

Machine Learning

6강

# 앙상블 학습

컴퓨터과학과 이관용 교수

# 학습목차

- 01 앙상블 학습의 개념
- 02 배깅과 보팅
- 03 부스팅
- 04 결합 방법

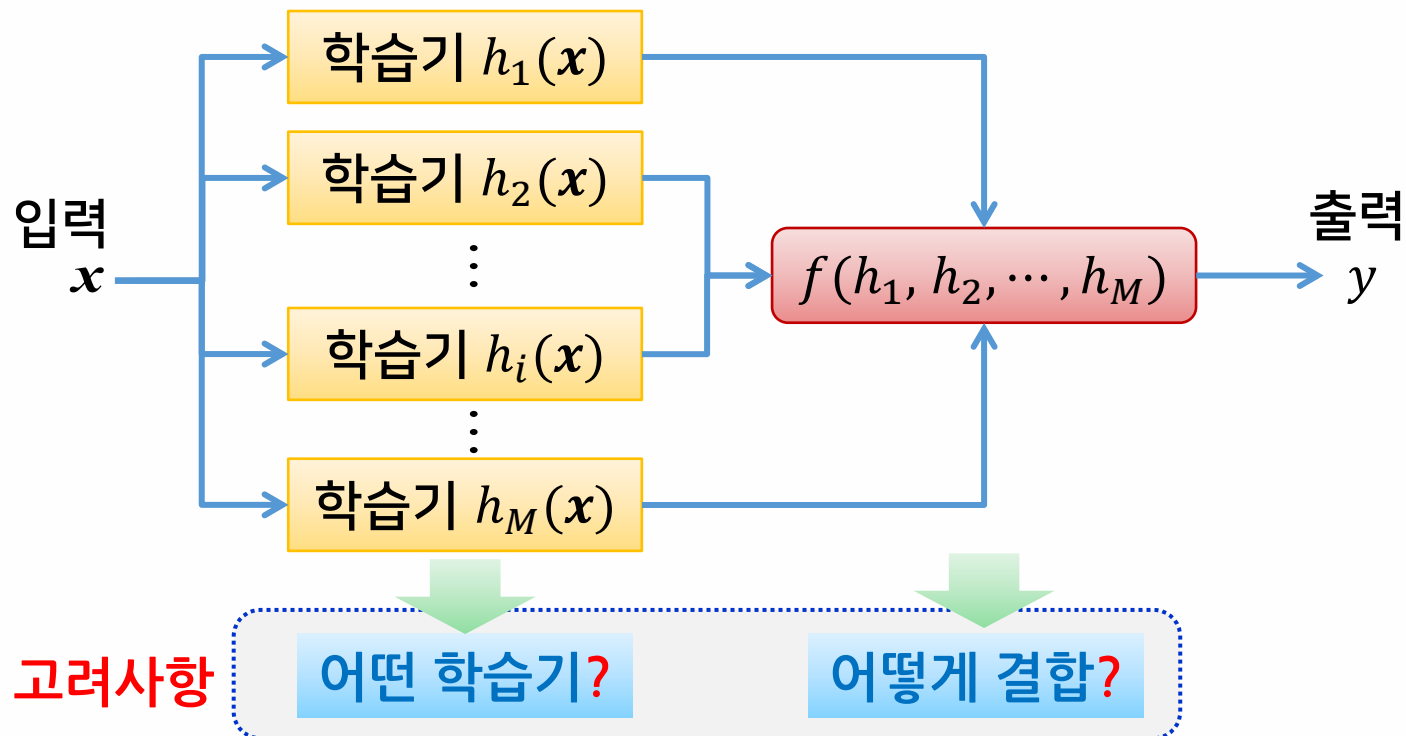
1

# 앙상블 학습의 개념

# 학습기 결합?

## ○ 앙상블 학습 ensemble learning

- 선형 분류기와 같은 간단한 학습기로 학습을 수행하지만, 복수 개의 학습기를 결합함으로써 결과적으로 더 좋은 성능을 가진 학습기를 만드는 방법



# 학습기 결합에서의 고려사항

## 학습기의 차별화 방법

### 학습 알고리즘의 차별화

접근 방법이 서로 다른 학습기 선택 → 베이즈 분류기 & K-NN, 신경망 & SVM

### 모델 선택과 관련된 파라미터의 차별화

K값이 서로 다른 복수의 K-NN, 은닉층의 뉴런 수가 서로 다른 복수의 MLP

### 학습 데이터의 차별화

같은 모델을 사용하되 학습 데이터 집합을 달리하여 복수 개의 학습기를 생성

## 학습기 결합 방법

### 병렬적 결합

각 학습기의 결과를 한 번에 모두 함께 고려하여 하나의 최종 결과를 생성

### 순차적 결합

각 학습기의 결과를 단계별로 결합

# 앙상블 학습의 개요

입력  $x$ 에 대한  $i$ 번째 학습기의 출력

$$h_i(x) = h(x, \theta_i)$$

$M$ 개 학습기 결합에 의한 최종 결과

$$f_M(x) = f(h_1(x), h_2(x), \dots, h_M(x))$$

학습기 출력  $f(x)$ 와 목표출력  $t$ 의 오차

$$e(x) = e(f(x), t)$$

$$e(x) = e(f(x), f^*(x)) \quad f^*(x) \rightarrow \text{목표 출력을 제공하는 실제 시스템}$$

제곱오차 함수

$$e^2(x, \theta) = \{f(x; \theta) - t\}^2$$

모든 입력 공간에 대한 일반화 오차

$$E_{\text{gen}} = E_x[e(f(x), t)] = \int e(f(x), f^*(x))p(x)dx$$

# 앙상블 학습의 개요

## 학습 데이터 생성 방법에 따른 분류

### 필터링 filtering에 의한 방법

→ 초기 부스팅 방법, 캐스케이딩 방법

각 학습기의 학습 때마다 새로운 데이터를 생성하고, 이를 이미 학습이 완료된 학습기에 적용하여 제대로 처리되지 못하는 데이터들만 필터링하여 학습

### 리샘플링 resampling에 의한 방법

→ 배깅 방법, MadaBoost 방법

각 학습기의 학습 때마다 학습 데이터를 새로 생성하지 않고, 주어진 전체 학습 데이터로부터 일부 집합을 추출하여 각 학습기를 학습

### 가중치 조정 reweighting에 의한 방법

→ AdaBoost 방법

모든 학습기에 대해 동일한 학습 데이터를 사용하되, 각 데이터에 대해 가중치를 주어 학습에 대한 영향도를 조정

2

## 배깅과 보팅



## 배깅 bagging에 의한 학습

- 부트스트랩 bootstrap 방법을 앙상블 학습에 적용한 것
  - bagging → “bootstrap aggregating”의 약자
  - 부트스트랩 → 제한된 데이터 집합을 이용하여 시스템의 학습과 평가를 동시에 수행하기 위한 리샘플링 기법
- 배깅에 의한  $M$ 개의 서로 다른 학습기의 학습 과정
  - ①  $N$ 개의 데이터로 이루어진 학습 데이터 집합  $X$ 를 준비하고, 학습을 위한 학습기 모델을 정의한다.  
 $M$ 개의 학습기를 각각 학습시키기 위해 사용될 데이터 집합의 크기  $\tilde{N}$  ( $\tilde{N} \leq N$ )을 정한다.

## 배깅에 의한 학습

- 배깅에 의한  $M$ 개의 서로 다른 학습기의 학습 과정
  - ②  $i$ 번째 학습기  $h_i(x)$  모델의 파라미터를 초기화하고, 학습 데이터 집합  $X$ 로부터  $\tilde{N}$ 개의 데이터를 랜덤하게 선출하여 데이터 집합  $X_i$ 를 만든다. 이때 같은 데이터가 중복해서 선출되는 것도 허락한다(복원 추출).
  - ③ 데이터 집합  $X_i$ 를 이용한 학습을 수행하여 최적화된 파라미터  $\theta_i$ 를 찾아  $i$ 번째 학습기를 위한 판별함수  $h_i(x, \theta_i)$ 를 얻는다.
  - ④ ②~③ 과정을  $M$ 번 반복하여 서로 다른  $M$ 개의 학습기를 생성하고, 이들을 결합하여 최종 판별함수  $f(h_1, h_2, \dots, h_M)$ 을 찾는다.

# 배깅에 의한 학습

## ○ 고려사항

### □ 데이터 집합의 크기 $\tilde{N}$

- ✓ 주어진 전체 학습 데이터 집합  $X$ 의 크기  $N$ 이 충분히 크지 않으면  $\tilde{N}$ 과  $N$ 은 같은 값으로 지정
- ✓ 복원추출을 사용하므로 매 단계마다 생성되는 데이터의 집합은 동일하지 않음

### □ 학습에 사용될 학습기의 모델

- ✓ 학습기에 의해 찾아지는 판별함수가 데이터 집합의 변화에 민감한 모델을 선택하는 것이 바람직
- ✓ 예: 다층 퍼셉트론, 최근접이웃 분류기 등

## 보팅에 의한 결합

### ○ 보팅법 voting, "committee machine"

- $M$ 개의 학습기 결과를 모두 동일한 정도로 반영하여 평균한 결과를 얻는 방법("단순평균법")

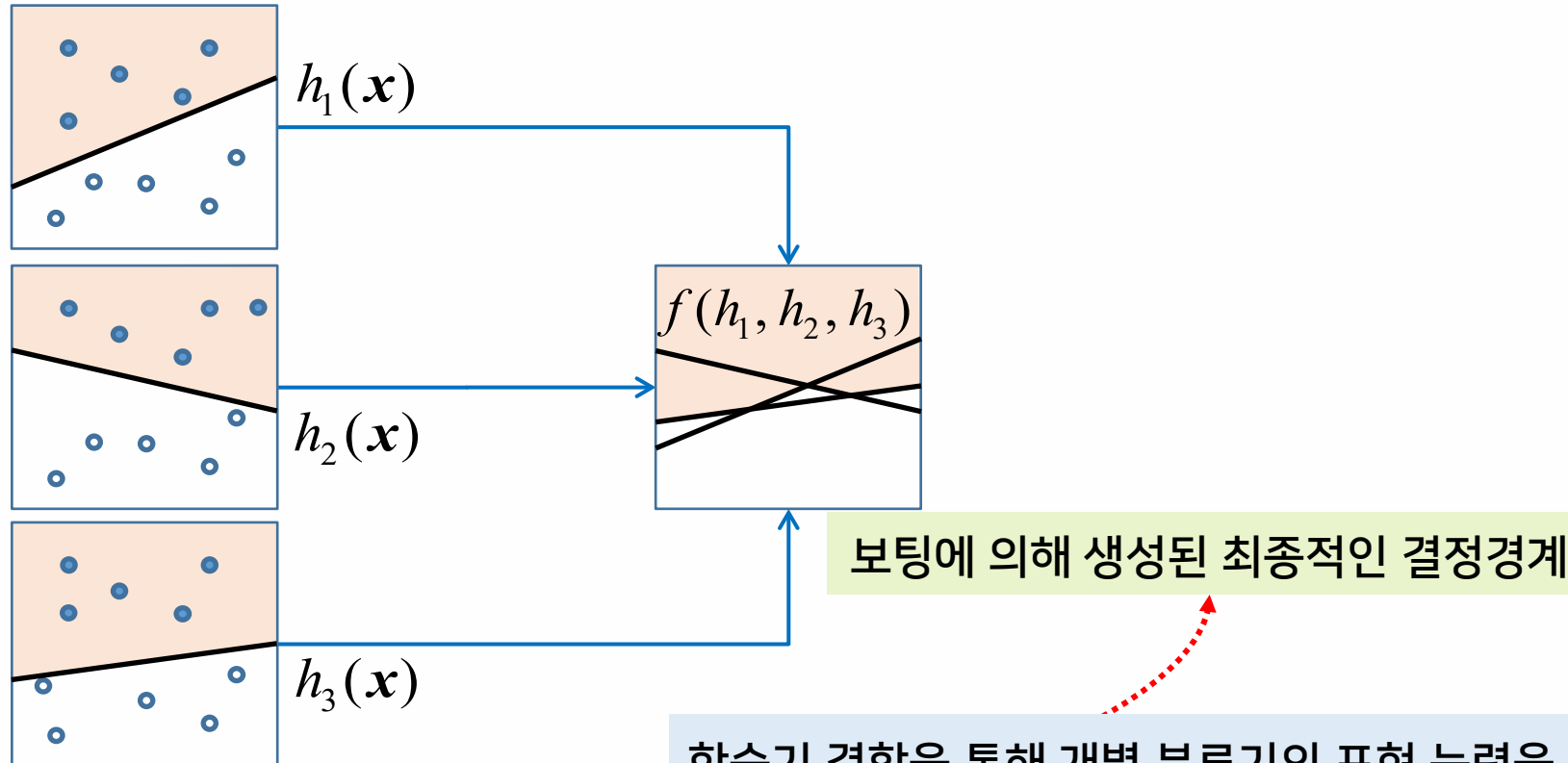
#### 보팅에 의한 결합함수

$$f(x) = f(h_1(x), h_2(x), \dots, h_M(x)) = \frac{1}{M} \sum_{i=1}^M h_i(x)$$

- 연속된 실수값을 내야 하는 함수 근사 문제에 적합
- 분류 문제의 경우, 결합 결과를 이용하여 최종 분류 결과를 결정해 주는 처리 과정이 필요

# 배깅과 보팅에 의한 결정경계

○ 이진 분류 문제,  $h_i(\mathbf{x}) = \text{sign}(\mathbf{w}_i^T \mathbf{x})$



보팅에 의해 생성된 최종적인 결정경계

학습기 결합을 통해 개별 분류기의 표현 능력을 증가하는 새로운 모델의 표현이 가능

배깅에 의해 생성된  
3개의 선형 분류기의 결정경계

# 배깅과 보팅에 의한 오차

각 학습기  $h_i(\mathbf{x})$ 에 대한 일반화 오차

$$E_x[e_i^2(\mathbf{x})] = E_x[\{h_i(\mathbf{x}) - f^*(\mathbf{x})\}^2]$$

$$f(\mathbf{x}) = \frac{1}{M} \sum_{i=1}^M h_i(\mathbf{x})$$

$$E_x[\{f(\mathbf{x}) - f^*(\mathbf{x})\}^2] = E_x\left[\left\{\frac{1}{M} \sum_{i=1}^M h_i(\mathbf{x}) - f^*(\mathbf{x})\right\}^2\right]$$

$$= E_x\left[\left\{\frac{1}{M} \sum_{i=1}^M e_i(\mathbf{x})\right\}^2\right]$$

$$= \frac{1}{M^2} \sum_{i=1}^M E_x[e_i^2(\mathbf{x})] + \frac{1}{M^2} \sum_{i,j} E_x[e_i(\mathbf{x})e_j(\mathbf{x})]$$

(각 학습기의 오차가 서로 독립적인 경우)

$$E_x[e_i(\mathbf{x})e_j(\mathbf{x})] = 0$$

$$E_x[\{f(\mathbf{x}) - f^*(\mathbf{x})\}^2] = \frac{1}{M} \left\{ \frac{1}{M} \sum_{i=1}^M E_x[e_i^2(\mathbf{x})] \right\}$$

3

## 부스팅

## 부스팅 boosting

- 간단한 학습기들이 상호보완적 역할을 할 수 있도록 **단계적으로 학습**을 수행하여 결합함으로써 성능을 증폭시키기 위한 방법
  - 먼저 학습된 학습기의 결과가 다음 학습기의 학습에 정보를 제공하여 이전 학습기의 결점을 보완





# AdaBoost 알고리즘

## ○ 특징

- ☐ 같은 데이터 집합을 반복해서 사용
- ☐ 학습할 때마다 각 데이터에 대한 가중치를 조정하여 학습의 변화를 꾀함
  - ✓ “데이터의 중요도가 적응적으로 변한다”  
→ Adaptive + Boost = AdaBoost
- ☐ 학습 방법 + 결합 방법
  - ✓ 분류기의 중요도 → 각 학습기의 결합 과정에서 결합계수로 사용

# AdaBoost에 의한 분류기의 학습과 분류

- ①  $N$ 개의 입출력 쌍으로 이루어진 학습 데이터 집합  $X = \{(x_j, t_j)\}_{j=1, \dots, N}$ 을 준비하고, 각 데이터에 대한 가중치  $w_j$  ( $j = 1, \dots, N$ )를  $w_j^{(1)} = 1/N$ 로 초기화함.  
 이때 목표 출력값은  $t_j \in \{-1, 1\}$  ( $j = 1, \dots, N$ )을 만족함(이진 분류 문제로 가정)

- ②  $i = 1, \dots, M$ 에 대해 다음과 같은 과정을 수행

- ②-1. 각 학습 데이터에 가중치가 적용된 오분류율을 다음과 같이 정의함

$$\varepsilon_i = \sum_{j=1}^N w_j^{(i)} I(h_i(x_j) \neq t_j)$$

이때  $I(h_i(x_j) \neq t_j)$ 는  $h_i(x_j) \neq t_j$ 이 만족할 때만 1의 값을 가지고, 그렇지 않은 경우는 0의 값을 가지는 함수임

# AdaBoost에 의한 분류기의 학습과 분류

②-2. 오분류율  $\varepsilon_i$ 를 최소화하는 분류기  $h_i(x)$ 를 학습을 통해 얻음

②-3. 오분류율  $\varepsilon_i$ 를 이용하여 각 분류기  $h_i(x)$ 의 중요도 값  $\alpha_i$ 를 계산함

$$\alpha_i = \frac{1}{2} \ln \left\{ \frac{1 - \varepsilon_i}{\varepsilon_i} \right\}$$

②-4. 각 데이터에 대한 가중치를 다음 식에 의해 수정

$$w_j^{(i+1)} = \frac{w_j^{(i)} \exp\{-\alpha_i t_j h_i(x_j)\}}{Z_i}$$

이때  $Z_i$ 는 가중치들의 합이 1이 되도록 정규화하기 위한 값으로, 다음과 같이 계산함

$$Z_i = \sum_{j=1}^N w_j^{(i)} \exp\{-\alpha_i t_j h_i(x_j)\}$$

## AdaBoost에 의한 분류기의 학습과 분류

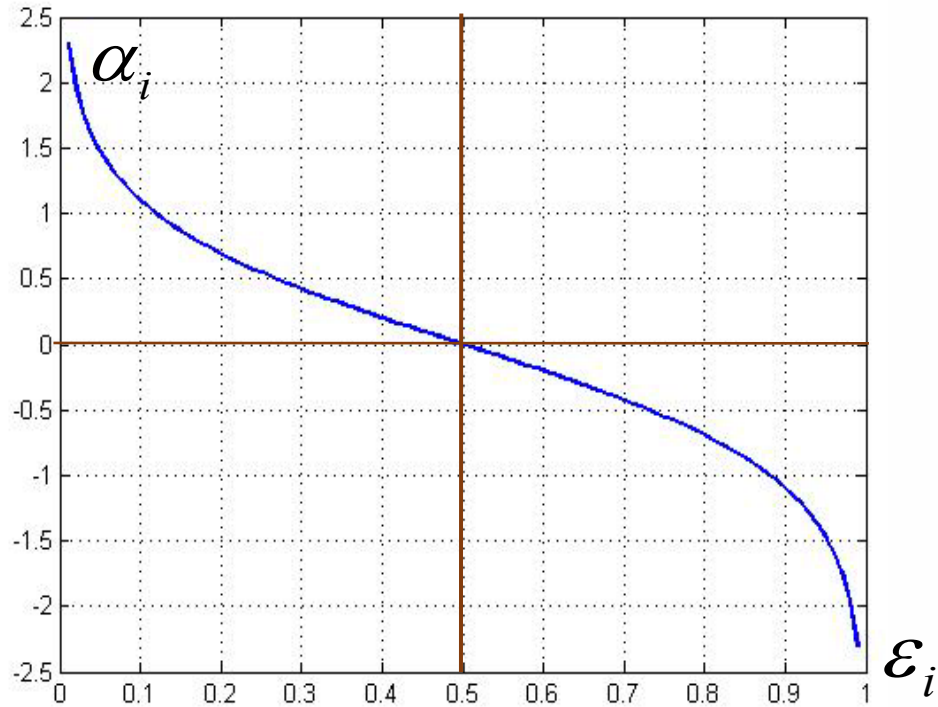
- ③  $M$ 개의 분류기가 모두 학습되면, 각 분류기의 중요도 값을 이용하여  $M$ 개의 분류기를 결합한 최종 판별함수를 만듦

$$f_M(\mathbf{x}) = \text{sign} \left( \sum_{i=1}^M \alpha_i h_i(\mathbf{x}) \right)$$

# AdaBoost 알고리즘

②-3. 오분류율에 의해 정의되는 각 분류기의 결합 중요도  $\alpha_i$

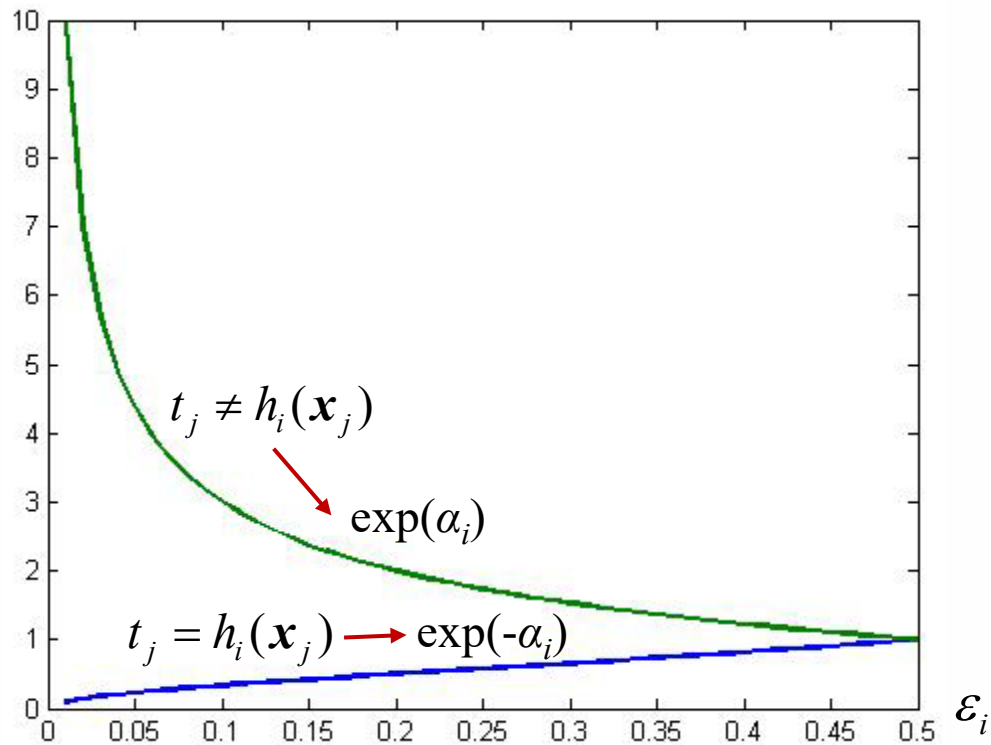
$$\alpha_i = \frac{1}{2} \ln \left\{ \frac{1 - \varepsilon_i}{\varepsilon_i} \right\}$$



# AdaBoost 알고리즘

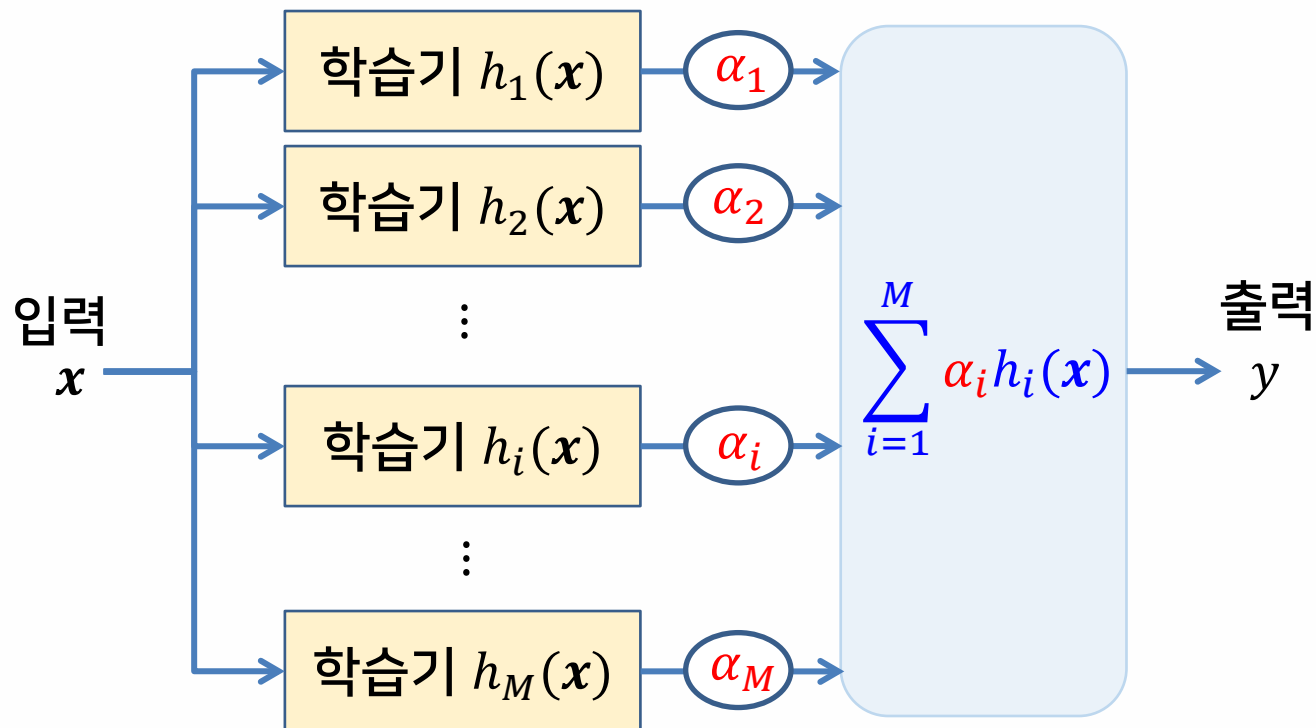
②-4. 가중치 수정에 사용되는 비례상수의 값  $\alpha_i$

$$w_j^{(i+1)} = \frac{w_j^{(i)} \exp\{-\alpha_i t_j h_i(x_j)\}}{Z_i}$$



# AdaBoost 알고리즘

- 단순한 보팅법에 가중치를 적용한 결합 방법





## AdaBoost 알고리즘?

- 이전 단계의 분류기의 학습 결과를 활용하여 다음 단계의 학습에 사용될 데이터에 가중치를 부여함으로써 분류기 간의 차별성 부여
- 각각의 간단한 분류기의 오분류율이 0.5보다 작은 조건만 만족하면 분류기의 결합을 통해 학습데이터에 대한 오차를 기하급수적으로 감소시킬 수 있음을 보임 (by Freund & Schapire)
- 최적화된 결합가중치를 찾아 분류기들을 결합
- 두 개의 클래스에 대한 분류 문제에 적합한 방법

4

## 결합 방법

## 기본적인 결합 방법

○ 대표적인 결합 방법 → 평균법, 보팅법(투표법)

○ 평균법

□ 학습기의 출력이 수치형일 때 적합

단순평균 
$$f(x) = \frac{1}{M} \sum_{i=1}^M h_i(x)$$

가중평균 
$$f(x) = \sum_{i=1}^M w_i h_i(x) \quad (w_i \geq 0, \sum_{i=1}^M w_i = 1)$$

# 기본적인 결합 방법

## ○ 보팅법

□ 분류 문제에서 주로 사용

**다수결 투표**  $f(x) = \operatorname{argmax}_j \sum_{i=1}^M h_i^j(x)$      $h_i^j(x) \rightarrow$  데이터  $x$ 에 대한 학습기  $h_i$ 의 클래스 레이블이  $j$ 임

**가중 보팅**  $f(x) = \operatorname{argmax}_j \sum_{i=1}^M w_i h_i^j(x)$  ( $w_i \geq 0, \sum_{i=1}^M w_i = 1$ )

□  $h_i^j(x)$ 의 출력 유형에 따른 분류

✓ 하드 보팅  $\rightarrow h_i^j(x) \in \{0, 1\}$

✓ 소프트 보팅  $\rightarrow h_i^j(x) \in [0, 1]$

## 기본적인 결합 방법

### ○ 기본 학습기의 결과를 결합하는 학습기("결합기") 사용

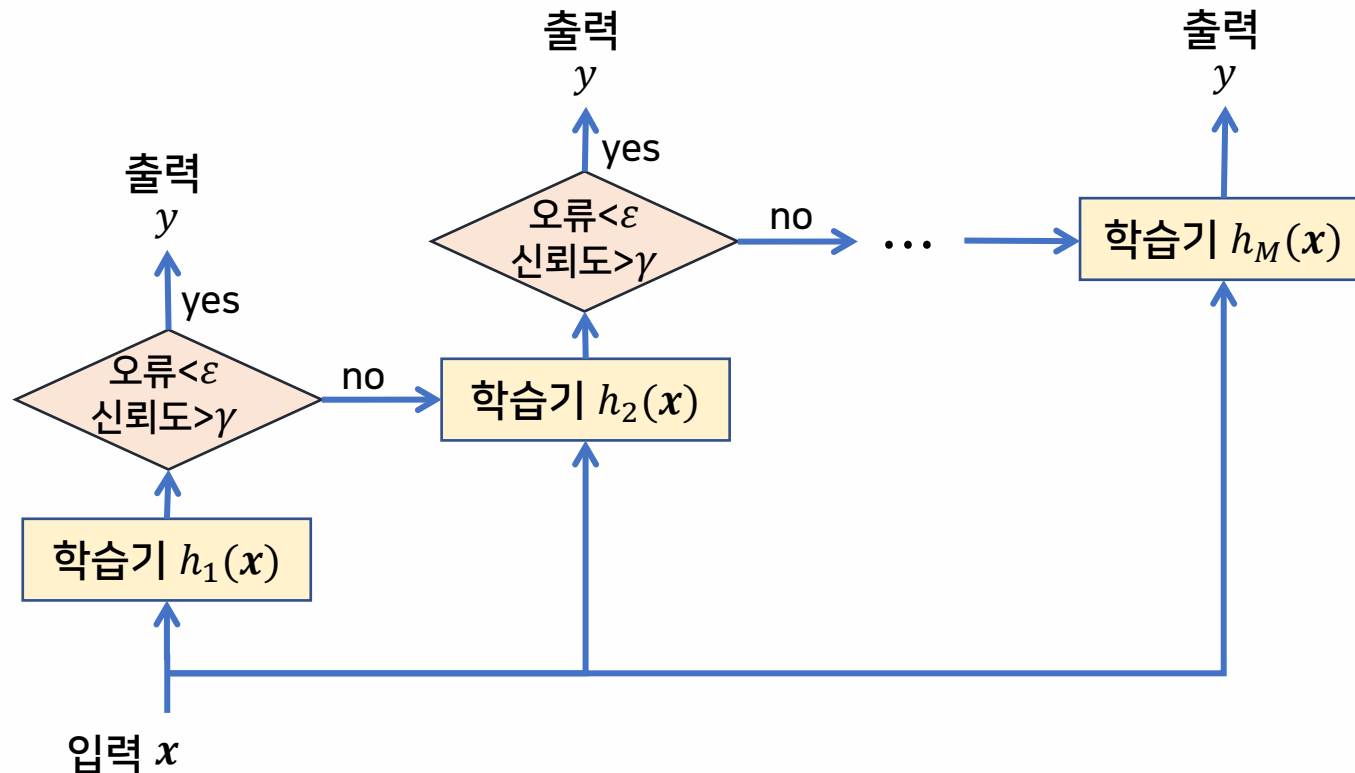
#### □ 결합기를 위한 학습 데이터 구성 방법

- ✓ 기본 학습기의 학습에 사용되지 않는 새로운 학습 데이터 집합을 준비
- ✓ 새로운 학습 데이터  $(x_i, y_i)$ 에 대해  
기본 학습기  $h_j$ 의 출력값을  $z_{ij}$ 라고 하면  
→ 결합기의 학습 데이터  $\rightarrow ((z_{i1}, z_{i2}, \dots, z_{iM}), y_i)$

# 캐스케이딩 cascading

○ 여러 복잡도를 가진 학습기들의 순차적인 결합에 중점을 둔 방법

□ 계산 시간을 줄이면서 성능도 보장

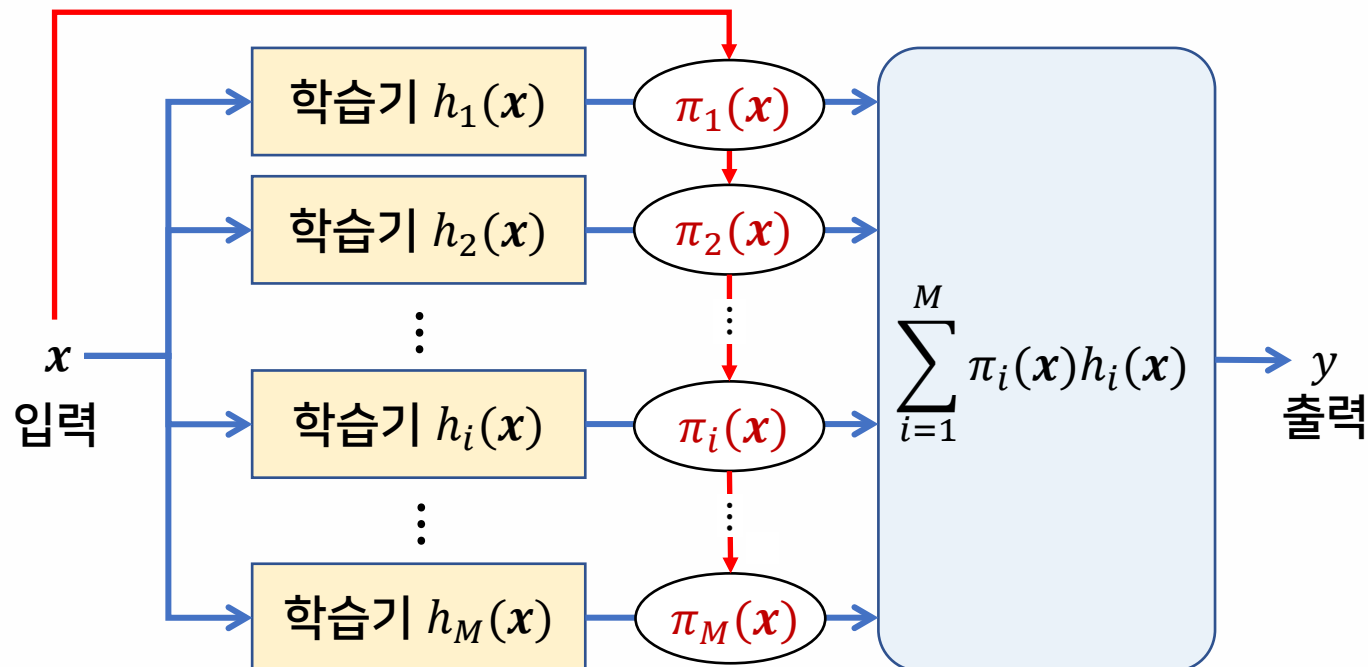


# 전문가 혼합 mixture of experts

○ 복수 개의 학습기를 가중합하여 최종 학습기를 만드는 결합 방법

□  $\pi_i(\mathbf{x}) \rightarrow$  주어진 입력에 따라 어떤 학습기를 중요하게 사용할지를 결정

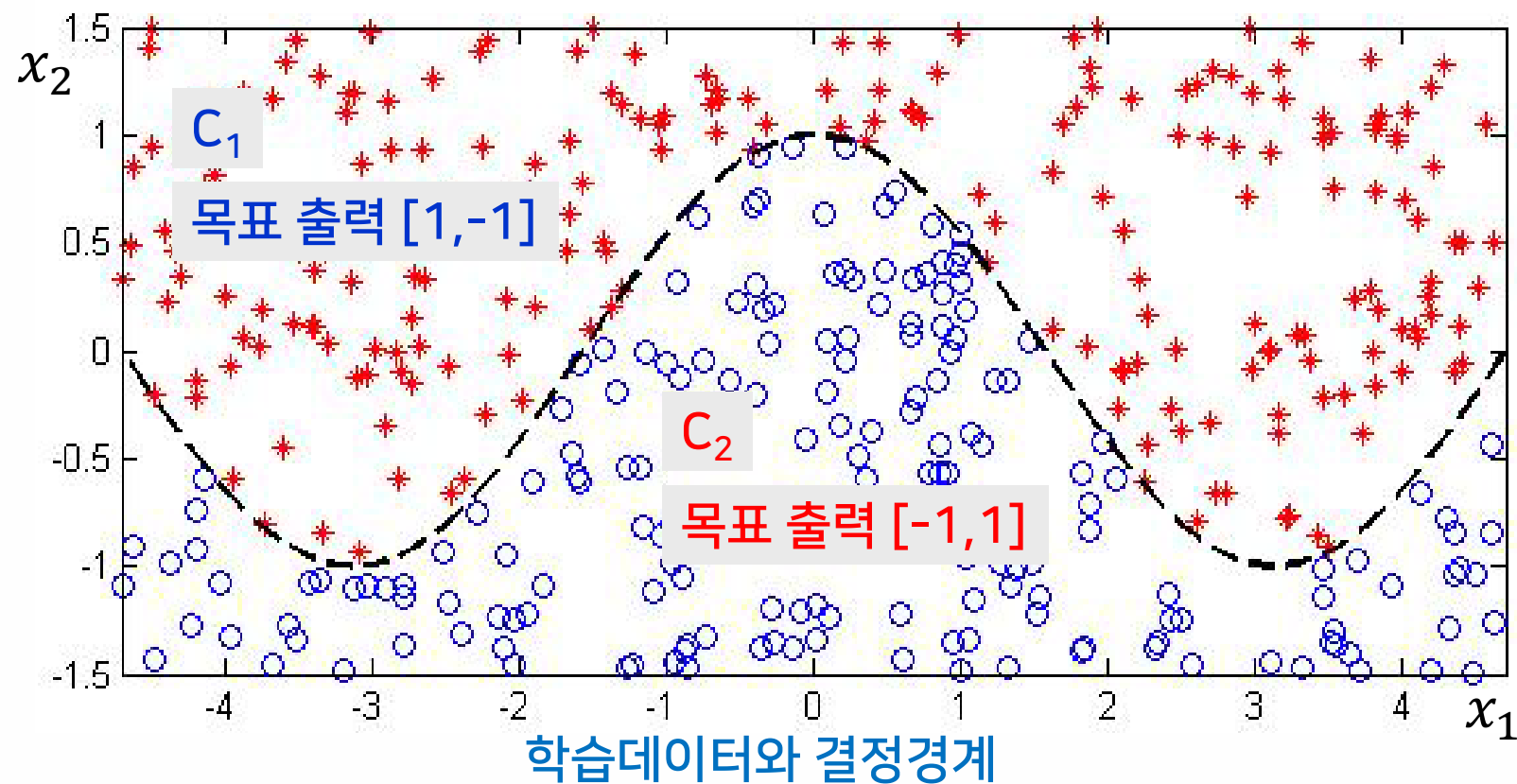
$$f(\mathbf{x}) = f(h_1(\mathbf{x}), h_2(\mathbf{x}), \dots, h_M(\mathbf{x})) = \sum_{i=1}^M \pi_i(\mathbf{x}) h_i(\mathbf{x})$$



# 앙상블 학습의 결과 예시

AdaBoost 방법을 이용한 퍼셉트론 분류기의 결합

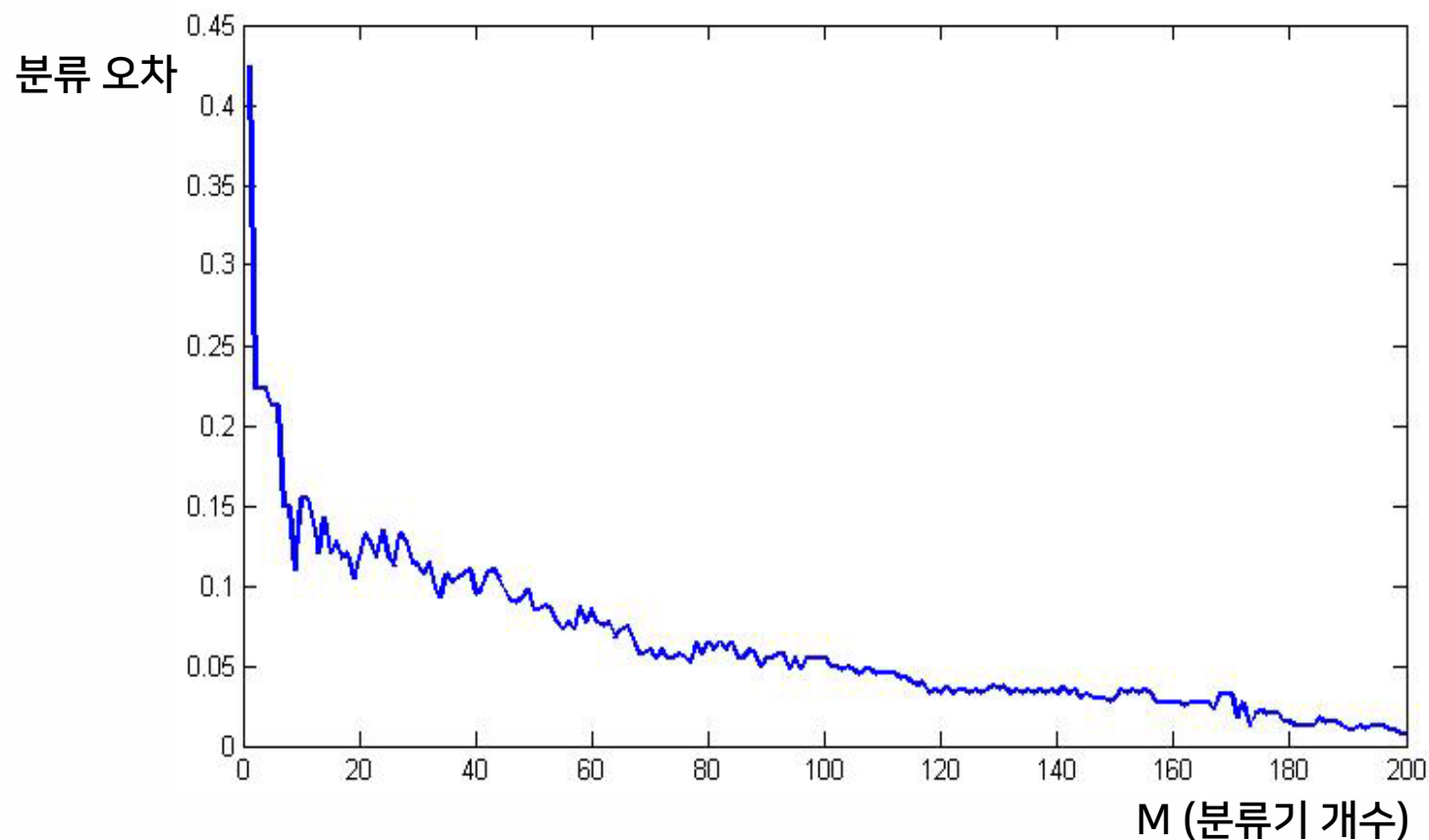
$\{(x_1, x_2) | x_1 \in [-1.5\pi, 1.5\pi], x_2 \in [-1.5, 1.5]\} \rightarrow 400\text{개}$





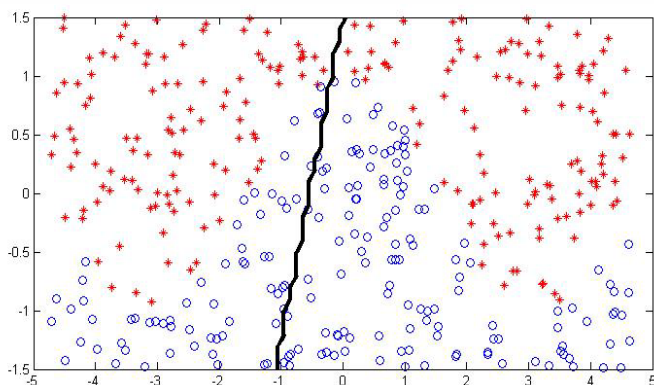
# 앙상블 학습의 결과 예시

결합하는 분류기의 개수 변화에 따른 분류 오차의 변화

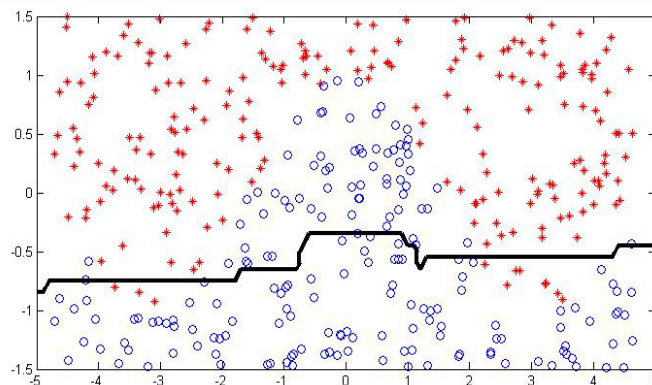


# 앙상블 학습의 결과 예시

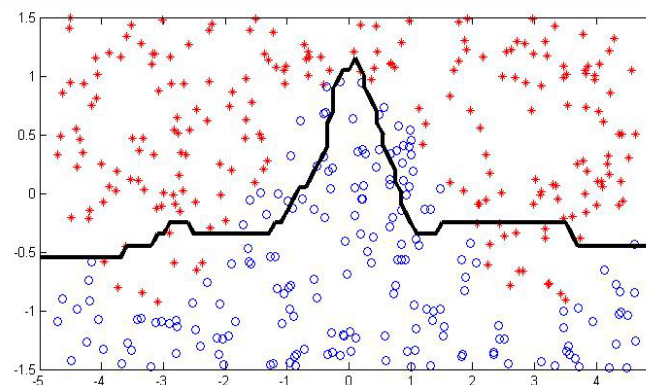
결합하는 분류기의 개수 변화에 따른 결정경계의 변화



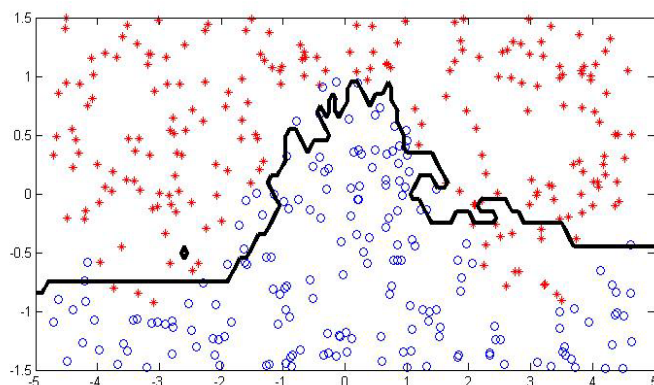
M=1



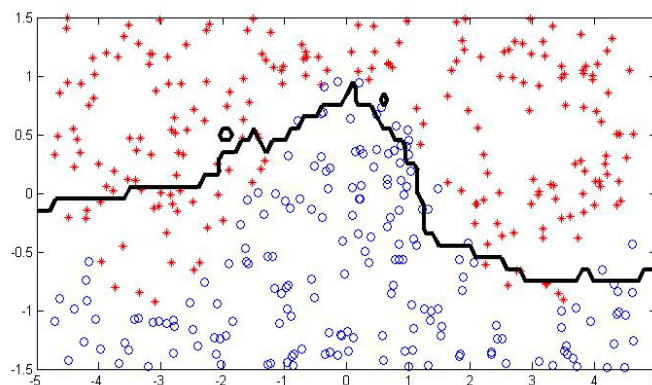
M=5



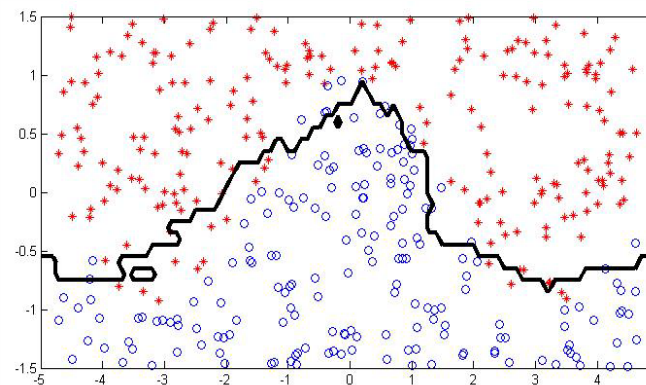
M=10



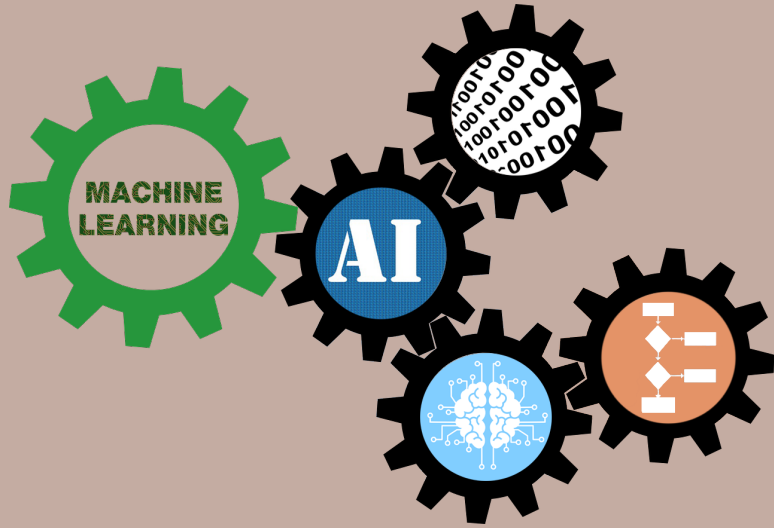
M=50



M=100



M=200



다음시간안내

## 제7강

# 결정 트리와 랜덤 포레스트