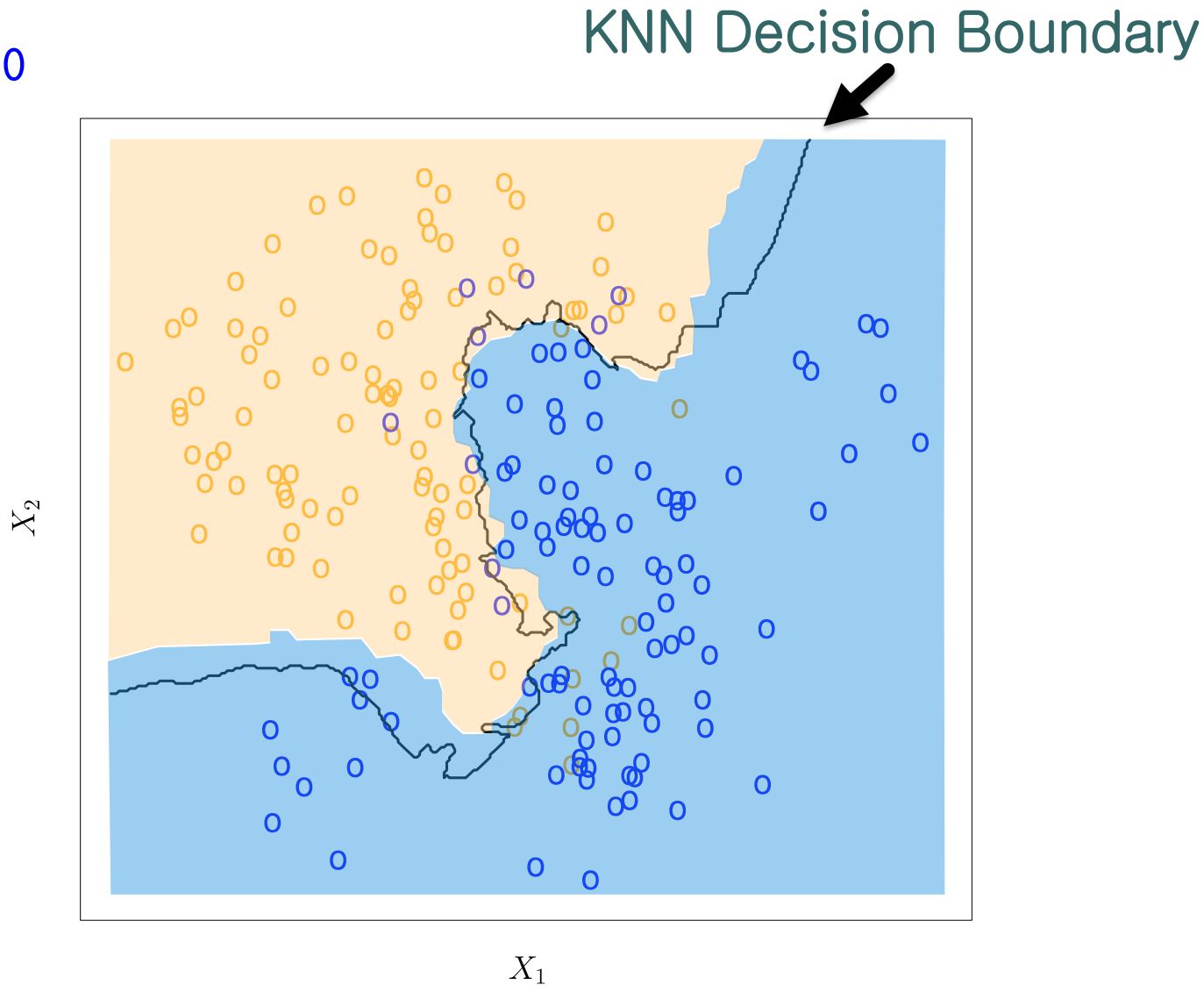


KNN: K=100



4. k-NN 모형

K=10



4. k-NN 모형

■ K-NN 모형의 특징

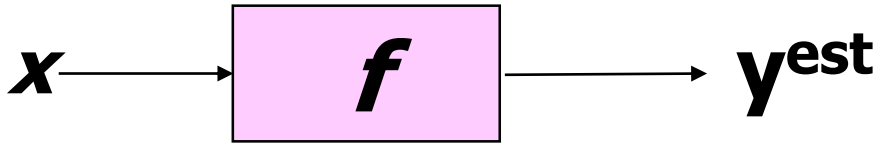
- 매우 자유로운 형태의 decision boundary를 형성
  - Decision tree는 직선의 형태
  - K의 값이 커질수록 이러한 성질은 약해진다.

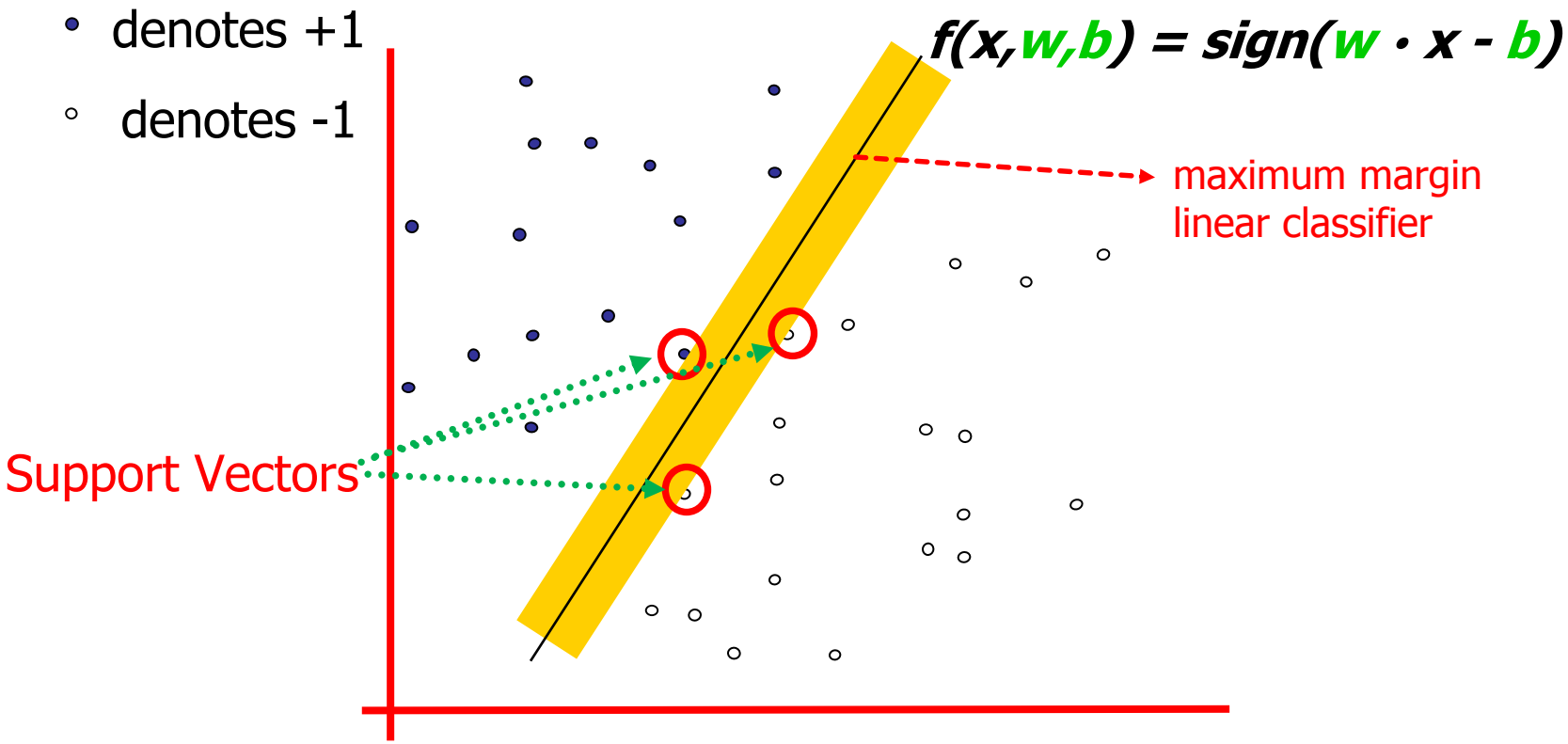
- Lazy learner
  - 모형을 구체적으로 설계할 필요가 없다. Test 자료가 입력된 후에야 분류를 한다.
  - 하나의 Test 자료를 분류할 때마다 모든 training data와의 거리를 측정해야 한다(expensive).

- (참고) Eager learner
  - decision tree algorithm
  - Training Data를 토대로 우선 모형을 만들어야 하므로 처음에 많은 자원을 필요로 한다. 그러나, 일단 모형이 만들어지면 test 자료를 분류하는 것은 매우 빠르다.

5. 서포트벡터머신

■ SVM

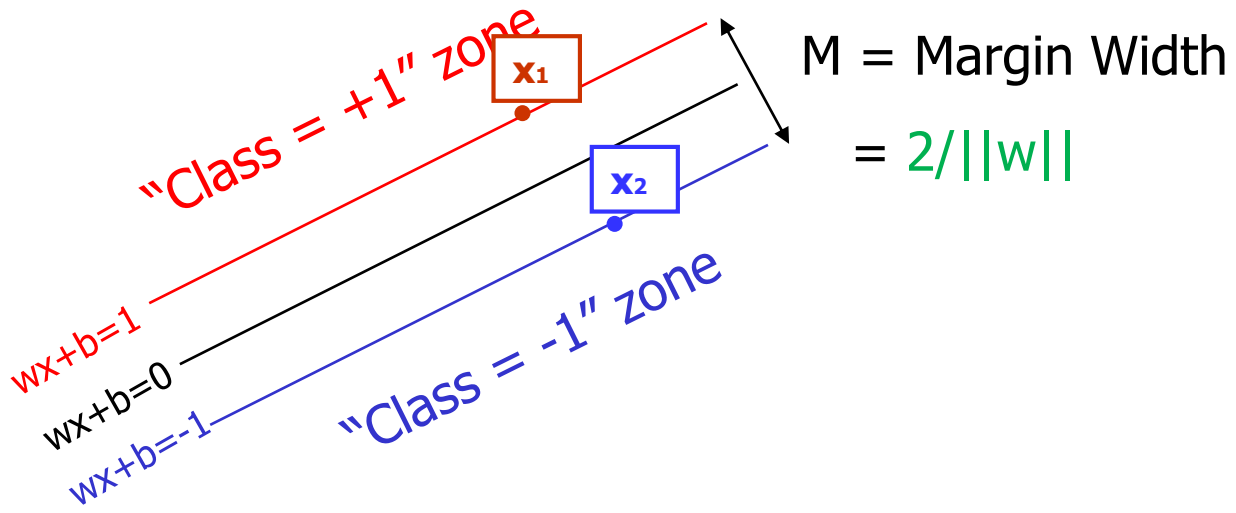




Source: Andrew Moor's Tutorial

5. 서포트벡터머신

■ 마진(margin) 최대화



- Plus-plane = $\{ x : w \cdot x + b = +1 \}$
- Minus-plane = $\{ x : w \cdot x + b = -1 \}$
- Separating plane = $\{ x : w \cdot x + b = 0 \}$

마진(margin)을 최대화 하는 w와 b를 찾자

: Max margin ➔ Max 2/||w|| ➔ Min ||w|| ➔ Min $||w||^2$

5. 서포트벡터머신

■ 최적화 식

$$\text{minimize:} \quad Q(\mathbf{w}) = \frac{1}{2}||\mathbf{w}||^2$$
$$\text{subject to:} \quad y_i(\mathbf{w}\cdot\mathbf{x}_i - b) \geq 1, \forall(\mathbf{x}_i, y_i) \in D$$

■ 최적해

$$\mathbf{w}^* = \sum_i \alpha_i^* y_i \mathbf{x}_i \qquad b^* = 1 - \mathbf{w}^* \cdot \mathbf{x}_i$$

서포트벡터가 아닌 경우 알파 값이 0이됨. 즉, SVM 해는 서포트벡터에만 의존함

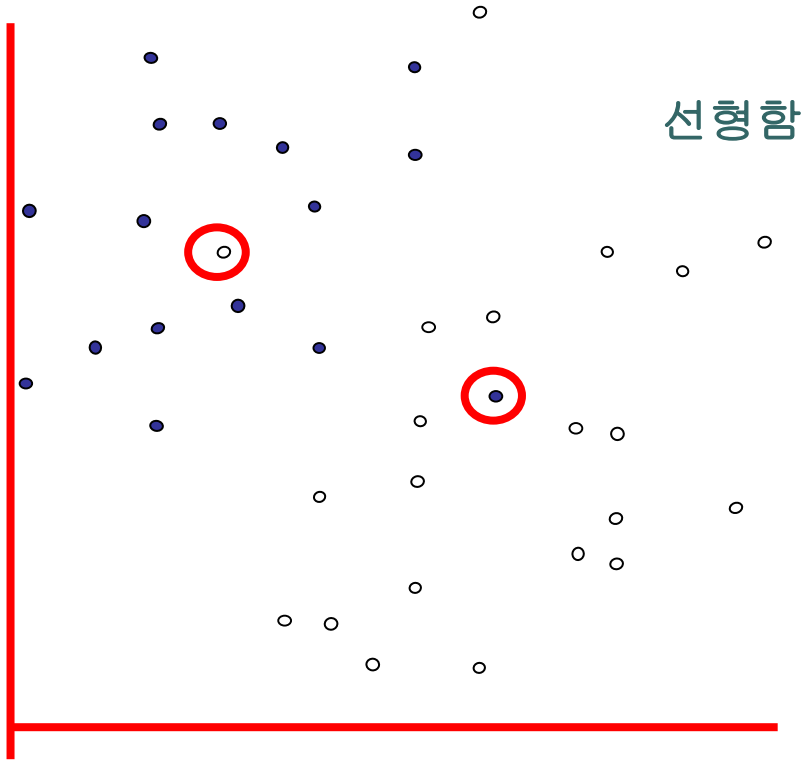
■ 선형판별함수

$$F(\mathbf{x}) = \sum_i \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x} - b$$

SVM 판별함수는 feature들의 내적에만 의존함 ➔ Kernel trick에 의해 비선형 SVM 구현

5. 서포트벡터머신

■ Nonseparable case

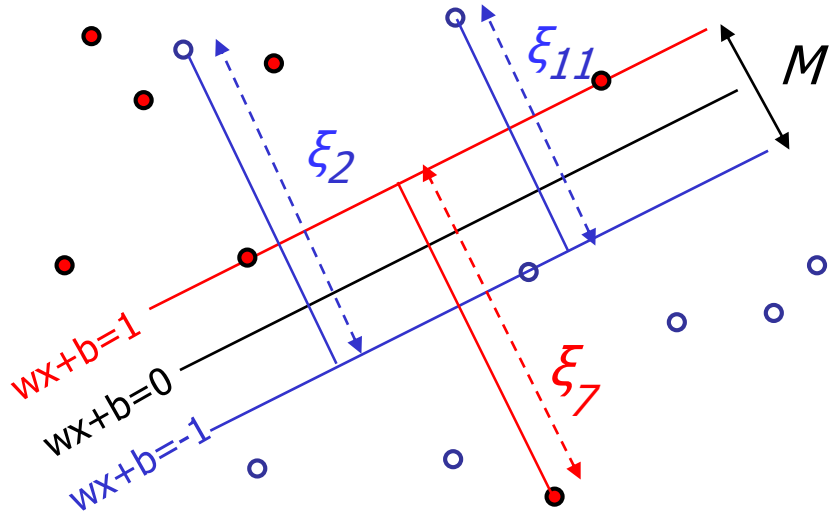


선형함수로는 +1과 −1을 분류할 수 없다.

2

5. 서포트벡터머신

■ Nonseperable case



Minimize $\frac{1}{2}||\mathbf{w}||^2 + C\sum_i \xi_i$

$\langle w, x_i \rangle + b \geq 1-\xi_i$ if $y_i = 1$

$\langle w, x_i \rangle + b \leq -1+\xi_i$ if $y_i = -1$

$\xi_i \geq 0$ for all i

5. 서포트벡터머신

■ SVM

Minimize $$\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\xi_i$$

s.t $$y_i \cdot ((\mathbf{w}\cdot\mathbf{x}_i) + b) \geq 1 - \xi_i, \quad i = 1, \ldots, m.$$

$$\xi_i \geq 0, \quad i = 1, \ldots, m$$

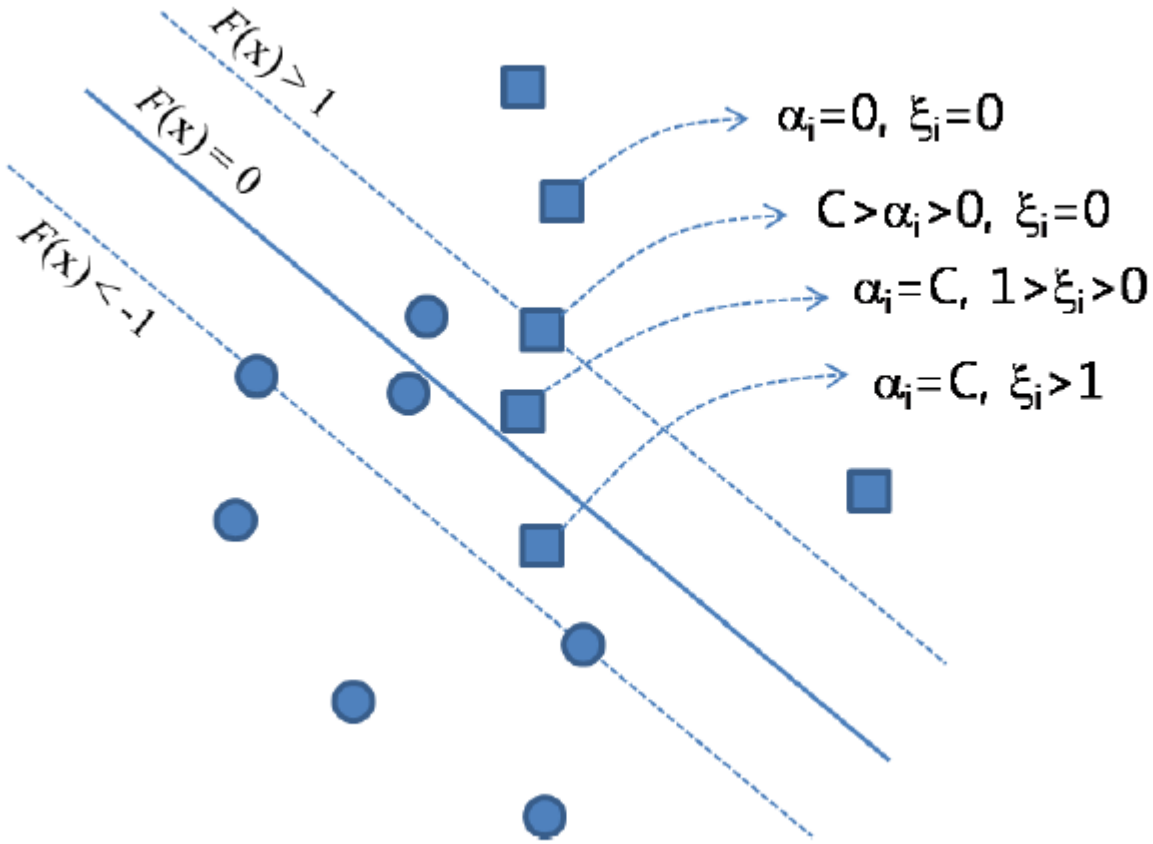
- 최적해 $$\mathbf{w}^* = \sum_{i=1}^{m_s}\alpha_i^* y_i \mathbf{x}_i$$ m_s : # of support vectors ($\alpha_i > 0$)

- 선형판별함수

$$F(\mathbf{x}) = \sum_i \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x} + b$$

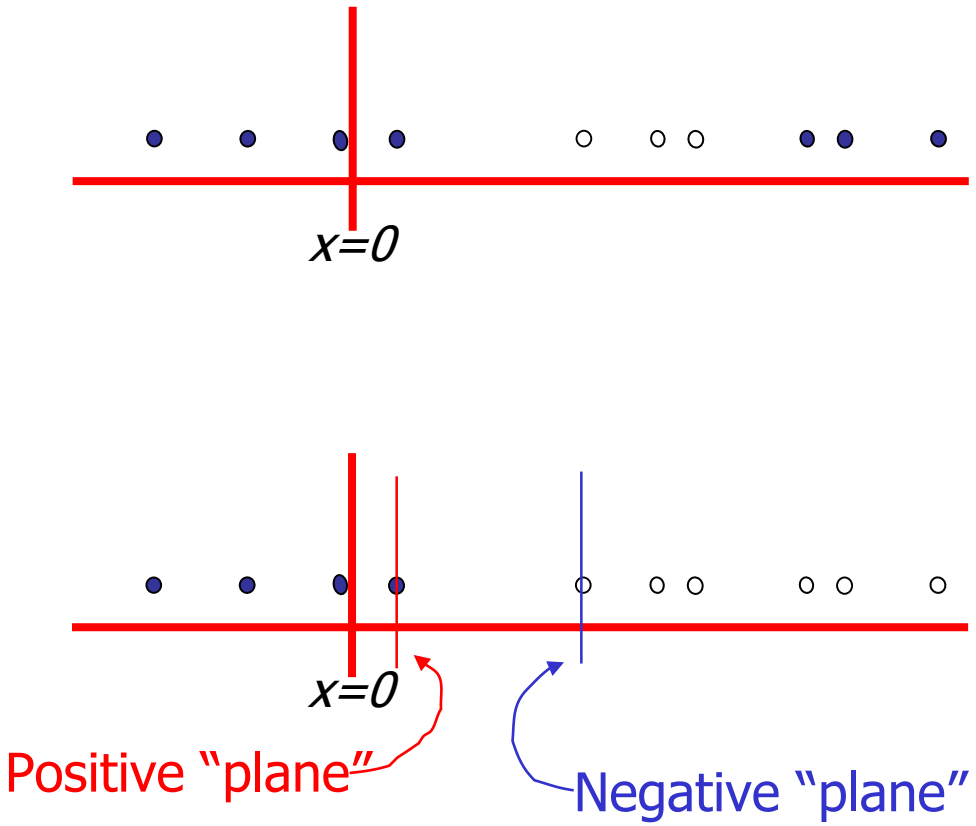
5. 서포트벡터머신

■ Relationship among α_i, ξ_i, and C



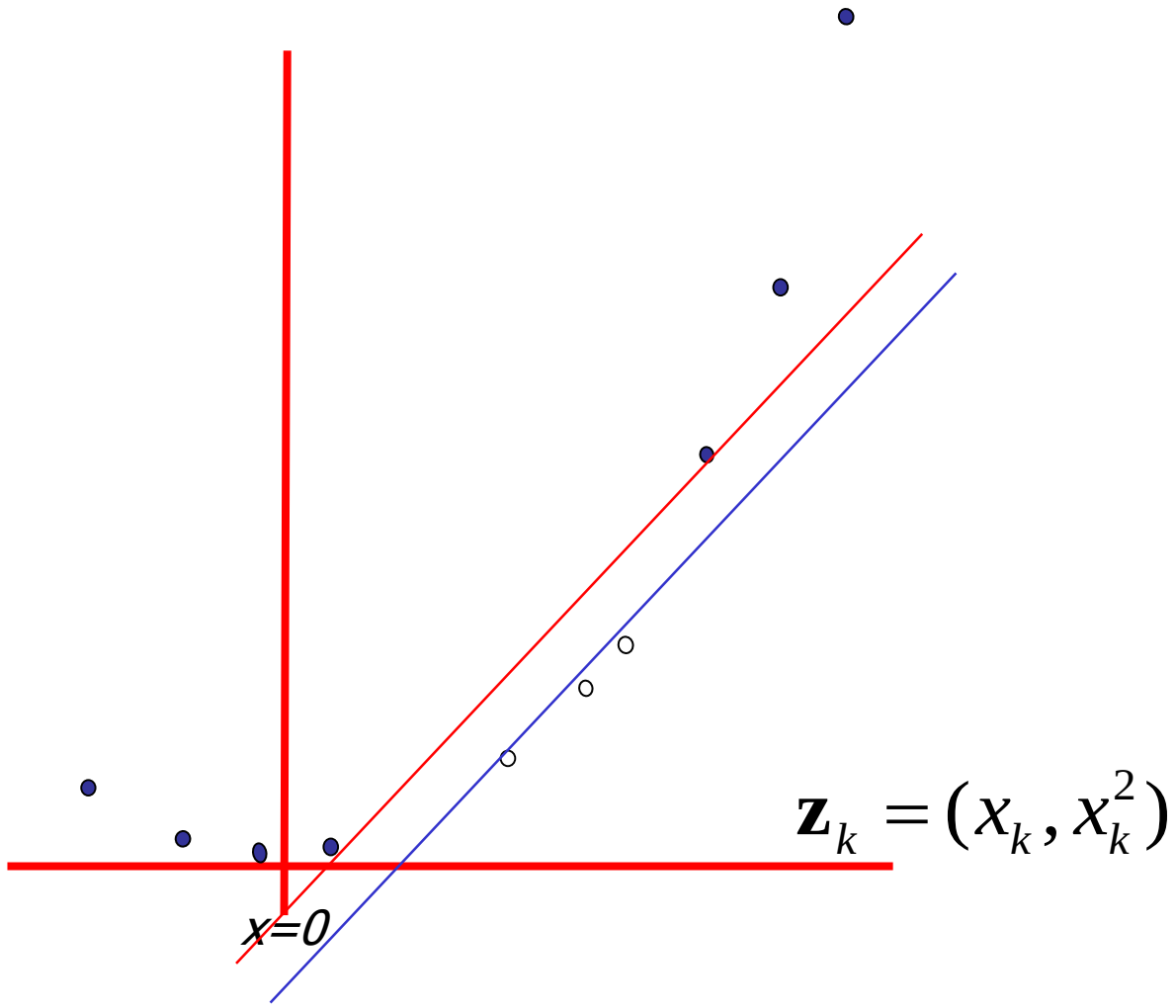
5. 서포트벡터머신

■ 비선형 SVM



5. 서포트벡터머신

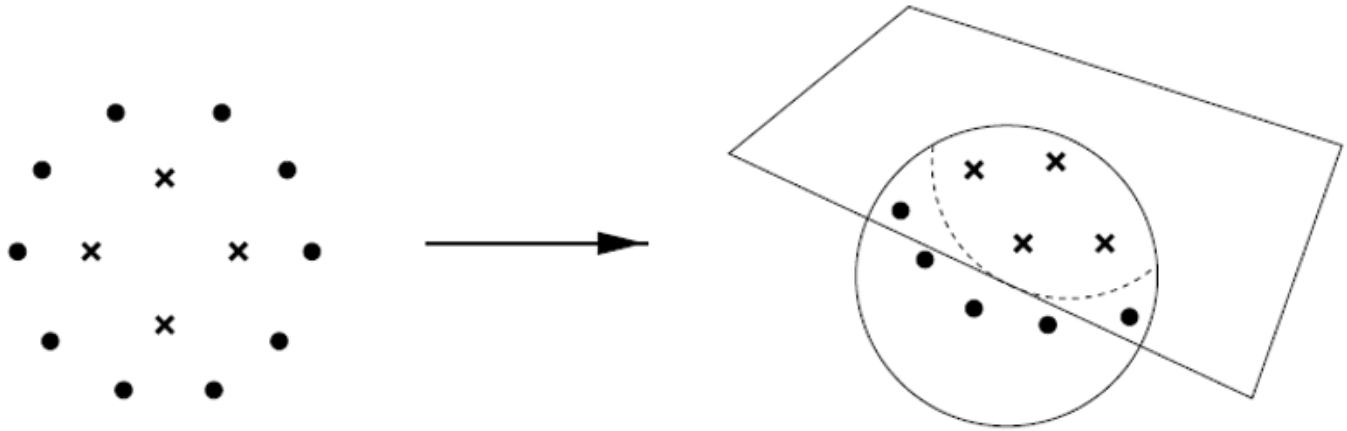
■ 비선형 SVM



■ 비선형 SVM

Kernel Trick

$$x \mapsto \phi(x) \qquad f(x) = w \cdot \phi(x) + b$$



Example of $\phi(x)$

$$\phi(\mathbf{x}) = [1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_3, x_1^2, x_2^2, x_3^2, \sqrt{2}x_1x_2, \sqrt{2}x_1x_3, \sqrt{2}x_2x_3]^T$$

■ 비선형 SVM의 수학적 표현

Linear SVM in a feature space

$$\begin{aligned} \min_{\mathbf{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\xi_i \\ \text{subject to} \quad & y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0,\ i = 1, \ldots, l. \end{aligned}$$

Optimal w

$$\mathbf{w} = \sum_{i=1}^{l} \alpha_i y_i \phi(\mathbf{x}_i)$$

■ 비선형 SVM의 수학적 표현

Classification function

$$\begin{aligned} & \mathbf{w}^T \phi(\mathbf{x}) + b \\ = & \sum_{i=1}^{l} \alpha_i y_i \phi(\mathbf{x}_i)^T \phi(\mathbf{x}) + b \\ = & \sum_{i=1}^{l} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b \end{aligned}$$

$\phi(\mathbf{x})$ 의 명시적 형태를 알 필요는 없음

다만, 커널함수(kernel function) K(x, x′) 가 필요함

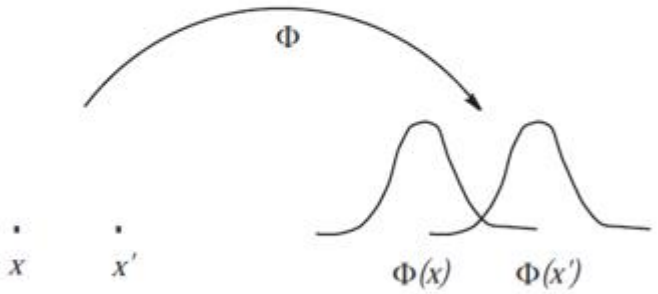
5. 서포트벡터머신

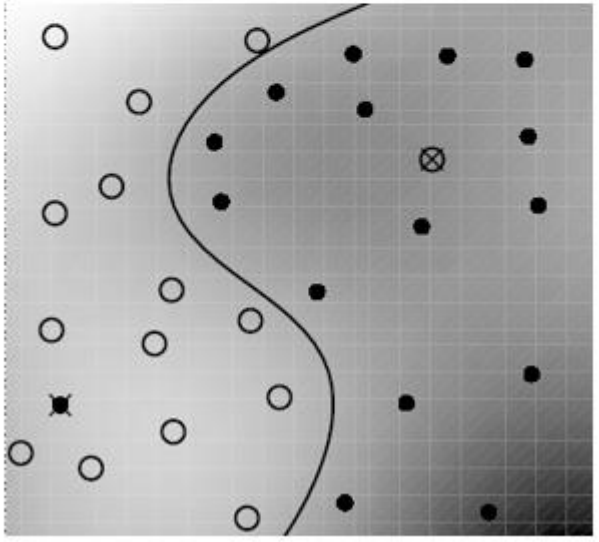
■ 대표적인 커널 함수

Polynomial kernel $K(\boldsymbol{x}, \boldsymbol{x}') = (\boldsymbol{x} \cdot \boldsymbol{x}' + 1)^d$

RBF (Radial basis function) kernel

$$K(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\gamma ||\boldsymbol{x} - \boldsymbol{x}'||^2)$$

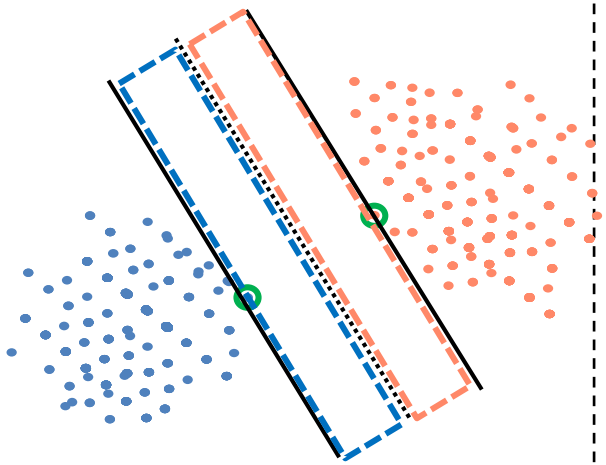




SVM 요약

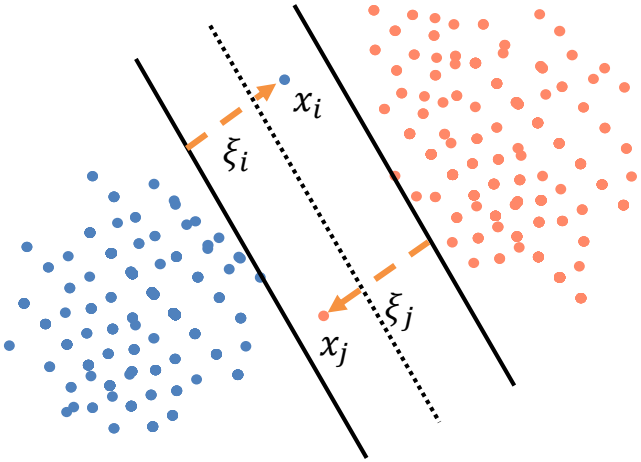
선형 SVM

- 데이터 :
  $y_i(w^T x_i - b) \geq 1$
- 최대 Margin :
  $\arg\min_{(w,b)} \|w\|$



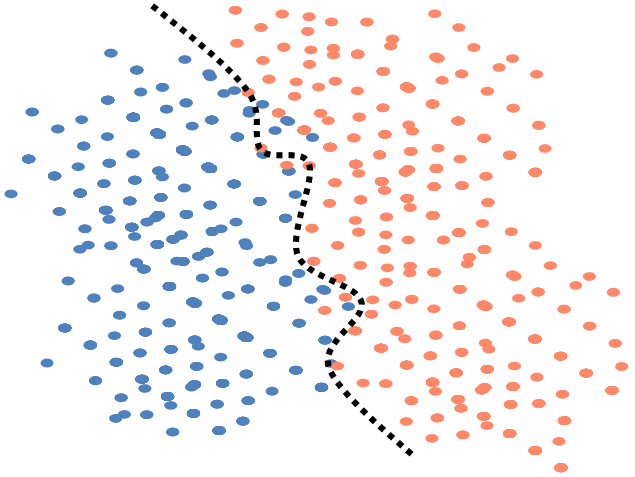
Non Separable SVM

- 데이터 :
  $y_i(w^T x_i - b) \geq 1 - \xi_i$
- 최대 Margin :
  $\arg\min_{(w,b,\xi)} \left\{ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i \right\}$



비선형 SVM

- 데이터 :
  $y_i\left(w^T K(x_i, x_j) - b\right) \geq 1 - \xi_i$
- 최대 Margin :
  $\arg\min_{(w,b,\xi)} \left\{ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i \right\}$



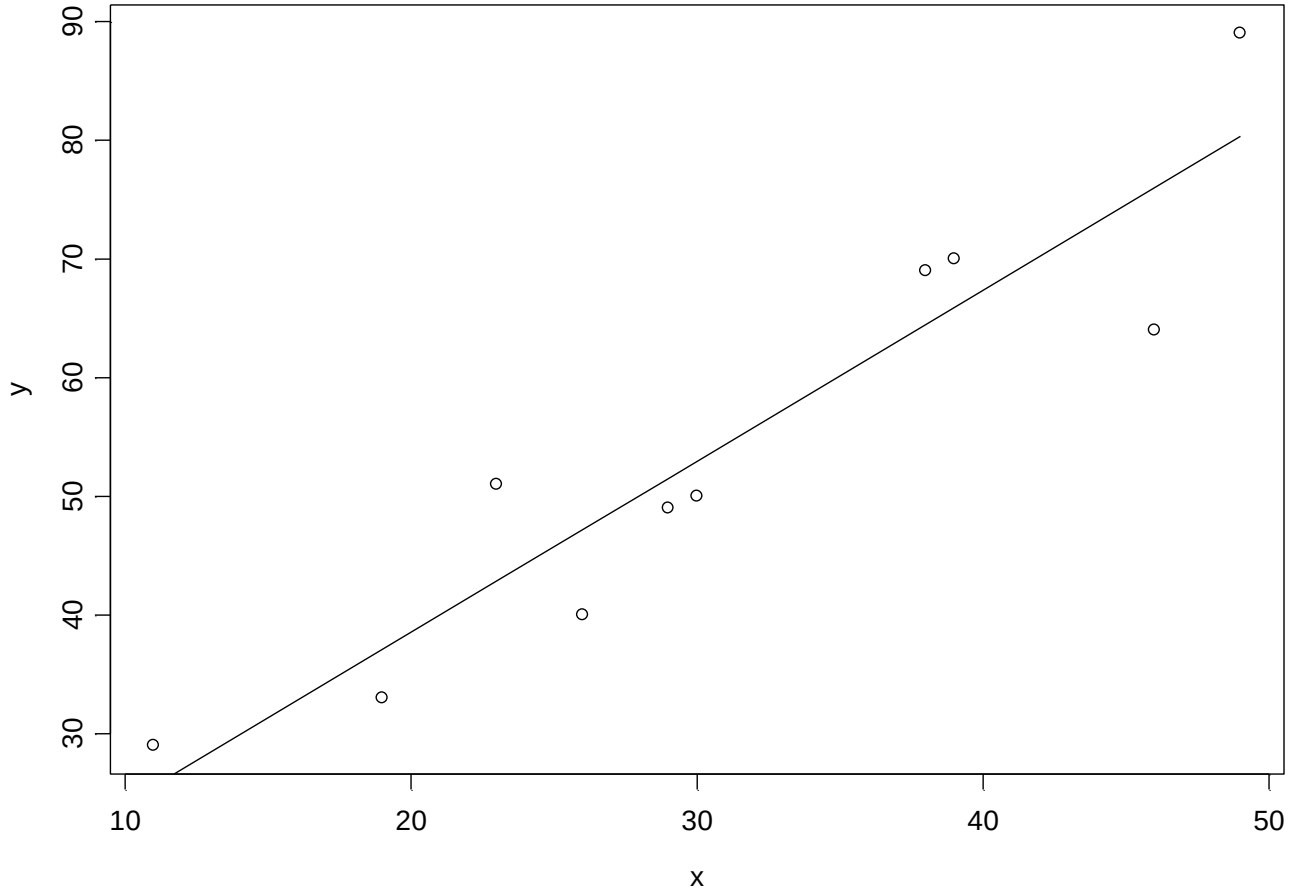
5. 서포트벡터머신

■ SVM 적합을 위한 절차

1. 데이터 표준화: (변수값-Min)/(Max-Min) 또는 (변수값-평균)/표준편차
2. 커널함수 선택 (RBF 커널이 주로 사용됨)
3. 조절모수 선택 (C , γ) -> CV(cross-validation) 이용
4. 선택된 조절모수 값을 사용하여 전체 데이터로부터 SVM 구축
5. 새로운 데이터에 대한 예측

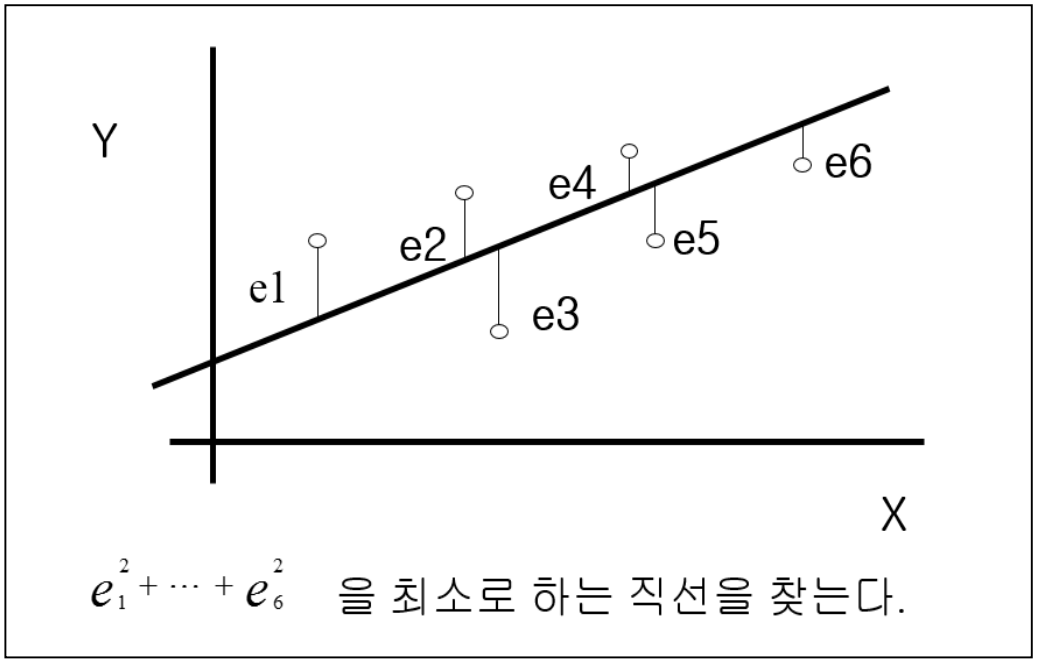
5. 서포트벡터머신

■ (Recall) 단순선형회귀



5. 서포트벡터머신

■ (Recall) 최소제곱법

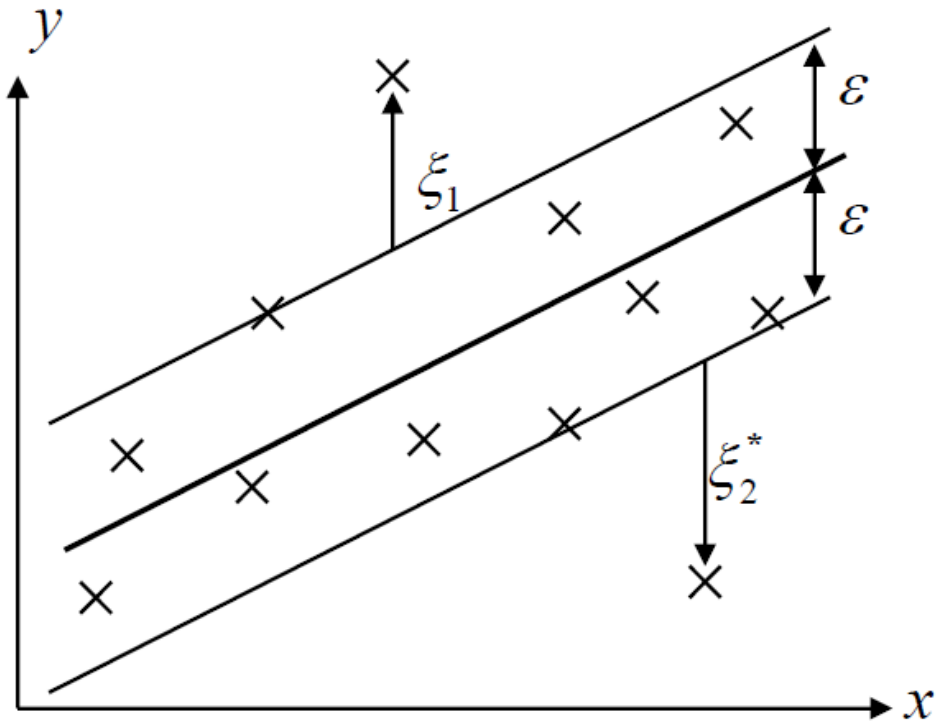


$$\min \sum_{i=1}^{m} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{m} (y_i - (\mathbf{w} \cdot \mathbf{x}_i + b))^2$$

5. 서포트벡터머신

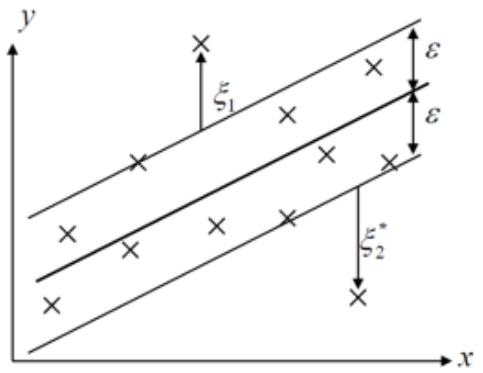
■ SVR의 아이디어

ε insensitive 손실을 최소화 하면서 가장 'flat'한 함수 찾음



■ SVR의 수학적 표현

- **자료** $(\mathbf{x}_i, y_i) \quad i = 1,\ldots,m$
- **모형** $f(\mathbf{x}, \omega) = \mathbf{w} \cdot \mathbf{x} + b$
- **최적화 방법**



Minimize

$$\frac{1}{2} \| \mathbf{w} \|^2 + C \sum_{i=1}^{m} (\xi_i + \xi_i^*)$$

subject to

$$y_i - (\mathbf{w} \cdot \mathbf{x}_i) - b \le \varepsilon + \xi_i$$
$$(\mathbf{w} \cdot \mathbf{x}_i) + b - y_i \le \varepsilon + \xi_i^*$$
$$\xi_i \, , \xi_i^* \ge 0, i = 1,\ldots,m$$

■ SVR의 수학적 표현

- **Optimal solution**

$$w = \sum_{i=1}^{m} (\alpha_i - \alpha_i^*) x_i$$

- **추정함수**

$$f(x) = \sum_{i=1}^{m} (\alpha_i - \alpha_i^*) \langle x_i, x \rangle + b$$

Note

- Optimal solution w는 support vector들에만 의존
- 선형 SVR의 최종 회귀함수 형태는 내적 $< \cdot, \cdot >$으로 표현

➔ Kernel trick을 통해 비선형 SVR 가능

■ SVR의 수학적 표현

- **Kernel trick for nonlinear function estimation**

$$x \mapsto \Phi(x) \ . \qquad f(x) = w \cdot \phi(x) + b.$$

$$w = \sum_{i=1}^{m} (\alpha_i - \alpha_i^*) \phi(x_i)$$

$$f(x) = \sum_{i=1}^{m} (\alpha_i - \alpha_i^*) k(x_i, x) + b.$$

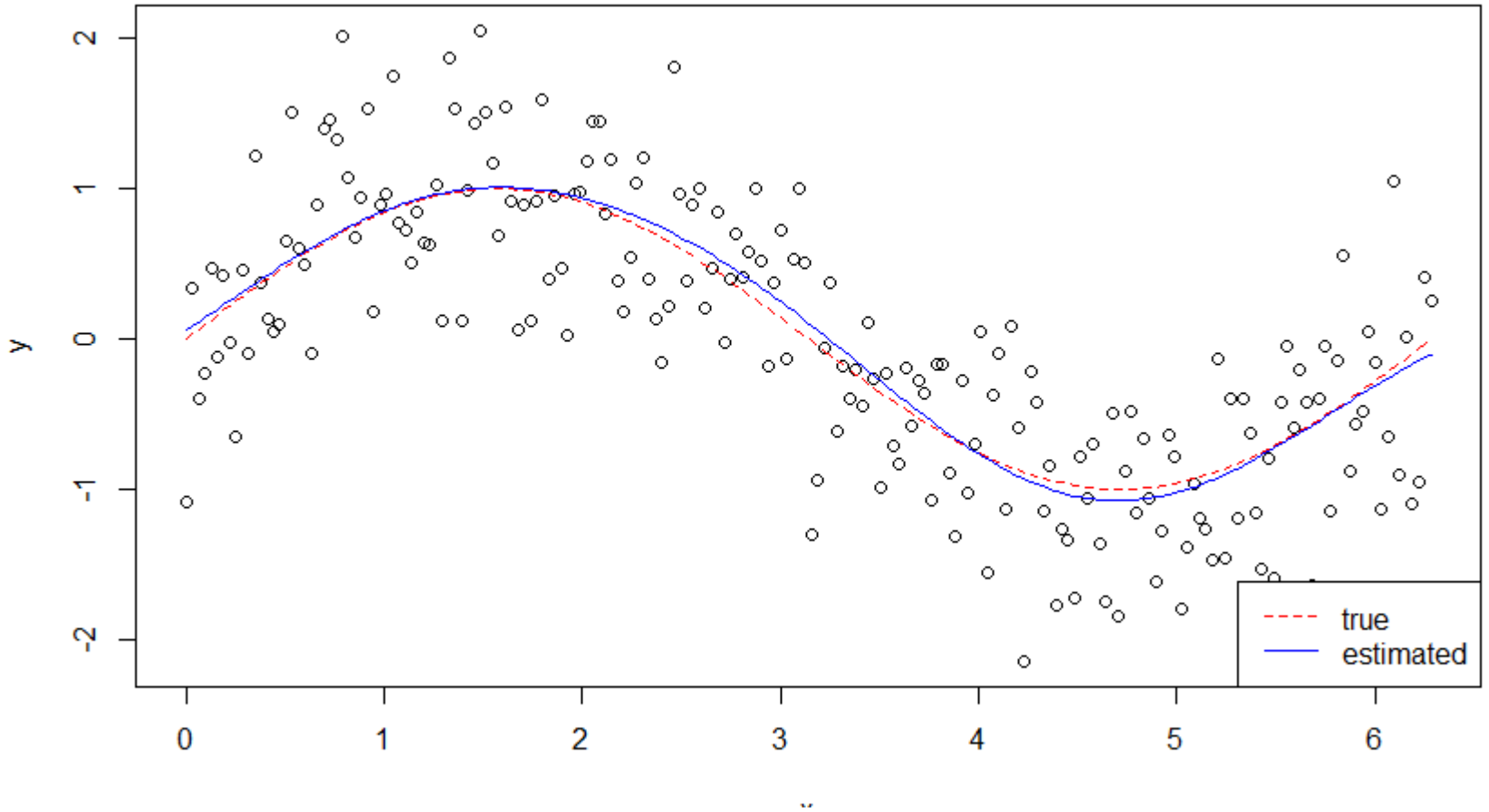
- **Kernel 함수로는 polynomial kernel 또는 RBF kernel이 주로 사용됨**

$$K(\boldsymbol{x}, \boldsymbol{x}') = (\boldsymbol{x} \cdot \boldsymbol{x}' + 1)^d$$

$$K(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\gamma ||\boldsymbol{x} - \boldsymbol{x}'||^2)$$

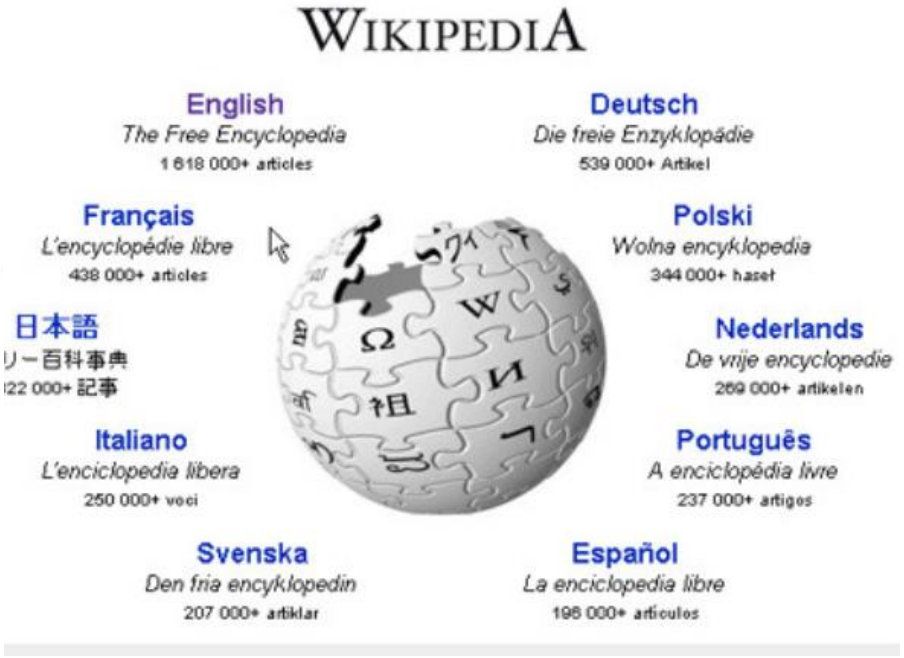
5. 서포트벡터머신

■ SVR 예시



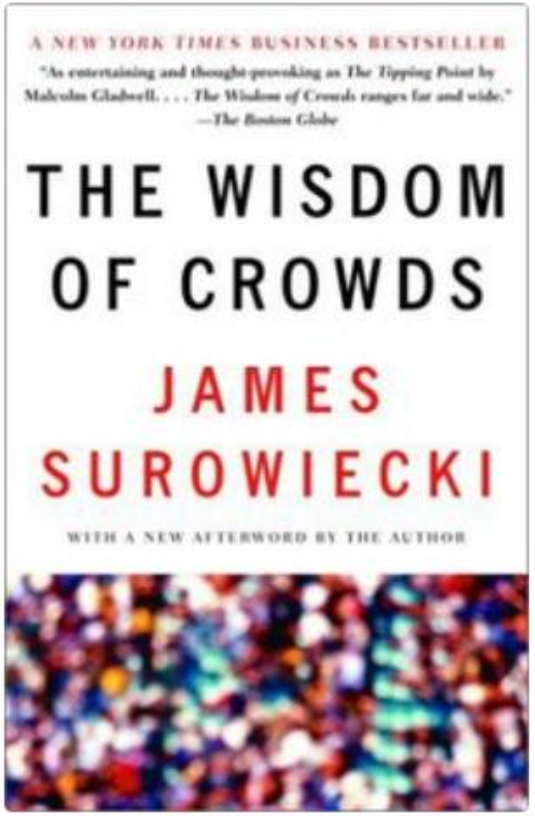
6. 앙상블 모형

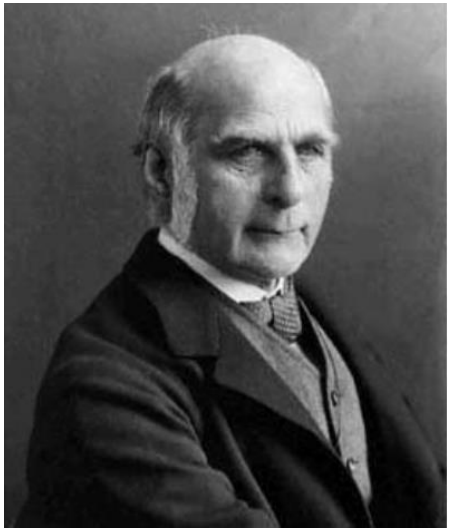
■ 집단지성의 파워



6. 앙상블 모형

■ 집단지성의 파워





800명의 집단지성

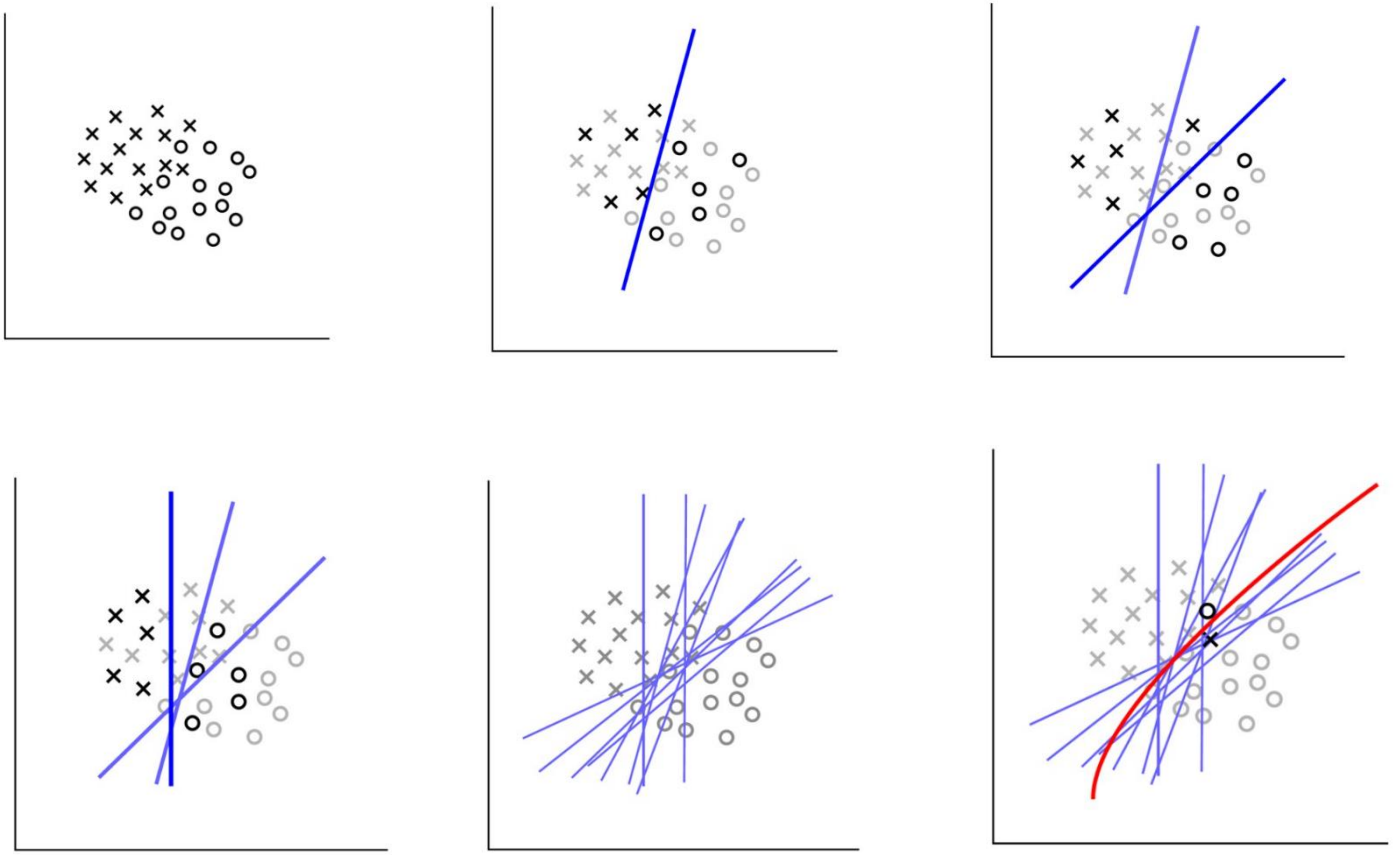
Crowd's guess:
1197 파운드

Actual weight:
1198 파운드

6. 앙상블 모형

■ 앙상블 러닝의 아이디어

- 간단한 모형을 결합함으로써 분류 정확도가 높은 복잡한 모형을 구축할 수 있음



6. 앙상블 모형

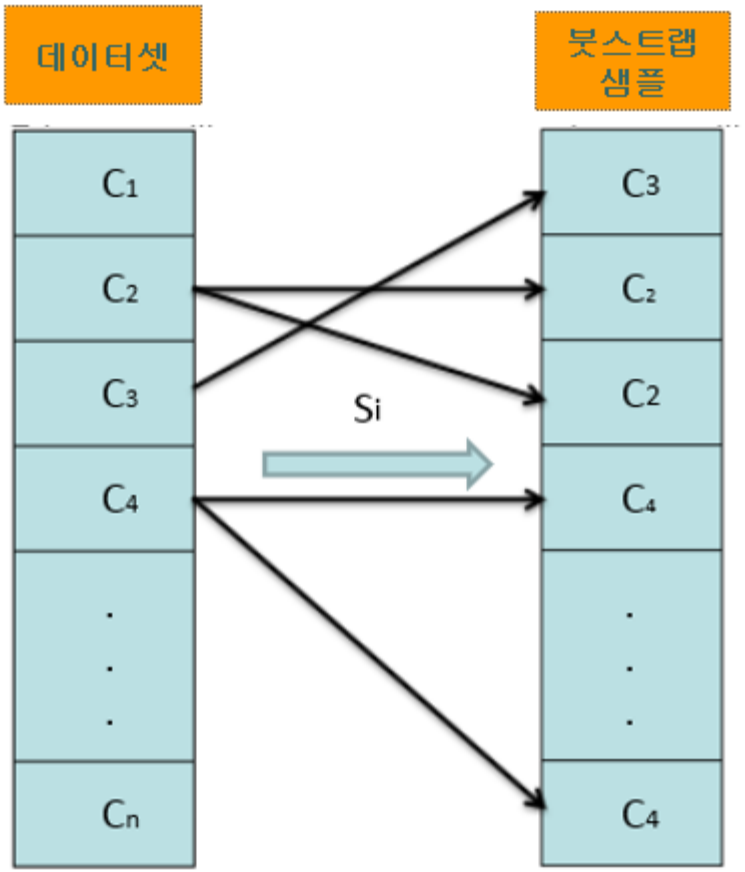
■ 배깅(Bagging)

- Bagging : Bootstrap + Aggregation
- Introduced by Breiman (1996)

- 훈련용 데이터로부터 B개의 붓스트랩(bootstrap) 표본을 구성
  - bootstrap sample: 임의복원추출로 생성된 자료
- 각 붓스트랩 표본으로부터 회귀모형 또는 분류모형 구축
- 최종 예측모형
  - 회귀: 각 모형의 결과를 단순평균
  - 분류: majority vote
- **분산을 줄임으로써 unstable한 기저모형의 성능을 개선 (예, 의사결정나무)**

6. 앙상블 모형

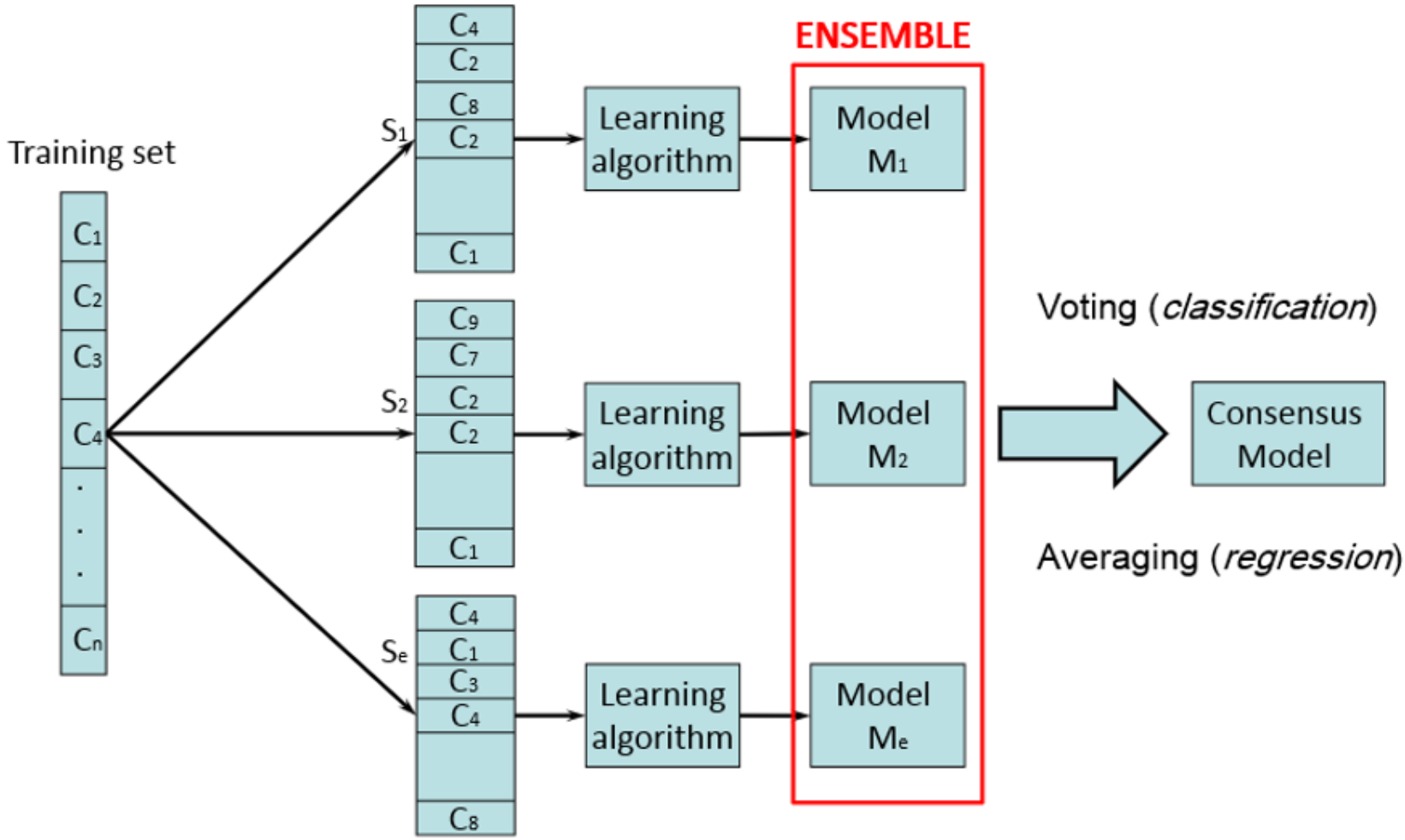
■ 붓스트랩표본



- 모든 개체가 표본으로 추출될 확률은 같음
- 어떤 개체는 중복해서 뽑힐 수도 있고
- 어떤 개체는 전혀 뽑히지 않을수도 있음
- 표본추출 확률: $1-(1-\frac{1}{n})^n$
  (대략 63%가 표본으로 추출)

6. 앙상블 모형

■ 배깅 모형 개념도



6. 앙상블 모형

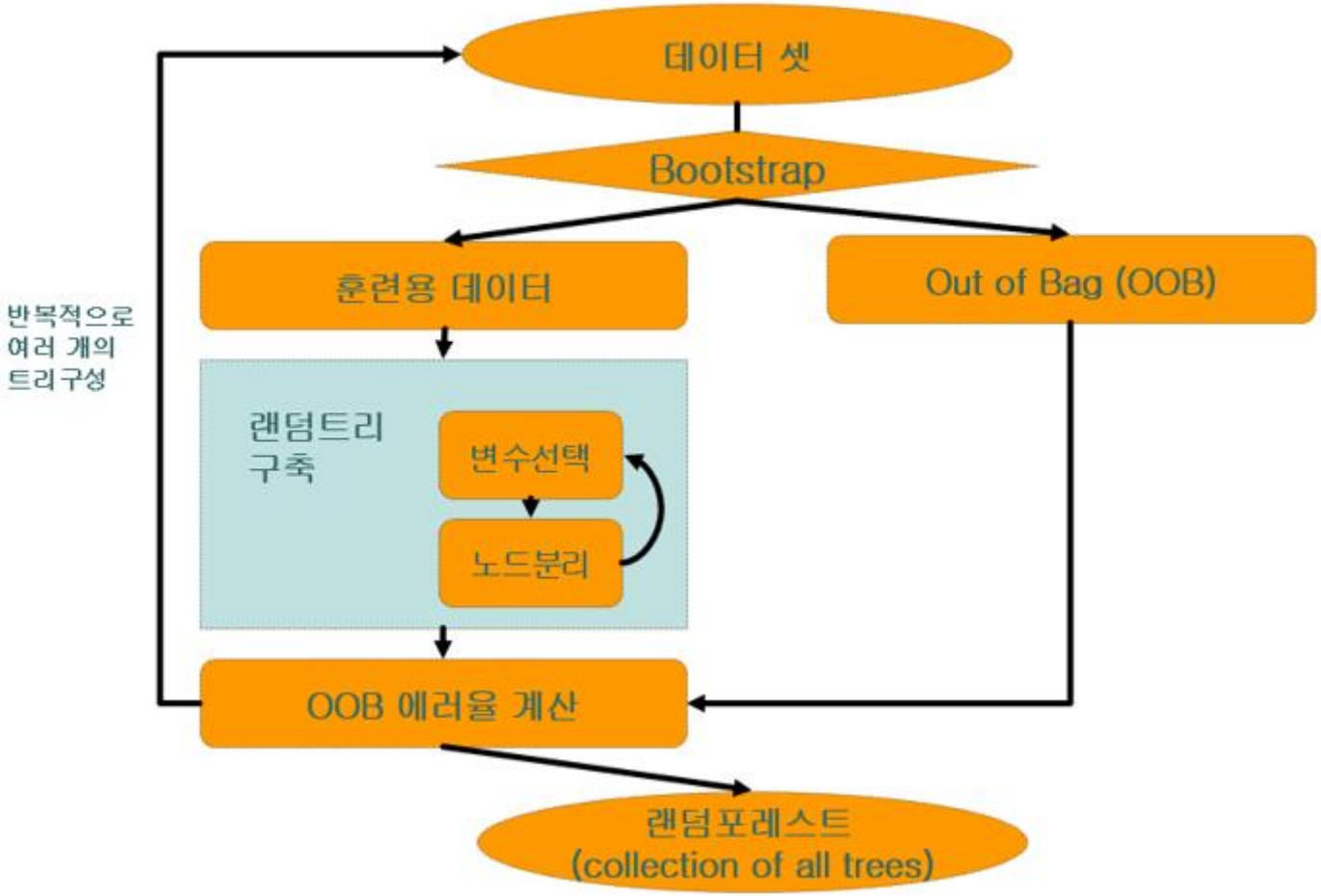
■ 랜덤 포레스트

- Bagging의 문제
  - 대량의 데이터인 경우 (거의) 같은 분류학습자를 도출
  - 이러한 모형들을 평균화하는 것은 도움이 되지 못함
- Random Forest

–앙상블을 구성하는 개별 모형들의 다양성을 제고하기 위해 트리 구성시 각 노드에서 일부의 변수들을 랜덤하게 선정

6. 앙상블 모형

■ 랜덤 포레스트



6. 앙상블 모형

■ 랜덤 포레스트의 특징

- 기존의 모형들보다 모형의 정확도가 좋다.
- 매우 큰 데이터에 대해서도 학습이 효율적이다.
- 많은 수(수천 개)의 변수를 변수를 제거하지 않고도 다룰 수 있다.
- 변수의 중요도에 대한 추정치를 제공한다.
- 모형 구축과정에서 일반화오류율(generalization error)에 대한 불편추정치(unbiased estimate)를 제공한다.

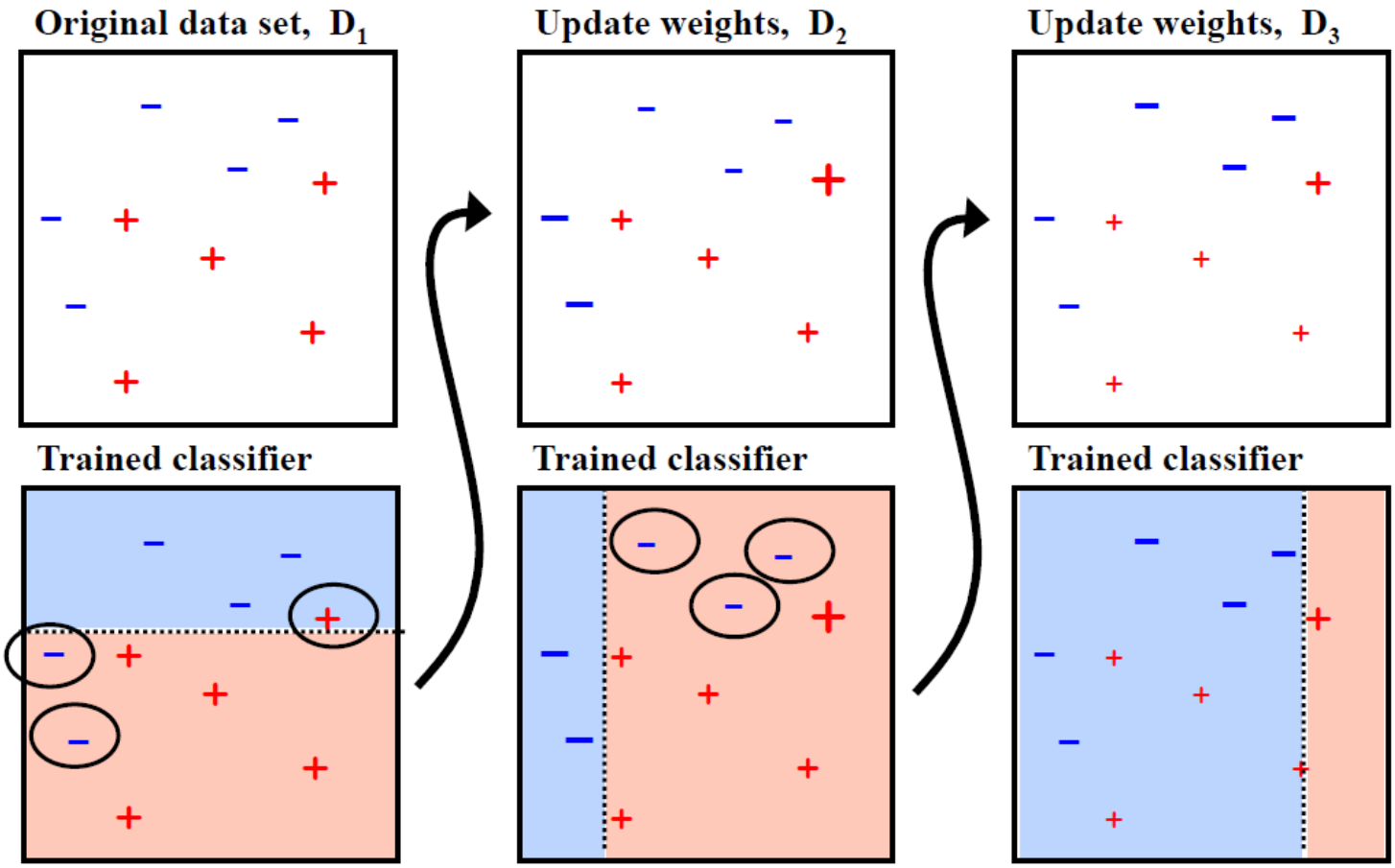
6. 앙상블 모형

■ 부스팅(Boosting)

- 앙상블 방법의 대표적인 방법론으로서 다양한 알고리즘의 부스팅이 존재
- 분류문제 뿐만 아니라 회귀문제에 적합한 부스팅도 있음
- 기저모형 (흔히 decision stump같은 weak learner)들을 순차적으로 구축한 후 , 최종적으로 병합(가중평균)하여 의사결정
- 각 기저모형들은 전 스텝에서 만들어진 기저모형에 의존
- 특히, 전 스텝에서 오분류된 개체에 더 많은 가중치를 주고 모형을 구축하게 됨
- 가장 성공적인 부스팅알고리즘으로서 이후 부스팅에 대한 많은 연구를 촉발시킨 것은 것은 AdaBoost (Freund and Schapire (1995))라는 알고리즘임

6. 앙상블 모형

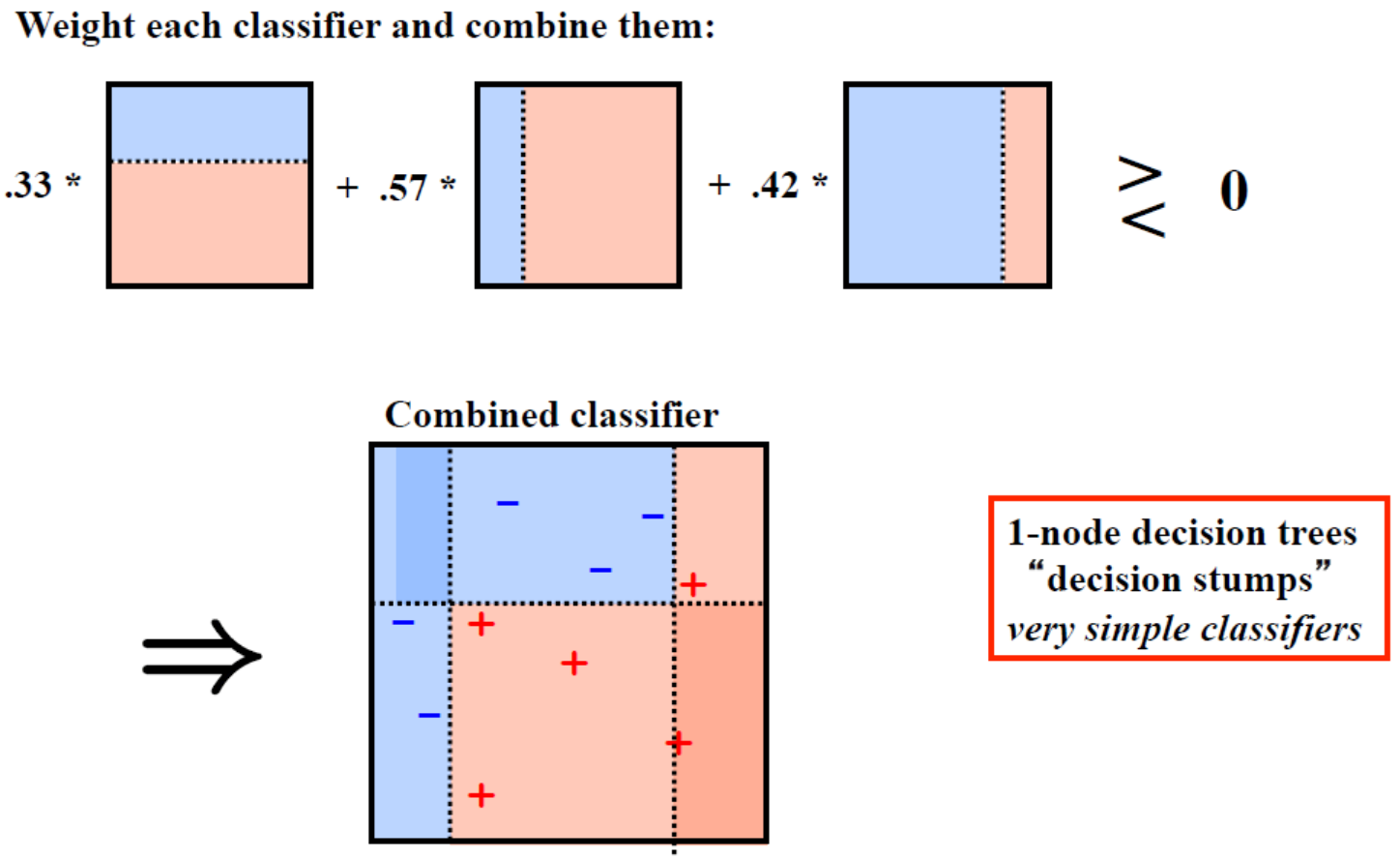
■ 부스팅(Boosting)



출처: http://sli.ics.uci.edu/Classes/2012F-273a?action=download&upname=10-ensembles.pdf

6. 앙상블 모형

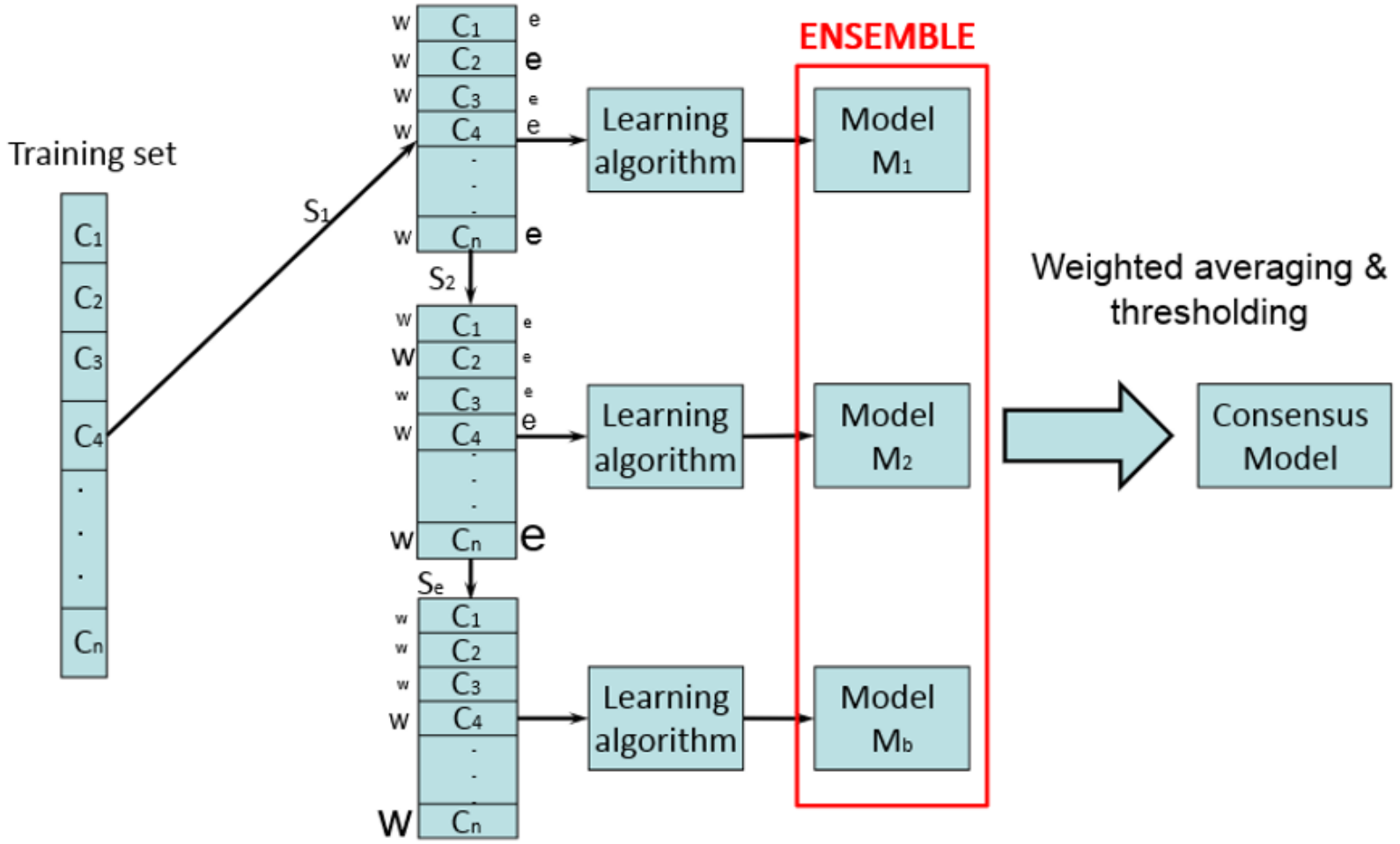
■ 부스팅(Boosting)



출처: http://sli.ics.uci.edu/Classes/2012F-273a?action=download&upname=10-ensembles.pdf

6. 앙상블 모형

■ AdaBoost 개념도



7. 예측모형의 평가

■ Metrics for Performance Evaluation: Confusion Matrix

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a	b
	Class=No	c	d

a: TP (True Positive)
b: FN (False Negative)
c: FP (False Positive)
d: TN (True Negative)

■ Metrics for Performance Evaluation: Confusion Matrix

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a (TP)	b (FN)
	Class=No	c (FP)	d (TN)

- Most widely-used metric:

$$\text{Accuracy} = \frac{a+d}{a+b+c+d} = \frac{TP+TN}{TP+TN+FP+FN}$$

■ Limitation of Accuracy

- Consider a 2-class problem
  - Class 0을 가지는 데이터의 개수는 9,990개
  - Class 1을 가지는 데이터의 개수는 10개

- 어떤 모델은 무조건 Class 0으로 예측을 한다.
  이 모델의 accuracy는 9,990 / 10,000 = 99.9%

7. 예측모형의 평가

■ Cost matrix

	PREDICTED CLASS		
ACTUAL CLASS	C(i|j)	**Class=Yes**	**Class=No**
	Class=Yes	C(Yes|Yes)	C(No|Yes)
	Class=No	C(Yes|No)	C(No|No)

$C(i|j)$: class i인 데이터를 class j로 예측할 때의 비용(cost)

7. 예측모형의 평가

■ Cost matrix

Cost Matrix	PREDICTED CLASS		
	C(i|j)	+	-
ACTUAL CLASS	+	-1	100
	-	1	0

Model M_1	PREDICTED CLASS		
		+	-
ACTUAL CLASS	+	150	40
	-	60	250

Accuracy = 80%

Cost = 3910

Model M_2	PREDICTED CLASS		
		+	-
ACTUAL CLASS	+	250	45
	-	5	200

Accuracy = 90%

Cost = 4255

7. 예측모형의 평가

■ Assessment Index

	PREDICTED CLASS		
ACTUAL CLASS		**Yes**	**No**
	Yes	a	b
	No	c	d

민감도(Sensitivity, True Positive Rate)

$$\frac{a}{a+b}$$

특이도(Specificity, True Negative Rate)

$$\frac{d}{c+d}$$

Precision(p) $$\frac{a}{a+c}$$

7. 예측모형의 평가

■ 사후확률에 대한 분류 임계값 설정

y		$P(y=1)$		$\hat{y}(0.50)$	$\hat{y}(0.25)$
1		0.75		1	1
0		0.12		0	0
1		0.93		1	1
1		0.53		1	1
0		0.15		0	0
0		0.31		0	1
0		0.12		0	0
0		0.30		0	1
1		0.41		0	1
1		0.75		1	1

Discriminant
판별
Modeling

Classification
분류
Cut-off value
(Threshold)

7. 예측모형의 평가

■ 임계값에 따른 분류 정확도

$\hat{y}(0.50)$

Actual \ Predicted	0	1	
0	5	0	5
1	1	4	5
	6	4	10

$\hat{y}(0.25)$

Actual \ Predicted	0	1	
0	3	2	5
1	0	5	5
	3	7	10

오류율 (Error rate)

= (false negative + false positive)/(grand total) = **(1+0)/10 = 10%** **(0+2)/10 = 20%**

정확도 (Accuracy)

= (true negative + true positive)/(grand total) = **(5+4)/10 = 90%** **(3+5)/10 = 80%**

민감도 (Sensitivity)

= (true positive)/(total actual positive) = **4/5 = 80%** **5/5 = 100%**

특이도 (Specificity)

= (true negative)/(total actual negative) = **5/5 = 100%** **3/5 = 60%**

7. 예측모형의 평가

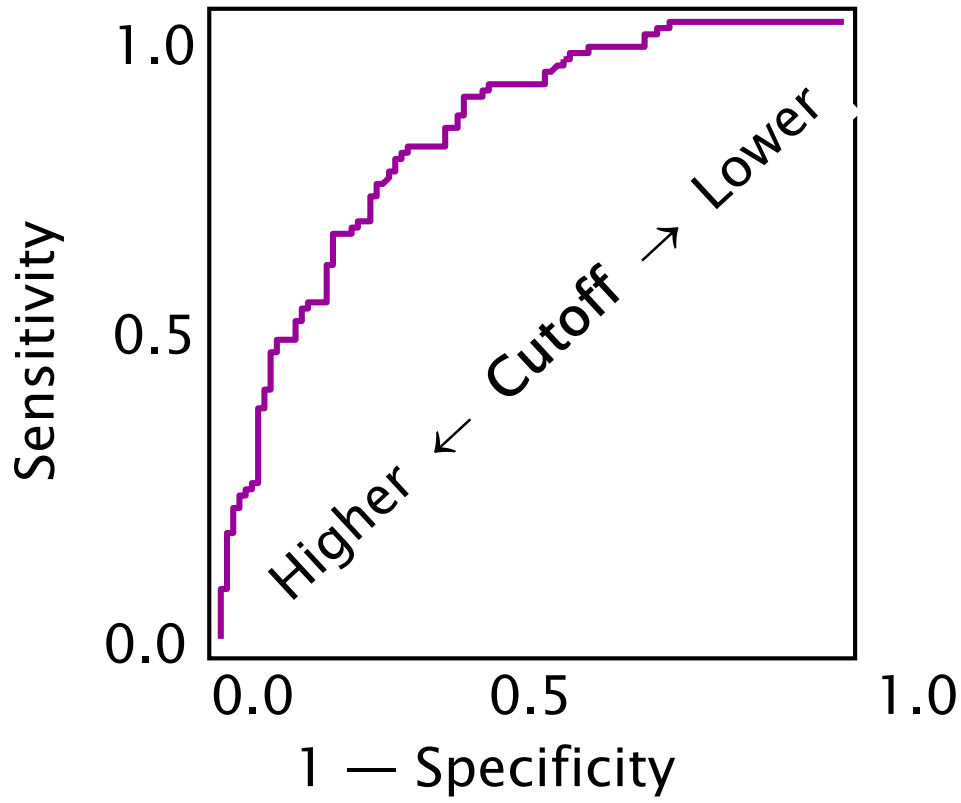
■ ROC Curve

X축: 1-특이도, Y축: 민감도

■ AUC or ROC index

Area Under the ROC curve

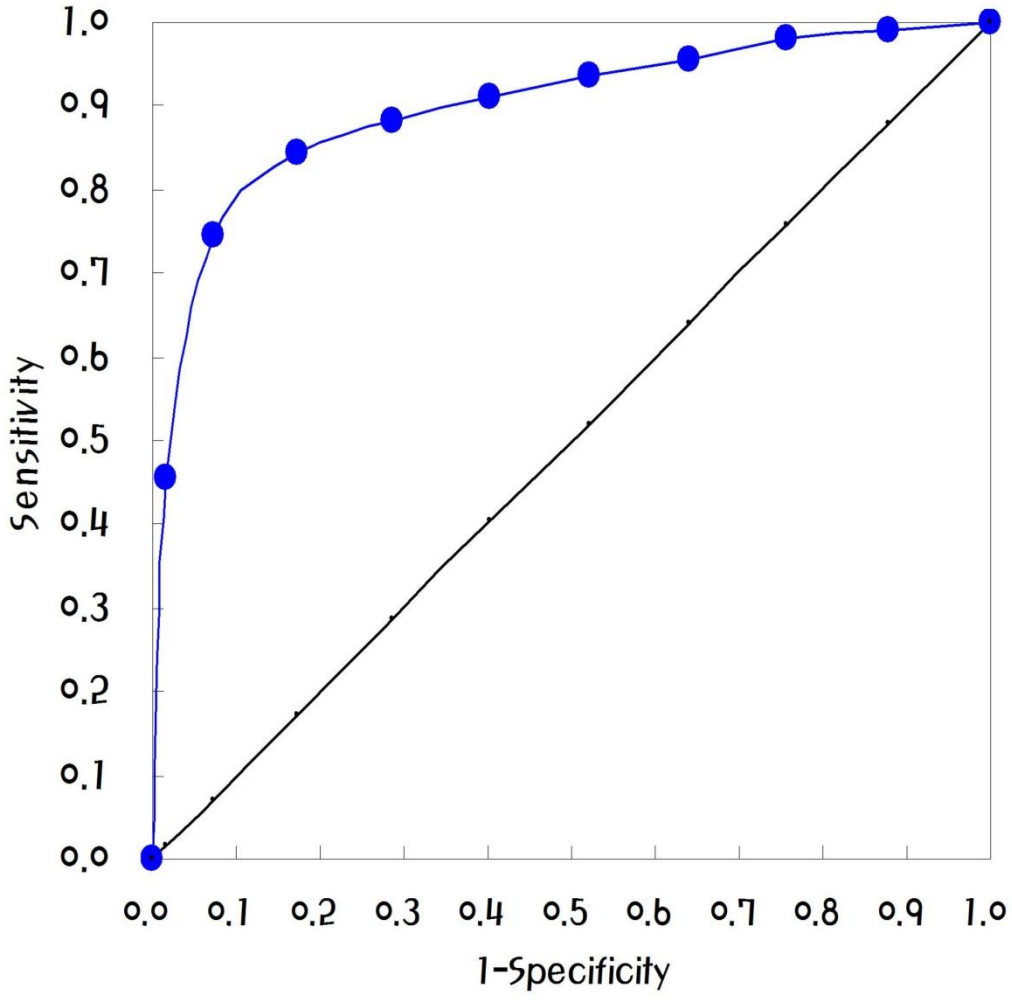
- Ideal: AUC = 1
- Random guess: AUC = 0.5



7. 예측모형의 평가

■ ROC Curve

민감도	1-특이도	C
0.000	0.000	0.000
0.457	0.016	0.007
0.745	0.072	0.041
0.845	0.172	0.085
0.882	0.287	0.101
0.911	0.403	0.106
0.937	0.521	0.110
0.955	0.640	0.114
0.982	0.757	0.115
0.990	0.879	0.120
1.000	1.000	0.121
합계		0.921

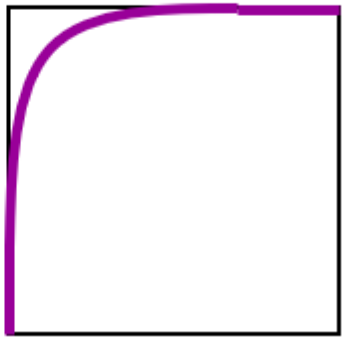


7. 예측모형의 평가

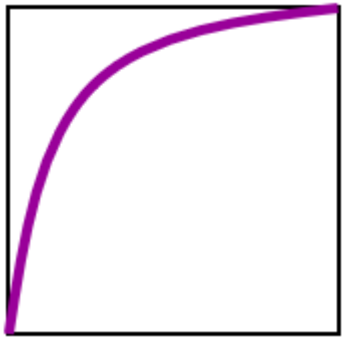
■ ROC Curve

ROC

매우 좋음



좋음



나쁨