PART. 선형회귀

- 기본적인 개념

- **회귀분석(Regression Analysis)**
회귀분석은 D차원 벡터 독립 변수 x(수치형 설명변수)와 이에 대응하는 스칼라 종속 변수 y(연속형 숫자)간의 관계를 정량적으로 찾아내는 작업. 회귀분석에서는 결정론적 모형과 확률적 모형이 있다.

Hypothesis : $h_\theta(x) = ax + b$

Parameters : a, b

Cost Function : $J(a,b) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$

Goal : $\underset{a,b}{\text{minimize}}\, J(a,b)$

PART. 선형회귀

결정론적 모형(Deterministic Model)

독립 변수 x에 대해 대응하는 종속 변수 y와 가장 비슷한 값 y-hat을 출력하는 함수 f(x)를 찾는 과정

$$\hat{y} = f(x) \approx y$$

만약 독립 변수 x와 이에 대응하는 종속 변수 y간의 관계가 다음과 같은 선형 함수 f(x)이면 선형 회귀분석(linear regression analysis)이라고 한다.

$$\hat{y} = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_D x_D = w_0 + w^T x$$

위 식에서 w를 함수 f(x)의 계수(coefficient)이자 이 선형 회귀모형의 모수(parameter)라고 한다.

PART. 선형회귀

- 선형 회귀의 목적은 일부 비용 함수(오류)를 최소화하는 회귀 함수 매개 변수(예: 평균 제곱 오차)를 계산하여 가장 좋은 선(또는 하이퍼/ 평면)을 주어진 점 세트에 맞추는 것이다.

- 선형 회귀 방정식

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

- 선형 회귀 방정식 (벡터화)

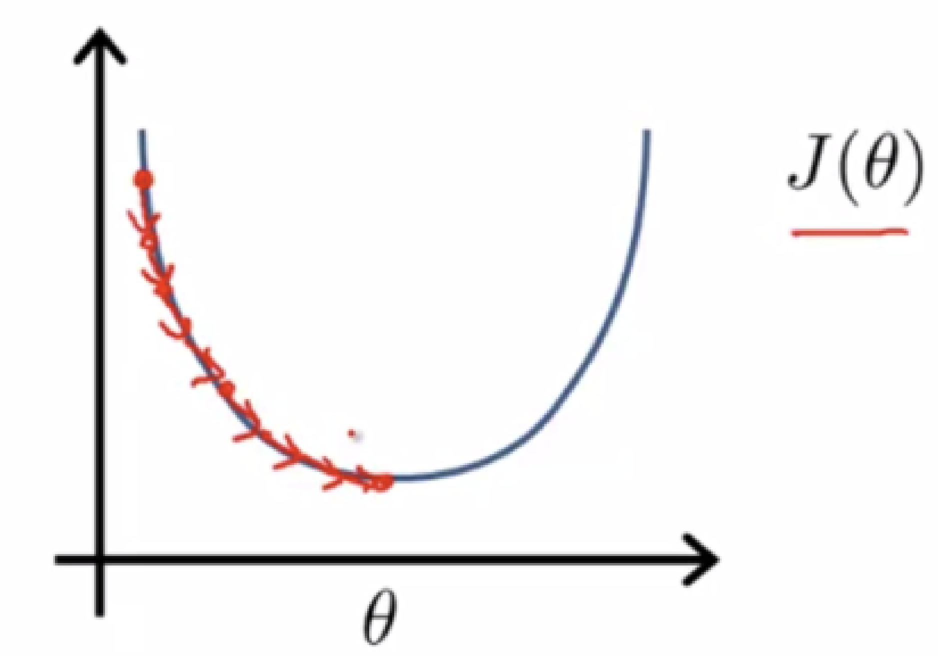
$$\hat{y} = \theta^T \cdot x$$

머신러닝에서 종종 벡터를 하나의 열을 가진 2D 배열인 열 벡터로 나타낸다.
따라서 이를 전치하여 행 벡터로 만들어 x와의 행렬 곱셈을 만들다.
예측 결과는 같지만 스칼라 값이 아니라 하나의 원소를 가진 행렬을 만든다.

PART. 선형회귀

정규방정식

정규방정식 이란 특정 선형 문제에서 파라미터값인 θ를 더 쉽게 풀 수 있게 만들어주는 방법이다.



지금까지의 경사하강법에서는, Global Minimum을 찾기 위해 많은 스텝들을 밟아야 했다.
하지만 정규방정식은 이것을 분석적으로 풀 수 있게 해준다.
경사하강법의 많은 반복을 하기보다, 분석적으로 θ값의 해를 구하면 한번에 구할 수 있다는 말이다.

PART. 선형회귀

예제)

변수가 많지 않은 아래 2차 방정식 수식을 보자.

$$J(\theta) = a\theta^2 + b\theta + c$$

위의 수식을 minimize하려면 미분 값을 구한 다음, =0을 하고 θ로 풀면 된다.

하지만 많은 문제들은 변수 θ가 2개에서 끝나지 않는다.

$$J(\theta_0, \theta_1, \ldots, \theta_m) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

이런 문제들은 어떻게 minimize를 할 수 있을까?
위의 수식을 풀 수 있는 방법은, 각 θ값에 대해 편미분을 하는것이다.
하지만 우리의 목적은 미분 없이 이것을 구하는 것이다.

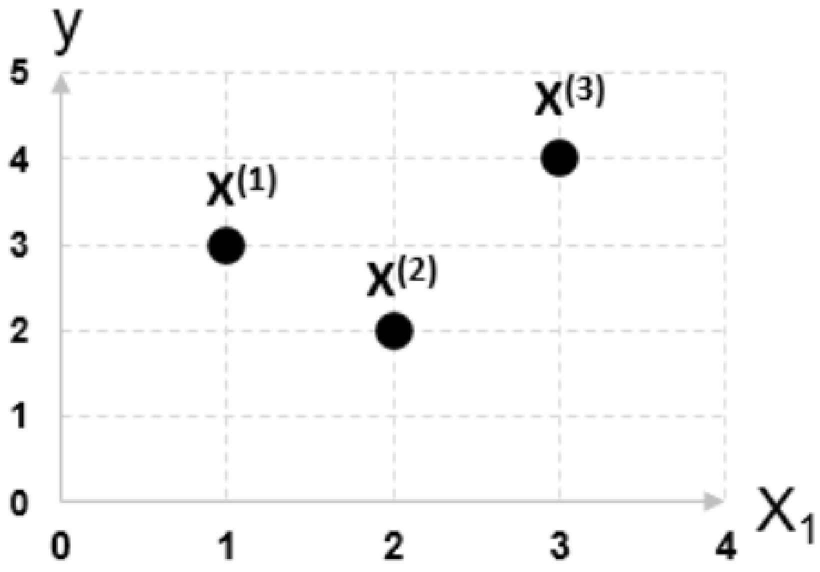
PART. 정규방정식

- 일반적으로 최상의 모델 매개 변수를 찾는 것은 비용 함수를 최소화하기 위해 일종의 최적화 알고리즘(경사하강)을 실행하여 수행된다. 그러나 정규 방정식이라고 하는 대수 방정식을 풀면 이러한 매개 변수의 값을 얻을 수 있다.

$$\hat{\theta} = (X^T \cdot X)^{-1} \cdot X^T \cdot y$$

- 구현

$$\hat{y} = \theta_0 + \theta_1 x_1$$



핸즈온 머신러닝 4장

PART. 정규방정식

$$\hat{y} = \theta_0 + \theta_1 x_1 \rightarrow$$

$$X = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix} \qquad y = \begin{bmatrix} 3 \\ 2 \\ 4 \end{bmatrix}$$

Default bias, X_0 $\qquad$ X_1

1단계: 행을 열로 바꾸기

- Xo은 평향 세타에 가상의 특성 Xo=1이 곱해졌다고 생각한다.

$$X^T = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \end{bmatrix}$$

2단계: 전치 행렬의 곱셈

$$\begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix}$$

$$X^T \cdot X = \begin{bmatrix} 3 & 6 \\ 6 & 14 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \end{bmatrix}$$

PART. 정규방정식

3단계: 결과 행렬 반전

$$A^{-1} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{det(A)} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

따라서 우리는 다음을 얻습니다.

$$(X^T \cdot X)^{-1} = \begin{bmatrix} 2/3 & -1 \\ -1 & 1/2 \end{bmatrix}$$

더 큰 행렬 (3X3보다 큼)의 경우 거꾸로하는 작업이 훨씬 번거롭고 일반적으로 가우시안 제거와 같은 알고리즘 방식이 사용

핸즈온 머신러닝 4장

PART. 정규방정식

4단계: 역행렬의 곱셈

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \end{bmatrix}$$

$$(X^T \cdot X)^{-1} \cdot X^T = \begin{bmatrix} {}^{4}/_{3} & {}^{1}/_{3} & -{}^{2}/_{3} \\ -{}^{1}/_{3} & 0 & {}^{1}/_{2} \end{bmatrix} \begin{bmatrix} {}^{2}/_{3} & -1 \\ -1 & {}^{1}/_{2} \end{bmatrix}$$

5단계:최적의 매개 변수 값을 얻기 위한 최종곱셈

선형회귀방정식 -> $\hat{y} = 2 + 0.5x_1$

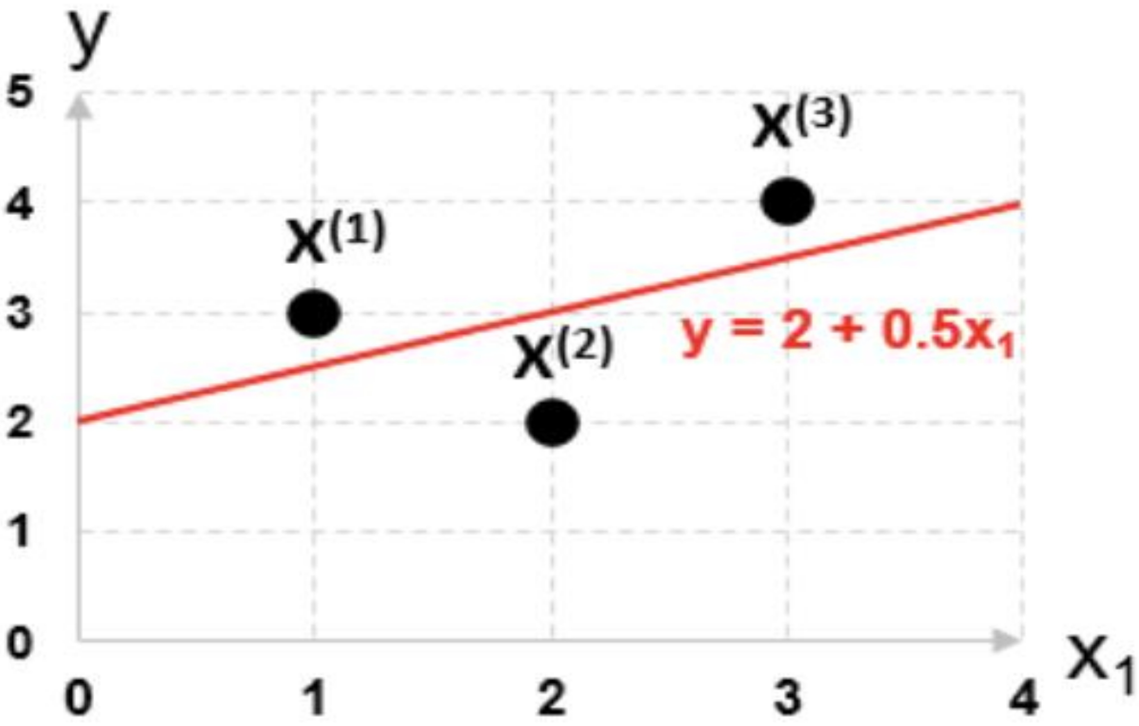
$$\begin{bmatrix} 3 \\ 2 \\ 4 \end{bmatrix}$$

$$\hat{\theta} = (X^T \cdot X)^{-1} \cdot X^T \cdot y = \begin{bmatrix} 2 \\ \frac{1}{2} \end{bmatrix} \begin{bmatrix} {}^{4}/_{3} & {}^{1}/_{3} & -{}^{2}/_{3} \\ -{}^{1}/_{3} & 0 & {}^{1}/_{2} \end{bmatrix}$$

PART. 정규방정식

- 선형 회귀 방정식

$$\hat{y} = 2 + 0.5x_1$$



핸즈온 머신러닝 4장

PART. 정규방정식

경사하강법 vs 정규방정식

경사하강법	정규방정식
• 학습률 α값을 결정해야 한다. • 많은 반복을 요구한다. • 많은 양의 특성(n)에도 잘 동작한다.	• 학습률 α값을 정해줄 필요가 없다. • 반복이 필요없다. • 수식 (X^T X)^-1을 풀어야 한다 • 특성의 갯수(n)가 많아지면 너무 느려진다. • (행렬전치, 행렬곱, 역행렬을 구하는데 O(n^3)의 계산시간이 걸린다.