

11

강

인공지능

머신러닝(2)

컴퓨터과학과 이병래교수

학습목차

1 선형회귀

2 로지스틱 회귀

3 군집화





선형회귀

1. 선형회귀의 개념

회귀분석(regression analysis)이란?

- 독립변수와 종속변수 사이의 상관관계를 추정하는 기법
 - 독립변수: 입력
 - 종속변수: 독립변수에 따른 출력

예 어떤 학생이 공부한 시간으로 그 학생의 성적을 어떻게 예측할 수 있는가?

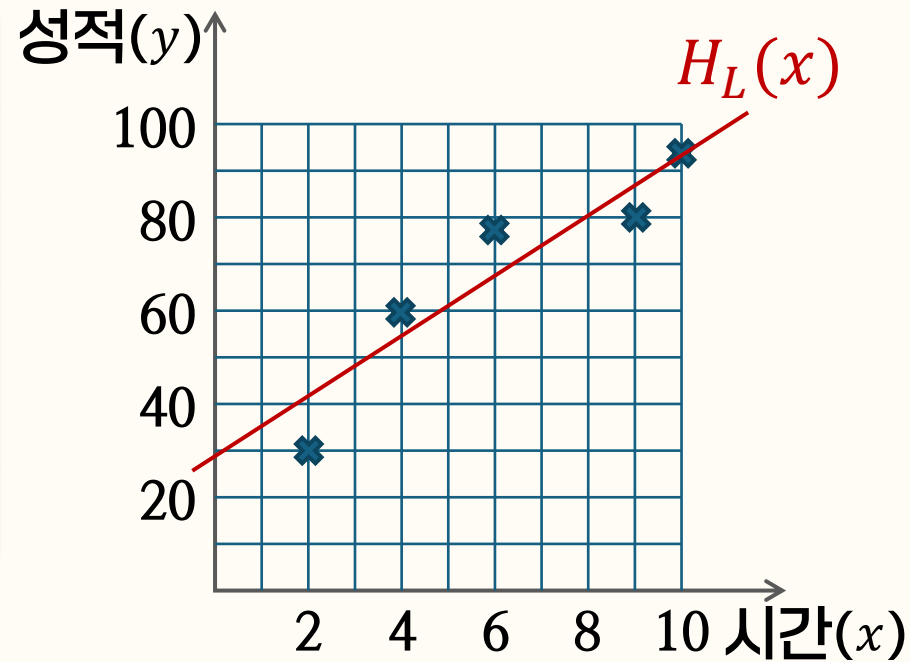
예 어떤 학생이 공부한 시간으로 그 학생의 시험 합격 여부를 어떻게 예측할 수 있는가?

1. 선형회귀의 개념

■ 선형회귀(linear regression)란?

- 독립변수와 종속변수 사이의 상관관계를 선형함수로 모델링하는 것

공부한 시간(x)	시험 성적(y)
10	93
9	80
6	77
4	60
2	30



→ $y = H_L(x)$

→ $H_L(x) = w_0 + w_1x$

1. 선형회귀의 개념



■ 선형회귀의 목표

- d 개의 독립변수 x_1, x_2, \dots, x_d 와 종속변수 y 에 대한 학습 표본 집합으로부터 선형가설 $H_L(x_1, x_2, \dots, x_d)$ 를 구하는 것

$$H_L(x_1, x_2, \dots, x_d) = w_0 + w_1x_1 + w_2x_2 + \dots + w_dx_d$$

$$\text{또는 } H_L(\mathbf{x}) = \mathbf{w}^T \mathbf{x}, \quad \mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}$$

 $d = 1$  단순 선형회귀(simple linear regression)

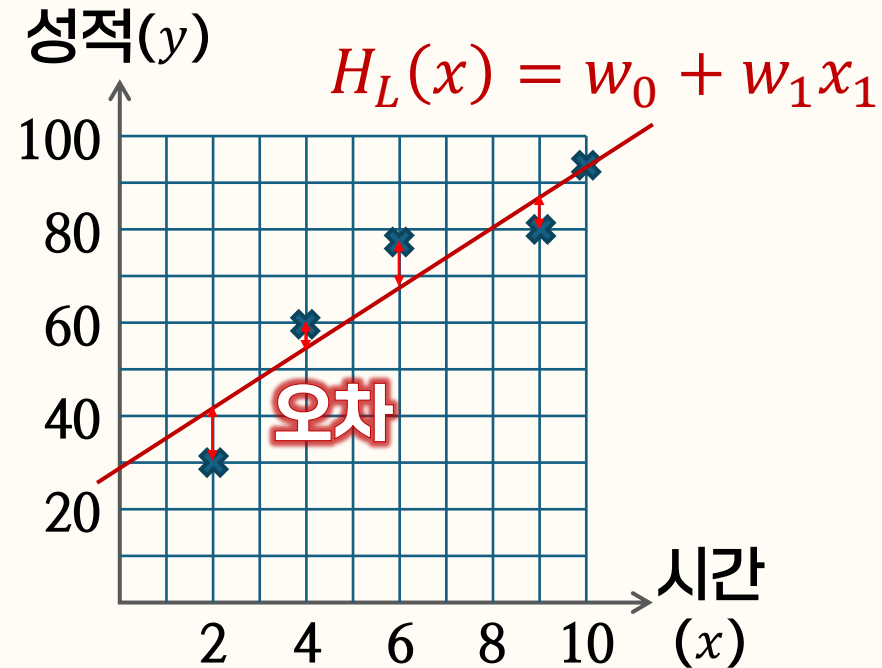
 $d \geq 2$  다중 선형회귀(multiple linear regression)

2. 선형가설의 학습

■ 선형가설과 비용함수

• 학습 표본

i	공부한 시간($x^{(i)}$)	시험 성적($y^{(i)}$)
1	10	93
2	9	80
3	6	77
4	4	60
5	2	30



📌 목표 : 오차에 따른 비용함수가 최소가 되는 \hat{w}_0 와 \hat{w}_1 을 구하는 것

$$(\hat{w}_0, \hat{w}_1) = \underset{(w_0, w_1)}{\operatorname{argmin}} C(w_0, w_1)$$

2. 선형가설의 학습

■ 선형가설과 비용함수

- 선형가설

$$H_L(x) = w_0 + w_1x$$

- 비용함수: 평균제곱오차(mean squared error: MSE)

$$\begin{aligned} C_{MSE}(w_0, w_1) &= \frac{1}{m} \sum_{i=1}^m \{H_L(x^{(i)}) - y^{(i)}\}^2 \\ &= \frac{1}{m} \sum_{i=1}^m (w_0 + w_1x^{(i)} - y^{(i)})^2 \end{aligned}$$



$x^{(i)}, y^{(i)}$: i 번째 표본

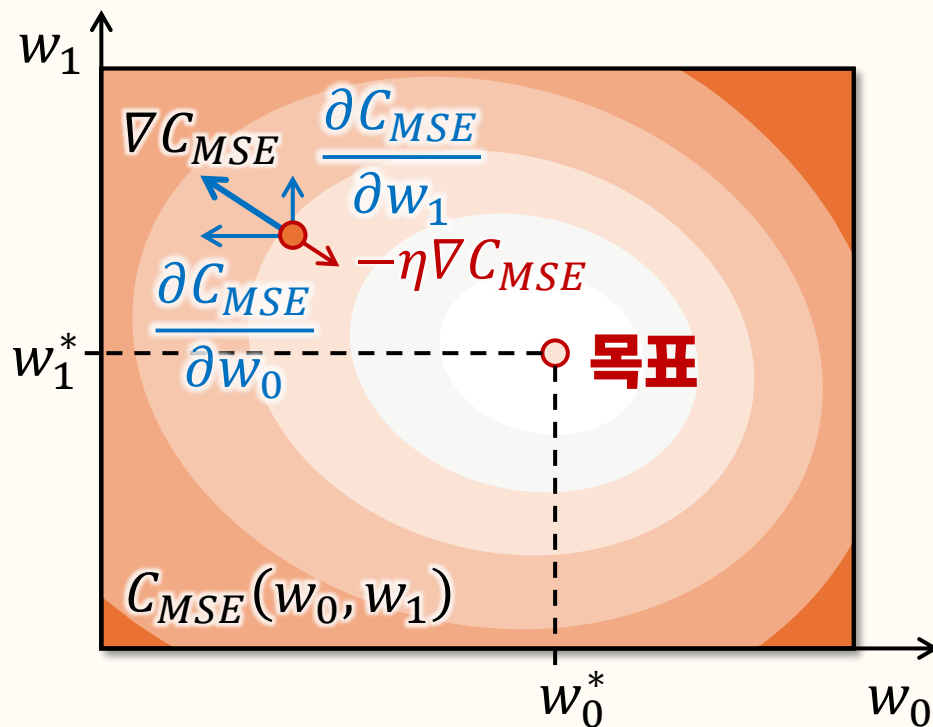


m : 표본의 수

2. 선형가설의 학습

■ 비용함수의 최소화

- 경사하강법(gradient descent method): w_0 와 w_1 을 임의의 값으로 초기화한 후 **비용함수 기울기의 음의 방향**으로 경사면을 따라 내려가도록 조금씩 변화시키는 것을 반복



2. 선형가설의 학습

■ 비용함수의 최소화

- 경사하강법(gradient descent method): w_0 와 w_1 을 임의의 값으로 초기화한 후 **비용함수 기울기의 음의 방향**으로 경사면을 따라 내려가도록 조금씩 변화시키는 것을 반복

$$\nabla C_{MSE}(w_0, w_1) = \left(\frac{\partial}{\partial w_0} C_{MSE}(w_0, w_1), \frac{\partial}{\partial w_1} C_{MSE}(w_0, w_1) \right)$$

$$\begin{aligned} \frac{\partial}{\partial w_0} C_{MSE}(w_0, w_1) &= \frac{\partial}{\partial w_0} \frac{1}{m} \sum_{i=1}^m (w_0 + w_1 x^{(i)} - y^{(i)})^2 \\ &= \frac{2}{m} \sum_{i=1}^m (w_0 + w_1 x^{(i)} - y^{(i)}) \end{aligned}$$

2. 선형가설의 학습

■ 비용함수의 최소화

- 경사하강법(gradient descent method): w_0 와 w_1 을 임의의 값으로 초기화한 후 **비용함수 기울기의 음의 방향**으로 경사면을 따라 내려가도록 조금씩 변화시키는 것을 반복

$$\nabla C_{MSE}(w_0, w_1) = \left(\frac{\partial}{\partial w_0} C_{MSE}(w_0, w_1), \frac{\partial}{\partial w_1} C_{MSE}(w_0, w_1) \right)$$

$$\begin{aligned} \frac{\partial}{\partial w_1} C_{MSE}(w_0, w_1) &= \frac{\partial}{\partial w_1} \frac{1}{m} \sum_{i=1}^m (w_0 + w_1 x^{(i)} - y^{(i)})^2 \\ &= \frac{2}{m} \sum_{i=1}^m (w_0 + w_1 x^{(i)} - y^{(i)}) x^{(i)} \end{aligned}$$

2. 선형가설의 학습

■ 비용함수의 최소화

- 경사하강법(gradient descent method): w_0 와 w_1 을 임의의 값으로 초기화한 후 **비용함수 기울기의 음의 방향**으로 경사면을 따라 내려가도록 조금씩 변화시키는 것을 반복

k 번째 학습 결과의 업데이트

$$w_0(k+1) = w_0(k) - \eta \frac{2}{m} \sum_{i=1}^m [\{w_0(k) + w_1(k)x^{(i)}\} - y^{(i)}]$$

$$w_1(k+1) = w_1(k) - \eta \frac{2}{m} \sum_{i=1}^m [\{w_0(k) + w_1(k)x^{(i)}\} - y^{(i)}]x^{(i)}$$

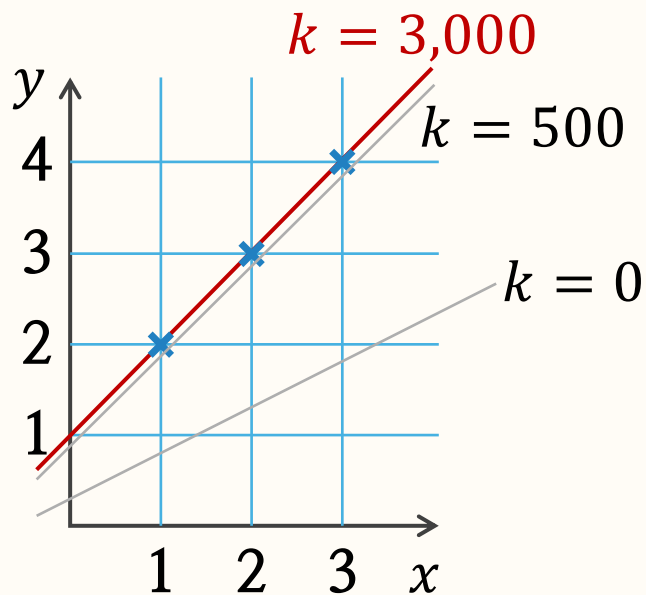


η : 학습률

2. 선형가설의 학습

■ 선형회귀 적용 예

i	$x^{(i)}$	$y^{(i)}$
1	1	2
2	2	3
3	3	4



반복횟수(k)	$w_0(k)$	$w_1(k)$	비용함수
0(초깃값)	0.3	0.5	3.0567
1	0.3340	0.5747	2.4209
2	0.3643	0.6410	1.9183
500	0.8793	1.0531	0.0021
1,000	0.9638	1.0159	0.000189
2,000	0.9967	1.0014	0.000002
3,000	0.9997	1.0001	0.000000

⇒ $H_L(x) = 0.9997 + 1.0001x$

3. 다중 선형회귀

❑ 다중(multiple) 선형회귀란?

- 독립변수가 2개 이상인 경우의 선형회귀

독립변수: x_1, x_2, \dots, x_d \rightarrow 종속변수: y

$$H_L(x_1, x_2, \dots, x_d) = w_0 + w_1x_1 + w_2x_2 + \dots w_dx_d$$

$$H_L(\mathbf{x}) = \mathbf{w}^T \mathbf{x}, \quad \mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}$$

3. 다중 선형회귀

■ 비용함수의 최소화

- 비용함수: 평균제곱오차(MSE)

$$C_{MSE}(\mathbf{w}) = \frac{1}{m} \sum_{i=1}^m \{H_L(\mathbf{x}^{(i)}) - y^{(i)}\}^2$$

- 비용함수를 최소화하는 H_L 의 학습(경사하강법 활용)

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \eta \nabla C_{MSE}(\mathbf{w}(k))$$

η : 학습률

$$\nabla C_{MSE}(\mathbf{w}(k)) = \frac{2}{m} \sum_{i=1}^m \{H_L(\mathbf{x}^{(i)}) - y^{(i)}\} \mathbf{x}^{(i)}$$



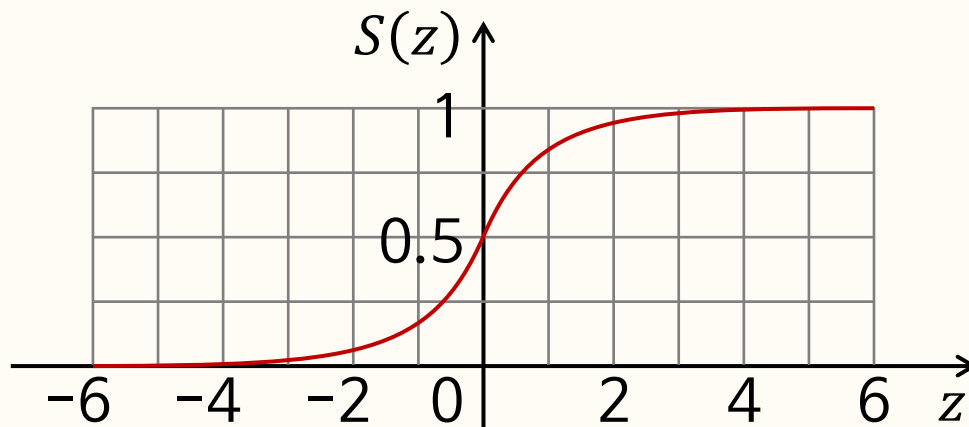
로지스틱 회귀

1. 로지스틱 회귀의 개념

❑ 로지스틱 회귀(logistic regression)란?

- 독립변수의 값에 대해 종속변수가 0 또는 1이라는 값을 낼 수 있는 가설을 구하는 것
- 📝 종속변수가 독립변수에 따른 유형을 구분하는 값일 경우
- 로지스틱 함수(logistic function) $S(z)$

$$S(z) = \frac{1}{1 + e^{-z}}$$



z : 로짓(logit)

1. 로지스틱 회귀의 개념

❑ 로지스틱 회귀의 가설 $H_S(\mathbf{x})$

- 독립변수의 값에 대해 종속변수의 값이 1일 확률을 계산
→ 선형함수 $H_L(\mathbf{x})$ 를 로지스틱 함수의 로짓으로 사용하여 0과 1 사이의 값을 구함

$$\begin{aligned} H_S(\mathbf{x}) &= S(H_L(\mathbf{x})) \\ &= \frac{1}{1 + e^{-H_L(\mathbf{x})}} \\ &= \frac{1}{1 + e^{-\mathbf{w}^T \mathbf{x}}} \end{aligned}$$

2. 가설의 학습

■ 비용함수의 정의

- 교차 엔트로피(cross entropy)

$$C_{CE}(\mathbf{w}) = \frac{1}{m} \sum_{i=1}^m [-y^{(i)} \ln H_S(\mathbf{x}^{(i)}) - (1 - y^{(i)}) \ln \{1 - H_S(\mathbf{x}^{(i)})\}]$$

 이진 교차 엔트로피(binary cross entropy)

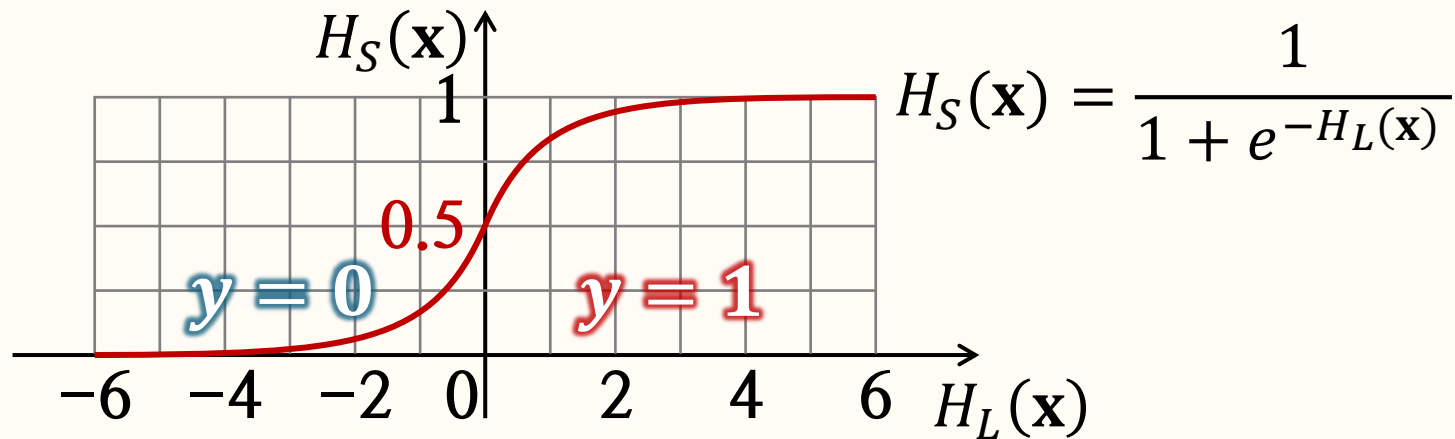
■ 경사하강법에 의한 학습

$$\nabla C_{CE}(\mathbf{w}) = \frac{1}{m} \sum_{i=1}^m \{H_S(\mathbf{x}^{(i)}) - y^{(i)}\} \mathbf{x}^{(i)}$$

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \eta \nabla C_{CE}(\mathbf{w}(k))$$

3. 로지스틱 회귀에 의한 분류(classification)

가설 $H_S(\mathbf{x})$ 와 결정경계

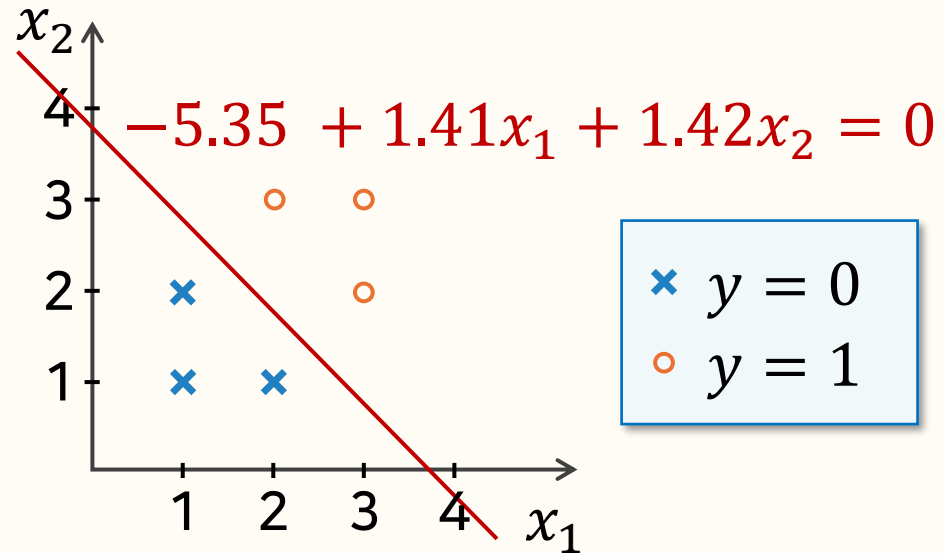


→ 결정경계: $H_L(\mathbf{x}) = 0$

4. 로지스틱 회귀의 적용 예

■ 학습 표본

i	$\mathbf{x}^{(i)}$	$y^{(i)}$
1	(1, 1)	0
2	(2, 1)	0
3	(1, 2)	0
4	(2, 3)	1
5	(3, 2)	1
6	(3, 3)	1



학습 조건: $\mathbf{w}(0)^T = [0.5 \quad -0.5 \quad 1.0]$, $\eta = 0.01$

→ $\mathbf{w}(1)^T = [0.497 \quad -0.503 \quad 0.997]$

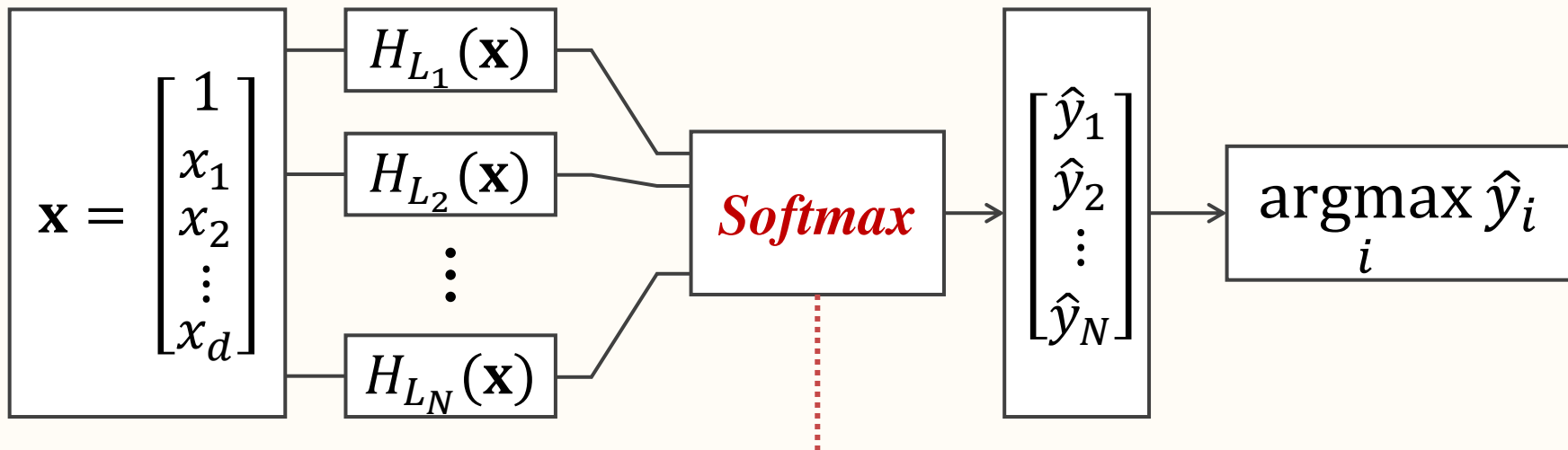
$\mathbf{w}(1000)^T = [-0.878 \quad -0.013 \quad 0.704]$

$\mathbf{w}(8000)^T = [-5.35 \quad 1.41 \quad 1.42]$

5. 다항 로지스틱 회귀

다항(multinomial) 로지스틱 회귀란?

- 식별하려는 클래스가 3개 이상인 경우 대한 로지스틱 회귀



$$\hat{y}_i = \frac{e^{H_{L_i}(x)}}{\sum_{j=1}^N e^{H_{L_j}(x)}}, i = 1, 2, \dots, N$$

5. 다항 로지스틱 회귀

■ 다항 로지스틱 회귀의 학습

- 비용함수: 교차 엔트로피

$$C_{CE}(\mathbf{w}) = \frac{1}{m} \sum_{t=1}^m \sum_{i=1}^N -y_i^{(t)} \ln \hat{y}_i^{(t)}$$

- 경사하강법에 의한 \mathbf{w} 의 학습

$$w_{i0}(k+1) = w_{i0}(k) - \eta \frac{1}{m} \sum_{t=1}^m \left(\hat{y}_i^{(t)} - y^{(t)} \right)$$

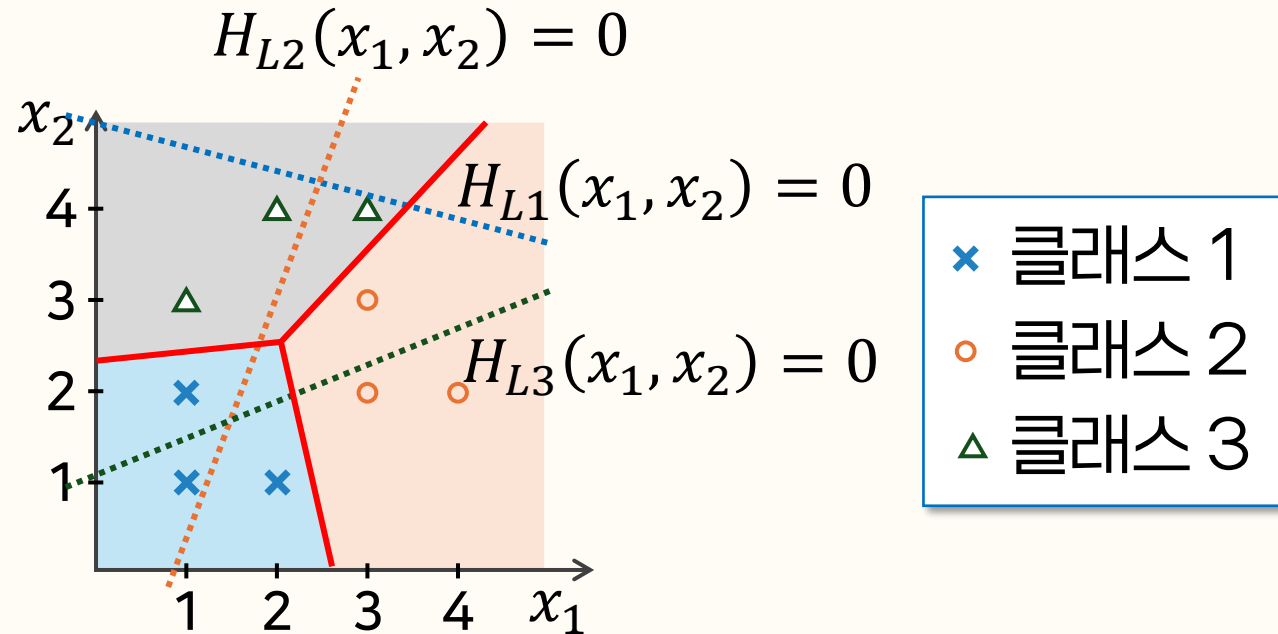
$$w_{ij}(k+1) = w_{ij}(k) - \eta \frac{1}{m} \sum_{t=1}^m \left(\hat{y}_i^{(t)} - y^{(t)} \right) x_j^{(t)}, \quad j = 1, 2, \dots, d$$

5. 다항 로지스틱 회귀

다항 로지스틱 회귀 적용 예

• 학습 표본

i	$\mathbf{x}^{(i)}$	$\mathbf{y}^{(i)}$
1	(1, 1)	(1, 0, 0)
2	(2, 1)	(1, 0, 0)
3	(1, 2)	(1, 0, 0)
4	(3, 3)	(0, 1, 0)
5	(3, 2)	(0, 1, 0)
6	(4, 2)	(0, 1, 0)
7	(1, 3)	(0, 0, 1)
8	(2, 4)	(0, 0, 1)
9	(3, 4)	(0, 0, 1)



$$H_{L1}(x_1, x_2) = 18.54 - 1.02x_1 - 3.74x_2$$

$$H_{L2}(x_1, x_2) = -6.93 + 8.39x_1 - 2.31x_2$$

$$H_{L3}(x_1, x_2) = -9.96 - 3.88x_1 + 9.05x_2$$

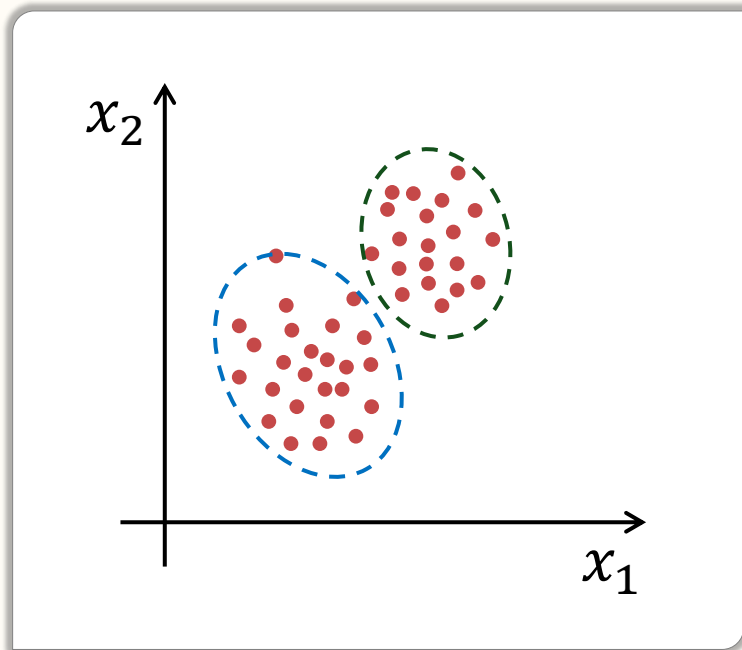


군집화

1. 군집화의 개념

■ 군집화(clustering)란?

- 패턴 집합이 주어졌을 때, 같은 종류라고 생각할 수 있는 몇 개의 서브클래스로 분할하는 것
- 비지도학습 방식으로 학습함



2. 군집화 알고리즘의 예

■ k -평균 군집화(k -means clustering)

- 학습표본 집합을 대표하는 k 개의 평균벡터를 구함
- 임의의 값으로 초기화한 평균벡터를 반복적으로 수정하여 적절한 위치로 수렴하도록 하는 방법을 사용
- 알고리즘의 기본 골격
 - 각각의 학습표본을 가장 가까운 평균벡터가 무엇인가에 따라 k 개의 군집으로 분류
 - 각각의 평균벡터에 해당되는 군집으로 분류된 표본들의 평균을 구하여 그 평균벡터를 수정

2. 군집화 알고리즘의 예

k -평균 군집화 알고리즘

```
1 function kMeansClustering(x, k)
2   // 매개변수 x[1..N]: 학습표본 데이터 집합, k: 평균벡터의 수
3   // m[1..k]: k개의 평균벡터
4   // mSum[1..k]: mSum[j]는 가장 가까운 평균벡터가 m[j]인
   학습표본의 합
5   // c[1..k]: c[j]는 가장 가까운 평균벡터가 m[j]인 학습표본의 수
6   m[1..k]의 초기위치 선택;
7   repeat
8     mOld = m; // 기존 평균벡터 보관
9     mSum과 c의 모든 값을 0으로 리셋;
10    // x[1..N]을 k개의 군집에 분류
11    for i=1 to N do
12      학습표본 x[i]에 대해 가장 가까운 평균벡터 m[j]를 선택;
13      .....
```

2. 군집화 알고리즘의 예

k -평균 군집화 알고리즘

```
1 function kMeansClustering(x, k)
...     .....
6     m[1..k]의 초기위치 선택;
7     repeat
8         mOld = m; // 기존 평균벡터 보관
9         mSum과 c의 모든 값을 0으로 리셋;
10        // x[1..N]을 k개의 군집에 분류
11        for i=1 to N do
12            학습표본 x[i]에 대해 가장 가까운 평균벡터 m[j]를 선택;
13            mSum[j] += x[i]; // x[i]를 군집 j로 분류하여 mSum[j]에 합산
14            ++c[j];         // 군집 j에 분류된 표본수 증가
15        end-for;
16        // m[1..k]를 새로운 군집의 평균으로 조정
...     .....

```

2. 군집화 알고리즘의 예

k -평균 군집화 알고리즘

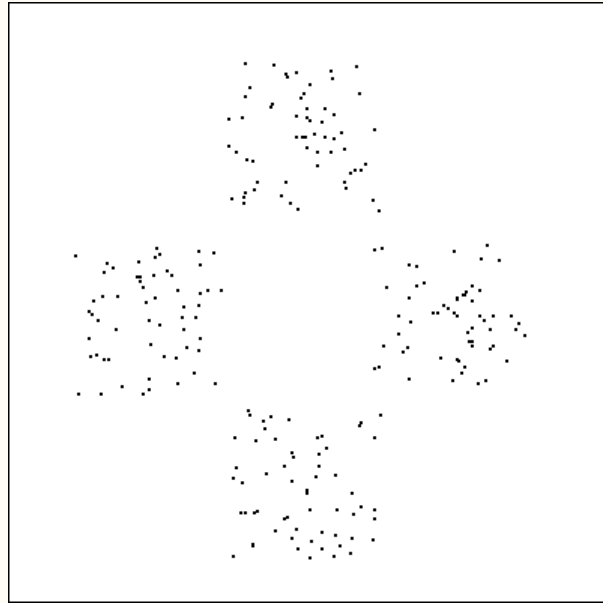
```
1 function kMeansClustering(x, k)
...     .....
7     repeat
...         .....
11        for i=1 to N do
...            .....
15        end-for;
16        // m[1..k]를 새로운 군집의 평균으로 조정
17        for j=1 to k do
18            if (c[j] != 0) m[j] = mSum[j] / c[j];
19        end-for;
20    until m == mOld;    // m에 변화가 없으면 마침
21    return m;
22 end-function;
```

2. 군집화 알고리즘의 예

■ k -평균 군집화(k -means clustering)

- 군집화 실행 예

학습 데이터

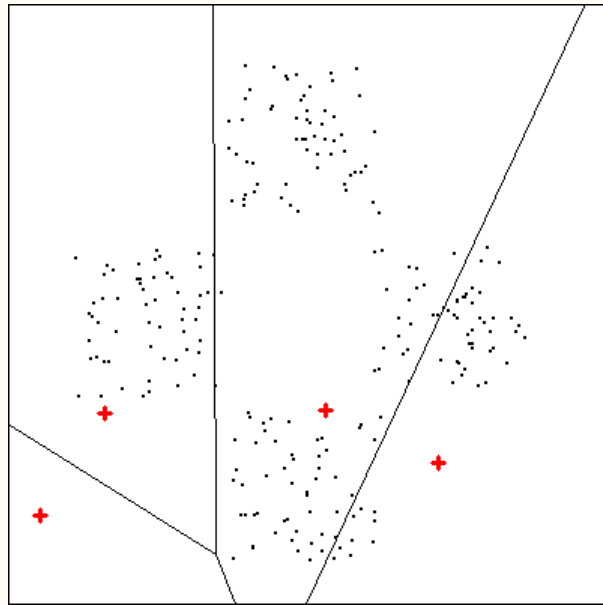


2. 군집화 알고리즘의 예

■ k -평균 군집화(k -means clustering)

- 군집화 실행 예

초기 평균벡터

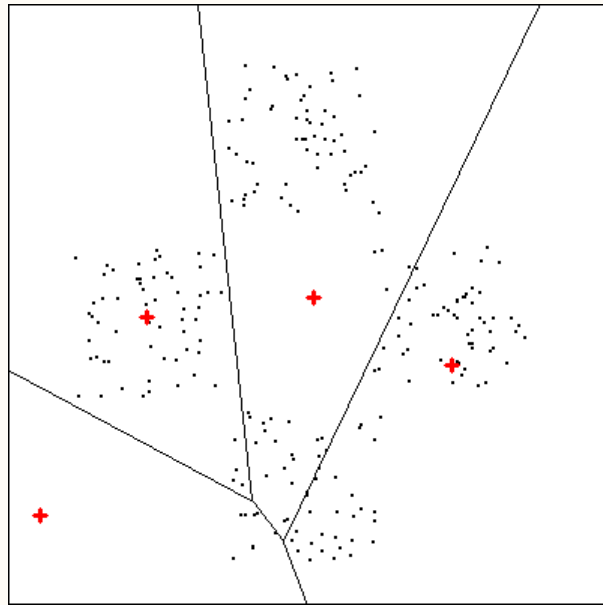


2. 군집화 알고리즘의 예

■ k -평균 군집화(k -means clustering)

- 군집화 실행 예

1회 반복

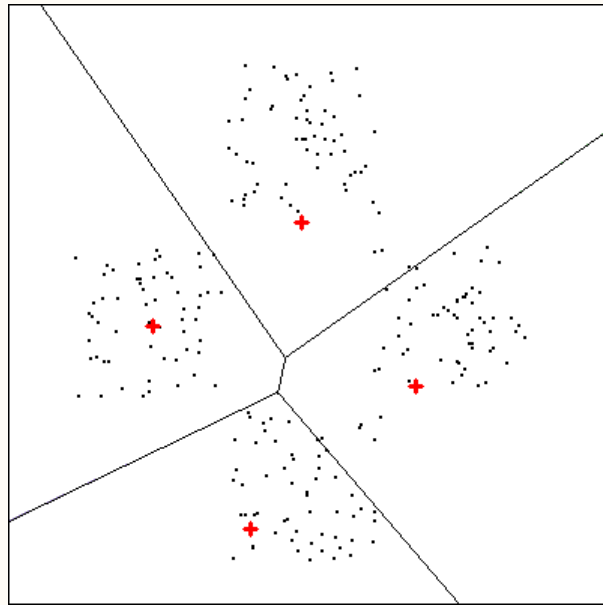


2. 군집화 알고리즘의 예

■ k -평균 군집화(k -means clustering)

- 군집화 실행 예

2회 반복

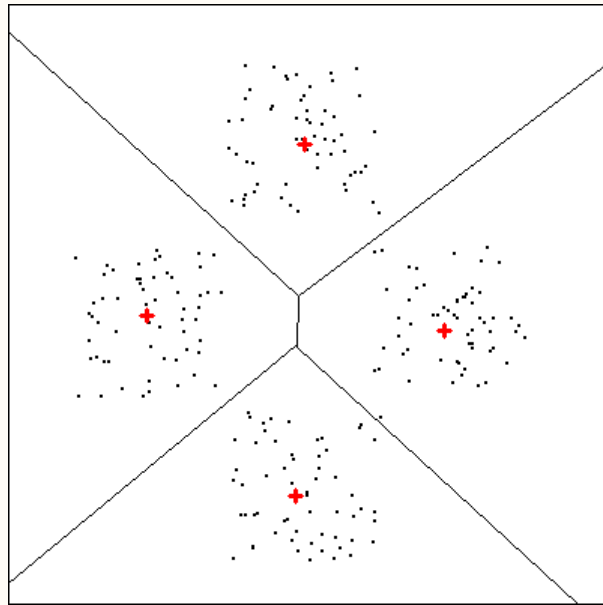


2. 군집화 알고리즘의 예

■ k -평균 군집화(k -means clustering)

- 군집화 실행 예

3회 반복

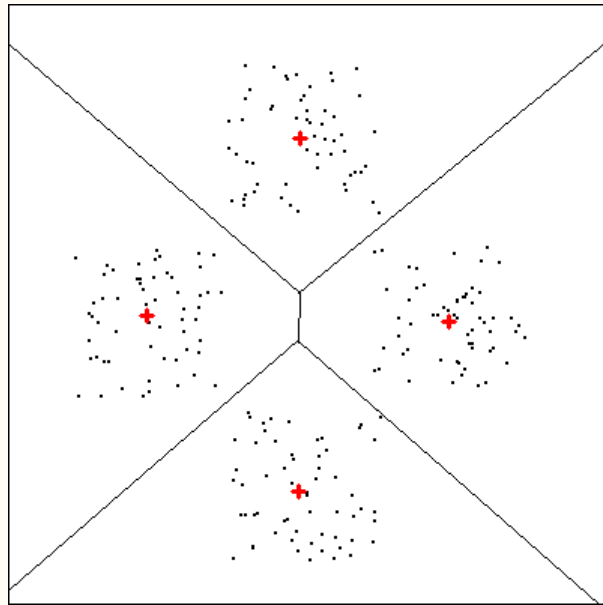


2. 군집화 알고리즘의 예

■ k -평균 군집화(k -means clustering)

- 군집화 실행 예

4회 반복



2. 군집화 알고리즘의 예

■ k -평균 군집화(k -means clustering)

- k -평균 군집화의 특성

- 평균벡터의 수를 지정해야 함
- 모든 표본에 대해 그 표본과 가장 가까운 평균벡터 사이의 거리의 합이 최소인 평균벡터를 구함
- 평균벡터의 초깃값에 따라 군집화 결과가 다를 수 있음

✓ 선형회귀

- 독립변수와 종속변수의 관계를 선형함수로 모델링한다.
- 이를 위해 평균제곱오차를 비용함수로 하여 경사하강법으로 선형함수의 계수를 조금씩 변화시키는 과정을 반복한다.

✓ 로지스틱 회귀

- 독립변수의 값에 대해 종속변수가 0과 1 사이의 값을 낼 수 있는 가설을 구한다.
- 로지스틱 회귀의 가설: 선형함수를 로짓으로 사용한다.
- 비용함수: 교차 엔트로피
- 학습 결과: 선형 결정경계를 형성한다.

✔ 다항 로지스틱 회귀

- 식별하려는 클래스가 3개 이상인 경우 사용할 수 있다.
- N개의 클래스에 대해 학습된 선형 판별함수에 소프트맥스 함수를 적용하여 각 클래스별로 0과 1 사이의 값을 구하며, 그중 가장 큰 값을 갖는 클래스로 분류할 수 있도록 가설을 학습한다.
- 교차 엔트로피를 비용함수로 사용하며, 경사하강법으로 비용함수가 최소화되도록 가설을 학습한다.

✓ k -평균 군집화

- 학습표본 집합을 대표하는 k 개의 평균벡터를 학습한다.
- 임의의 위치로 초기화한 k 개의 평균벡터로 시작하여 학습표본 집합을 이용하여 반복적으로 업데이트함으로써 평균벡터가 수렴하도록 학습한다.

12강

다음시간안내 ▶▶▶

인공 신경망(1)