양재 AI School 인공지능 캠프

Lecture 5

Supervised Learning: 회귀 (Regression)

박상기 선생님

Lecture 5

Supervised Learning:
회귀(Regression)

수업 목표

1 Supervised Learning을 이해한다

Supervised learning이 Unsupervised learning과 어떻게
다른지 이해하고 해당하는 알고리즘에는 무엇이 있는지 알아봅시다.

2 Regression의 다양한 방식을 알아본다

변수의 개수, cost 함수의 최소화 등, 여러가지 회귀 방식들을
알아봅시다.

Machine Learning

- Arthur Samuel (1959). Machine Learning: Field of study that gives computers the ability to learn without being explicitly programmed.
- Tom Mitchell (1998) Well-posed Learning Problem: A computer program is said to *learn* from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.

Machine Learning

- 어떤 작업을 컴퓨터가 해야 할 때, 사람이 규칙을 정해주는 대신 컴퓨터 스스로가 데이터를 통해 패턴을 학습하는 방식

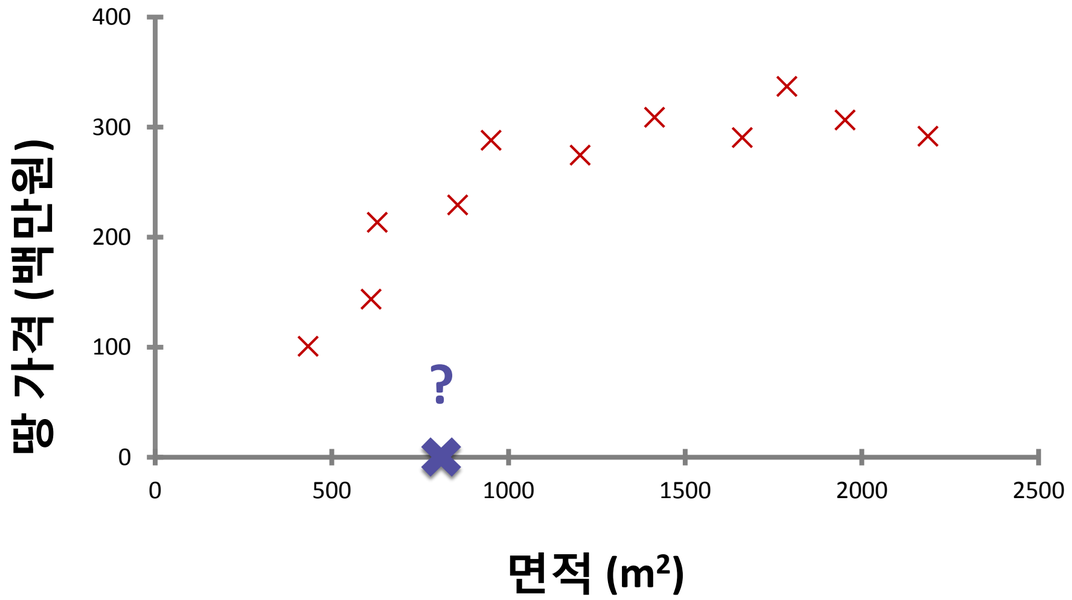
- 빅데이터 분석, 손 글씨 인식, 자연어 처리, 컴퓨터 비전 …

Machine Learning

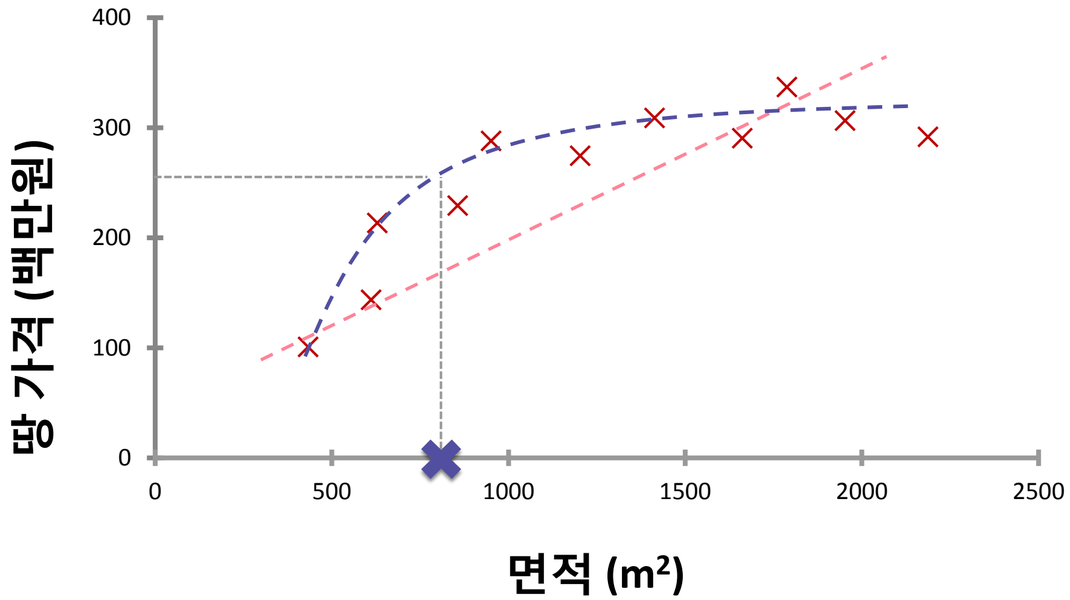
크게 Supervised Learning 또는
Unsupervised Learning,
Reinforcement Learning 으로 구분

Supervised Learning

특정 **input** 값에 대해 **“정답 (label)”** 이 있는 데이터가 주어진다.

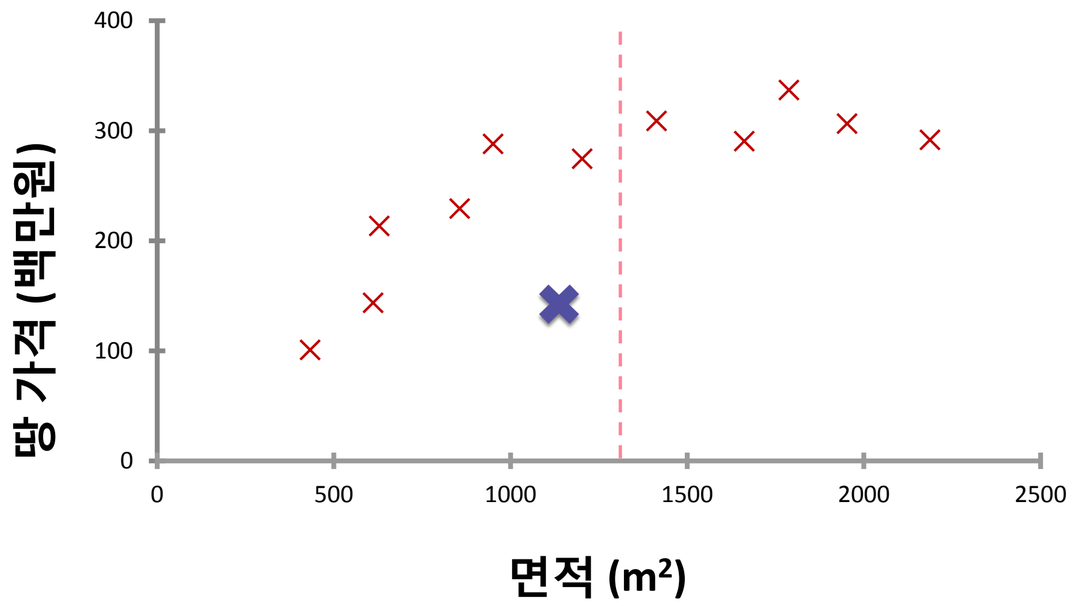


Supervised Learning



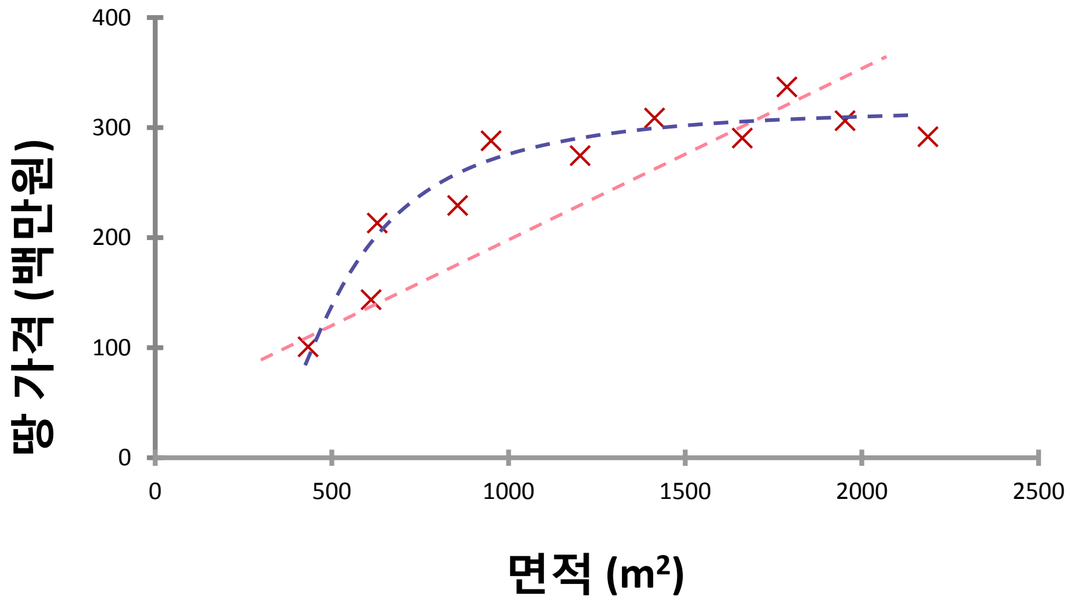
- **Regression** : 어떤 input 값을 특정 output 값에 대응시켜주는 함수를 찾는 과정

Supervised Learning



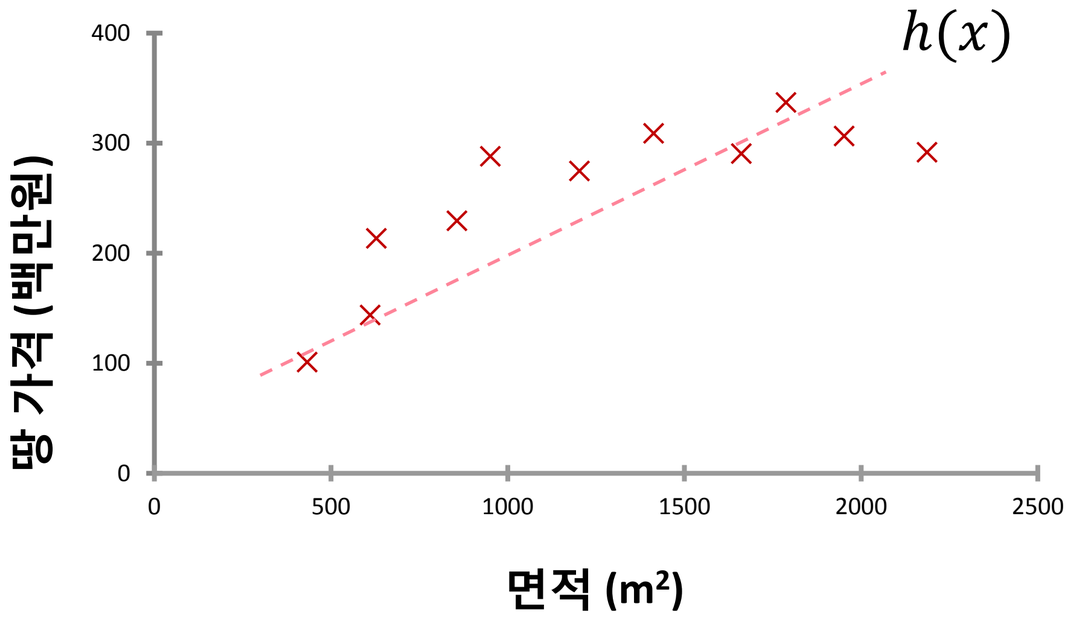
- **Classification** : 주어진 input이 어느 카테고리에 있는지 판별

Regression



- 분포된 데이터를 **방정식(Hypothesis)**을 통해 결과의 **값**을 예측

Regression



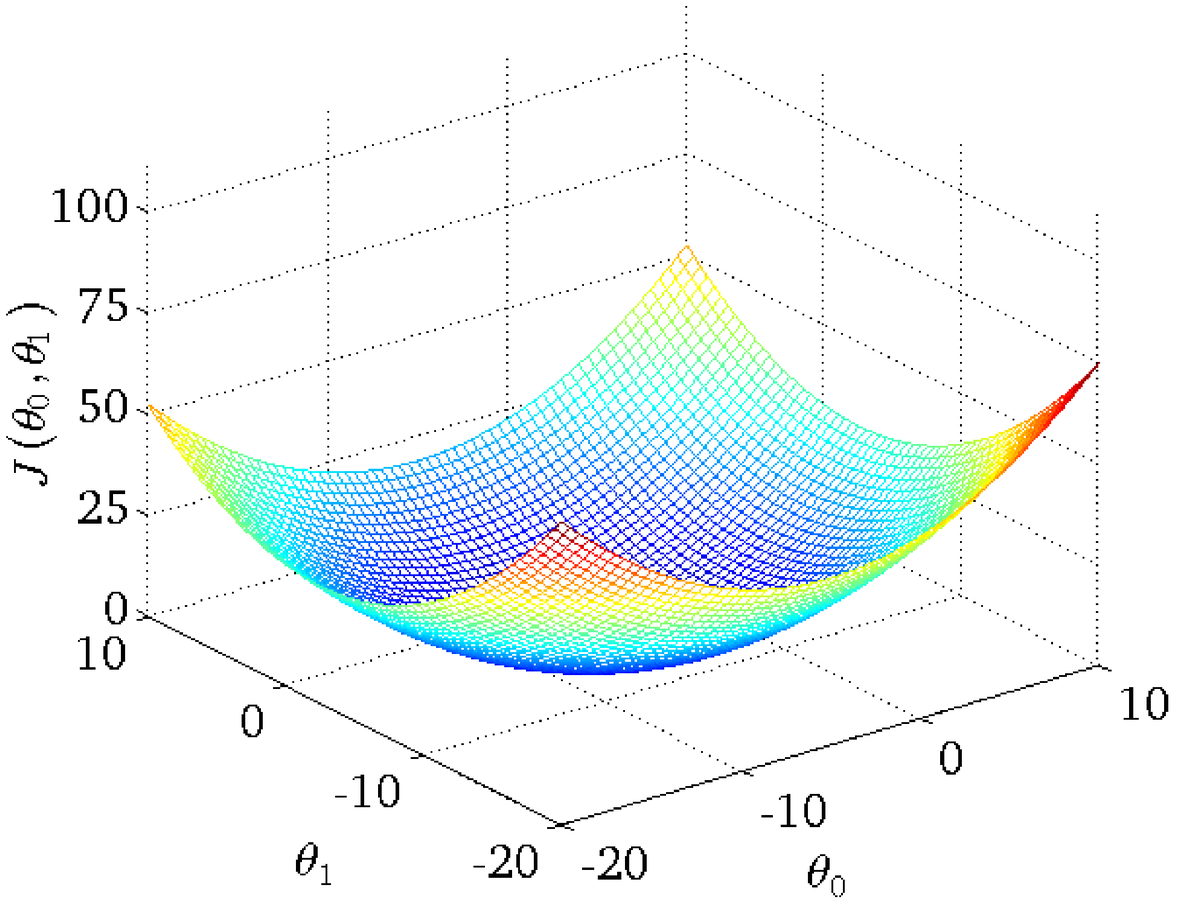
면적 (x)	가격 (y_i)
480	100
580	125
620	220
890	230
990	290
…	…

- $h(x)$를 어떻게 나타내야 할까?
- **Linear Regression**: $h(x) = \theta_0 + \theta_1 x$
- θ_i의 값은 어떻게 결정해야 할까?

Cost Function

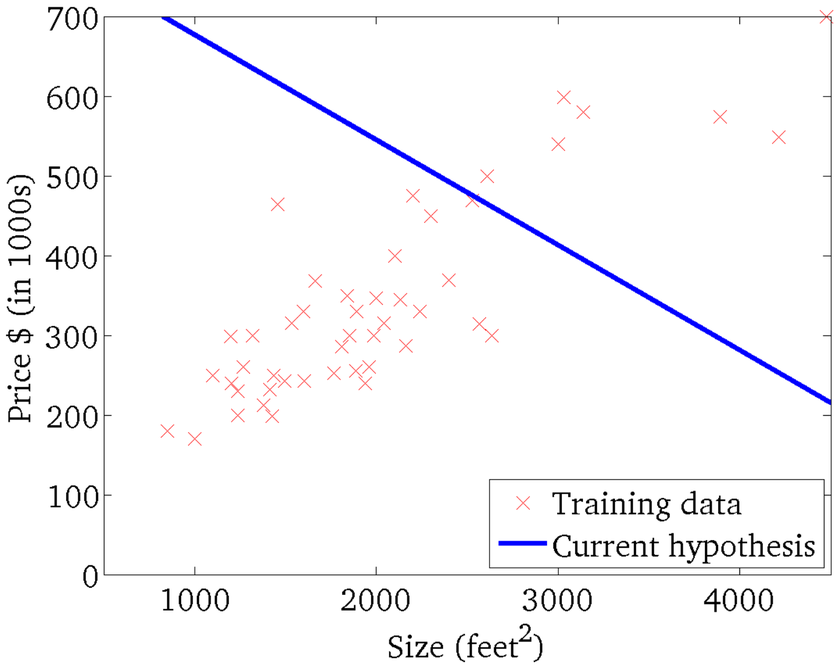
- $error = h(x_i) - y_i$ 의 값을 최소화 하는 θ_i
- 에러값은 양수, 음수 일수도 있으므로
  Mean-squre-error를 구한다
- $J(\theta_0,\ \theta_1) = \frac{1}{2m}(\sum_{i=1}^{m}(h(x_i) - y_i)^2)$: **Square error function**
- m은 label의 개수

- 목표: $\underset{\theta_0,\theta_1}{\text{minimize}}\, J(\theta_0,\ \theta_1)$

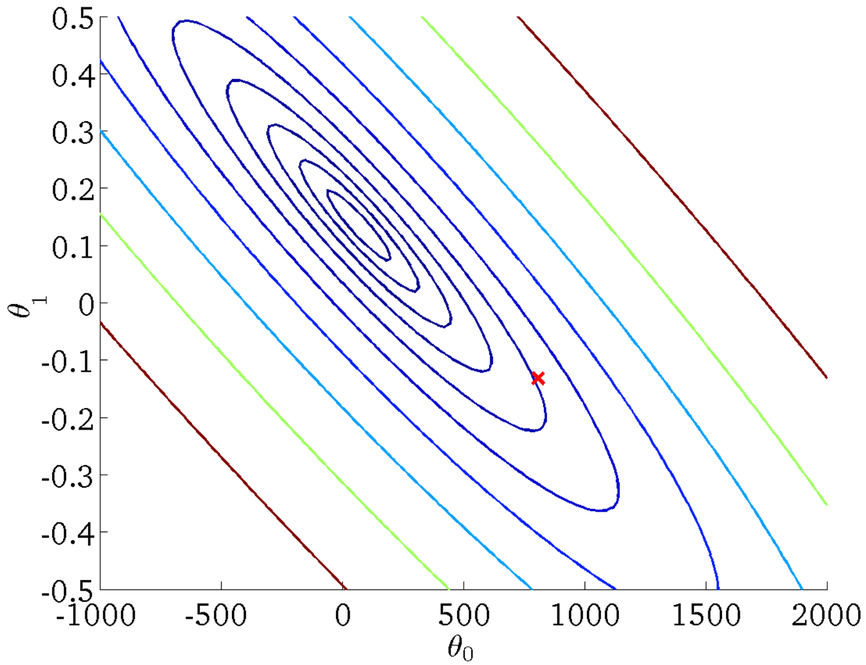


볼록함수 (convex): 최소값을 쉽게 찾을 수 있음

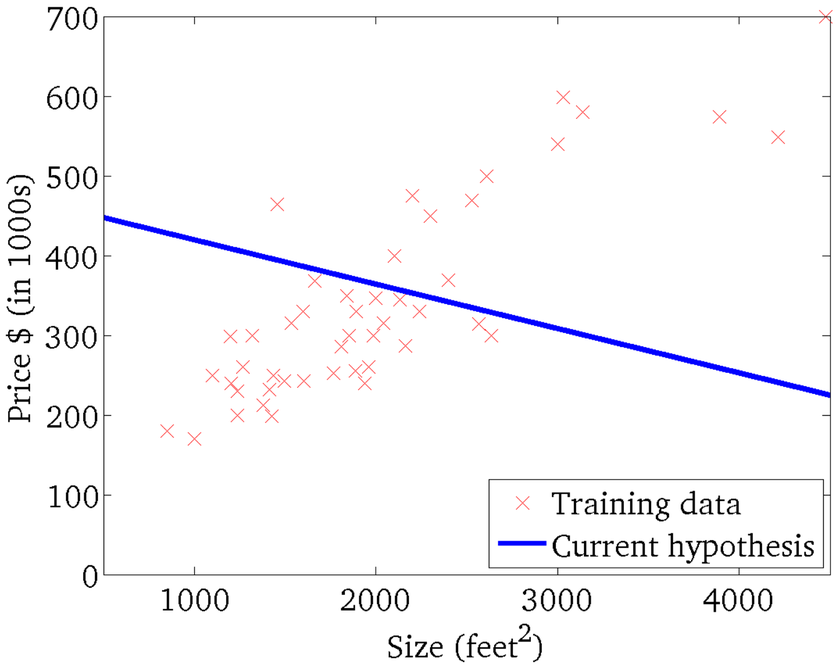
$h(x)$



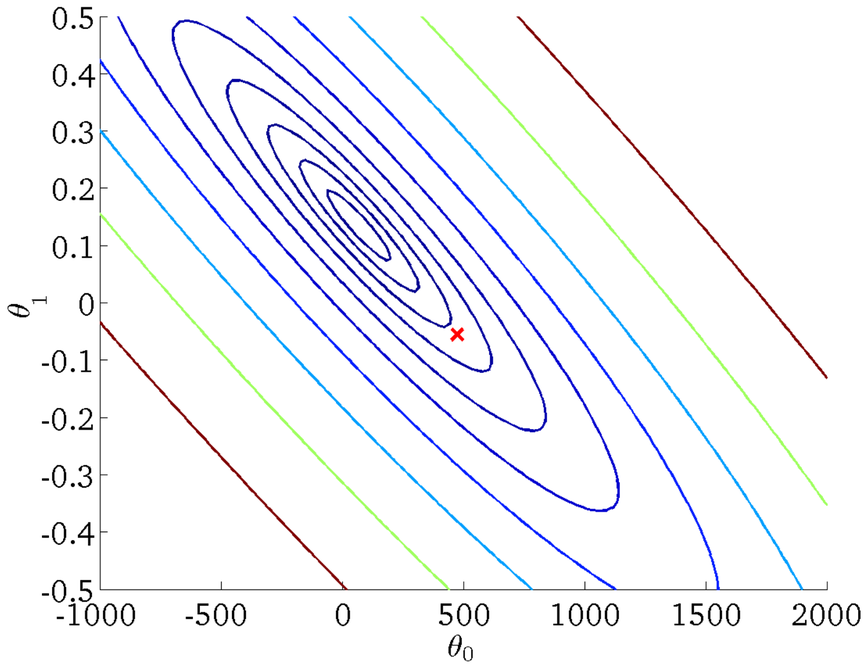
$J(\theta_0, \theta_1)$



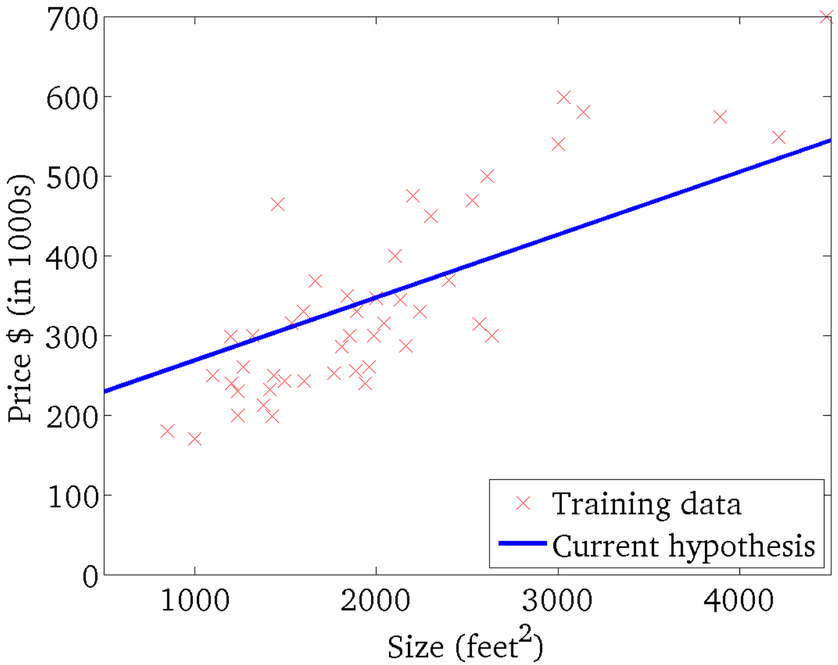
$h(x)$



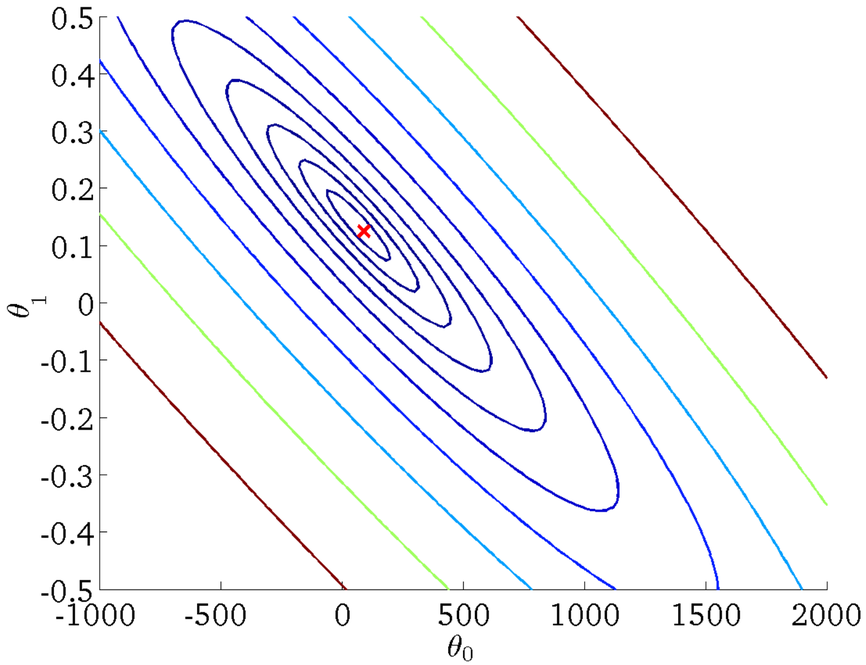
$J(\theta_0, \theta_1)$



$h(x)$



$J(\theta_0, \theta_1)$



Gradient Descent

- 어떤 식의 최소 값을 찾아보자!
- E.g.) $y = 5x + 4x^2$의 최소값은

$\frac{dy}{dx} = 5 + 8x = 0$ 이 되는 $x = -0.625, y = -1.6$

- Gradient descent도 동일하게 미분 값을 이용한다
- 다만 각 식의 계수가 미지수 $y = \gamma + \alpha x + \beta x^2$

Gradient Descent

- $h(x)$의 최적 parameter를 찾는 방법
- Cost function을 최소화 하는 방식

$$\text{repeat until convergence } \{$$
$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \quad (\text{for } j = 0 \text{ and } j = 1)$$
$$\}$$

- α : learning rate, $\alpha > 0$
- $:=$: 할당 (assignment) 연산자
- θ_0, θ_1은 한번에 업데이트 해야 한다

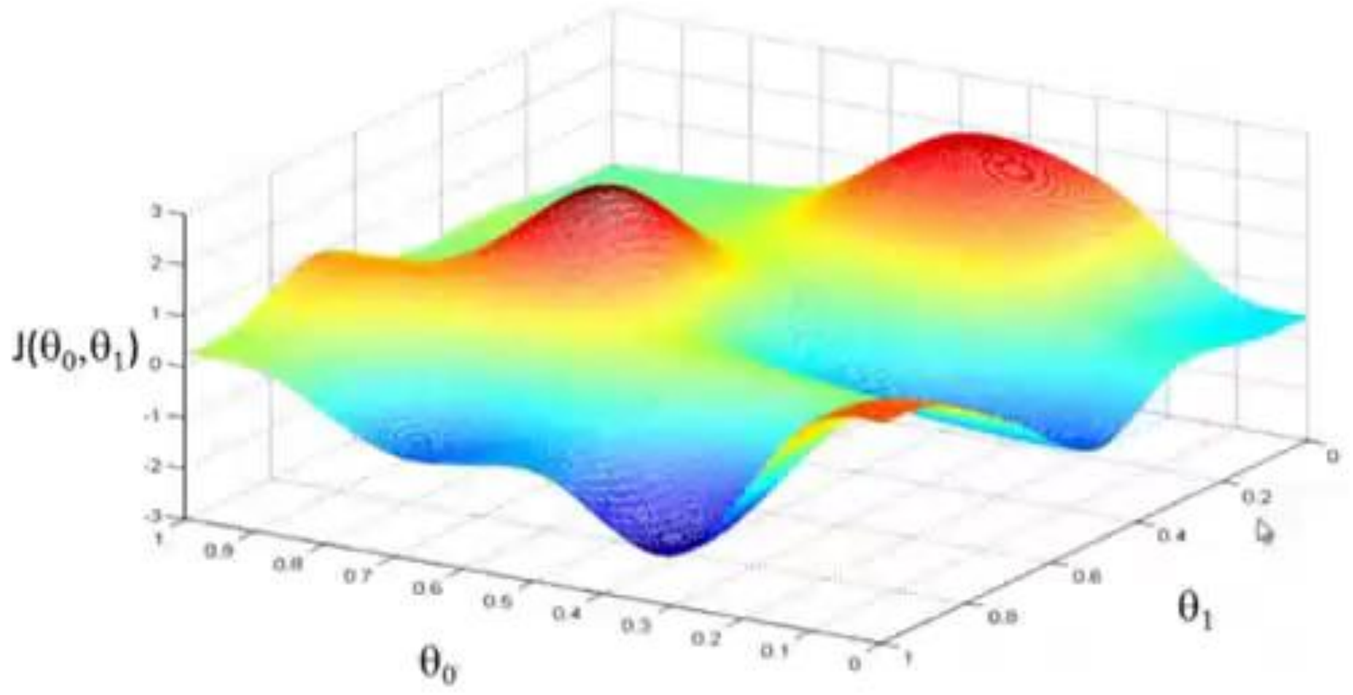
Correct

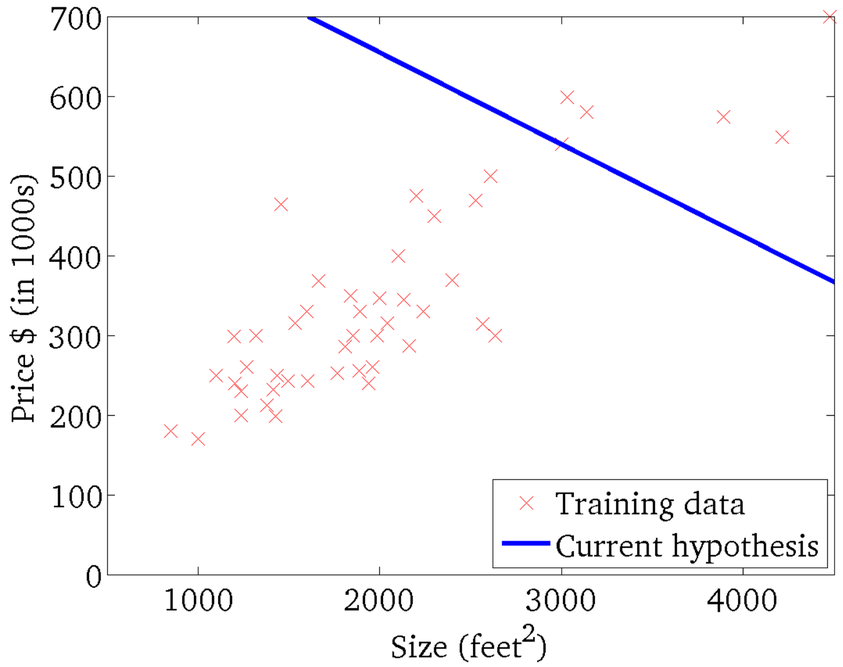
$$\text{temp0} := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$
$$\text{temp1} := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$
$$\theta_0 := \text{temp0}$$
$$\theta_1 := \text{temp1}$$

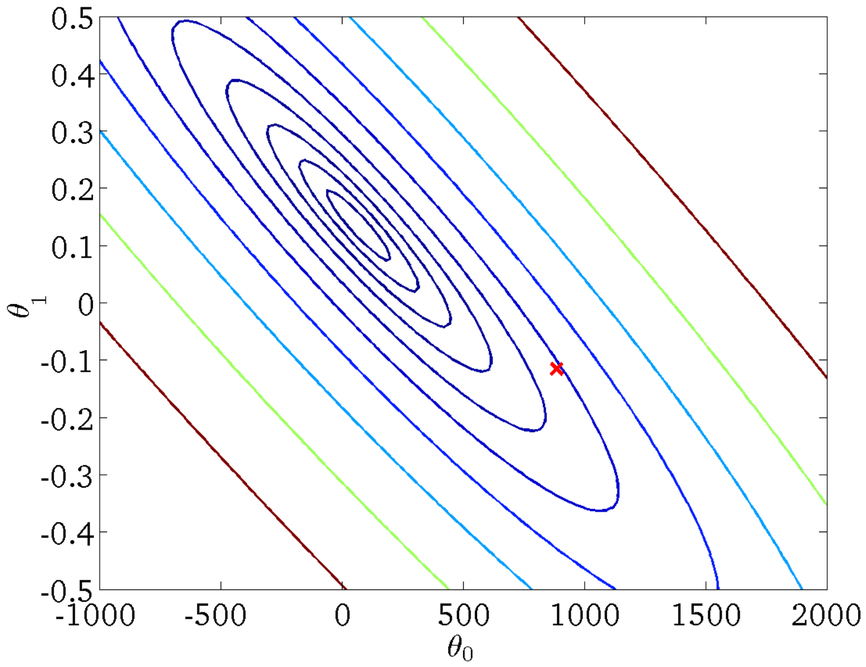
Incorrect

$$\text{temp0} := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$
$$\theta_0 := \text{temp0}$$
$$\text{temp1} := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$
$$\theta_1 := \text{temp1}$$

- learning rate (α)값이 클수록 한번에 더 많이 이동
- α가 너무 작으면 수렴하는데 오래 걸리고, 너무 크면 최소값을 지나쳐 버릴 수도 있다
- $\frac{\partial J}{\partial \theta_i}$항은 다음에 이동할 방향과 크기를 결정


$J(\theta_0,\theta_1)$
3
2
1
0
-1
-2
-3
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
θ_0
0
0.2
0.4
0.6
0.8
1
θ_1


Price $ (in 1000s)
Size (feet^2)
Training data
Current hypothesis
0
100
200
300
400
500
600
700
1000
2000
3000
4000


θ_1
θ_0
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
-1000
-500
0
500
1000
1500
2000

$$\frac{\partial J}{\partial \theta_j} = \frac{\partial}{\partial \theta_j}\left[\frac{1}{2m}\left(\sum_{i=1}^{m}(\theta_0 + \theta_1 x_i - y_i)^2\right)\right]$$

Repeat until convergence, do{

$$\theta_0 := \theta_0 - \alpha\frac{1}{m}\left(\sum_{i=1}^{m}(\theta_0 + \theta_1 x_i - y_i)\right)$$

$$\theta_1 := \theta_1 - \alpha\frac{1}{m}\left(\sum_{i=1}^{m}(\theta_0 + \theta_1 x_i - y_i)x_i\right)$$

}

Multivariate Regression

면적 x_1	방의 개수 x_2	층 수 x_3	주변 역 개수 x_4	건축년도 x_5	가격 (y_i)
480	5	1	1	10	100
580	3	1	2	5	125
620	1	1	3	1	220
890	4	2	2	15	230
990	6	2	3	4	290
…	…	…	…	…	…

- 영향을 미치는 변수가 여러 개인 경우?
- $h(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$
- n: 데이터의 개수 ; $x_j^{(i)}$: j번째 변수의 i번째 input

- $h(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$

- $x = \begin{bmatrix} x_0 \\ x_1 \\ \cdots \\ x_n \end{bmatrix}$, $\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \cdots \\ \theta_n \end{bmatrix}$, $h(x) = \theta^{\mathrm{T}} x$

Repeat until convergence, do{

$$\theta_j := \theta_j - \alpha \frac{1}{m} \left(\sum_{i=1}^{m} \left(h_\theta \left(x^{(i)} \right) - y_i \right) x_j^{(i)} \right), for\ (j = 0,1,2\ \ldots n)$$

}

Feature Scaling

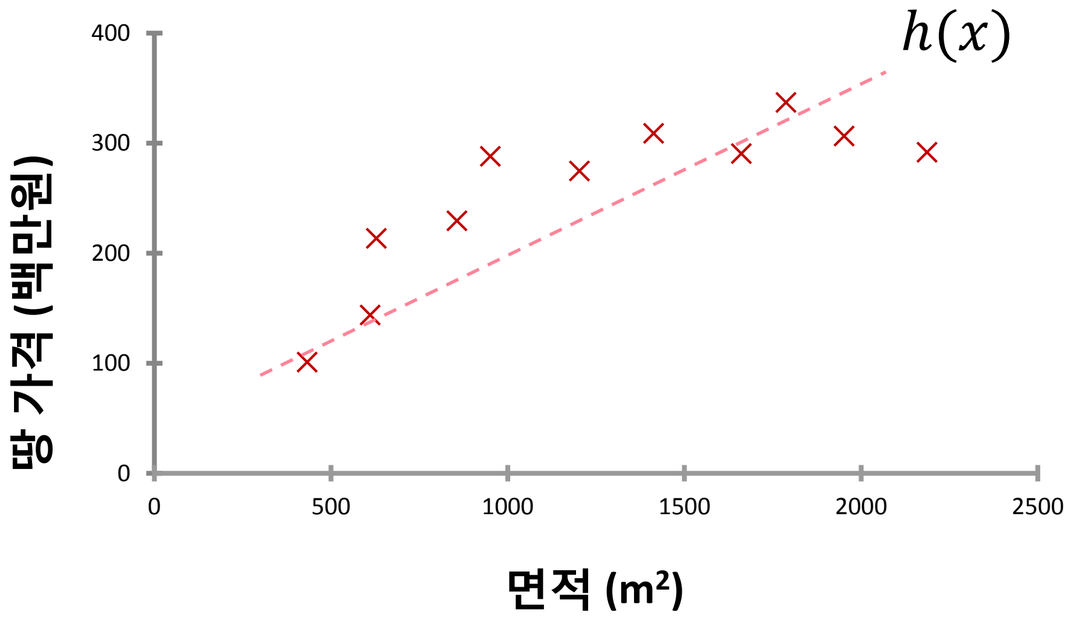
- 모든 feature가 비슷한 범위에 있도록 만들어 준다
- E.g. x_1 $0\sim2000 \Rightarrow x_1 := \frac{x_1}{2000-0}, x_2 : 0\sim5 \Rightarrow x_2 := \frac{x_2}{5}$

- Feature들이 비슷한 범위에 있으면 수렴이 빠르다

Feature Scaling

- **Mean normalization**: Feature Scaling의 한 방법
- x_i대신 $x_i - \mu_i$로 바꿔 평균을 0 근처 값으로 만든다
- E.g. $x_1 := \frac{x_1 - 1000}{2000 - 0}$, $x_2 := \frac{x_2 - 2}{5}$
- $-0.5 \leq x_1 \leq 0.5, -0.5 \leq x_2 \leq 0.5$

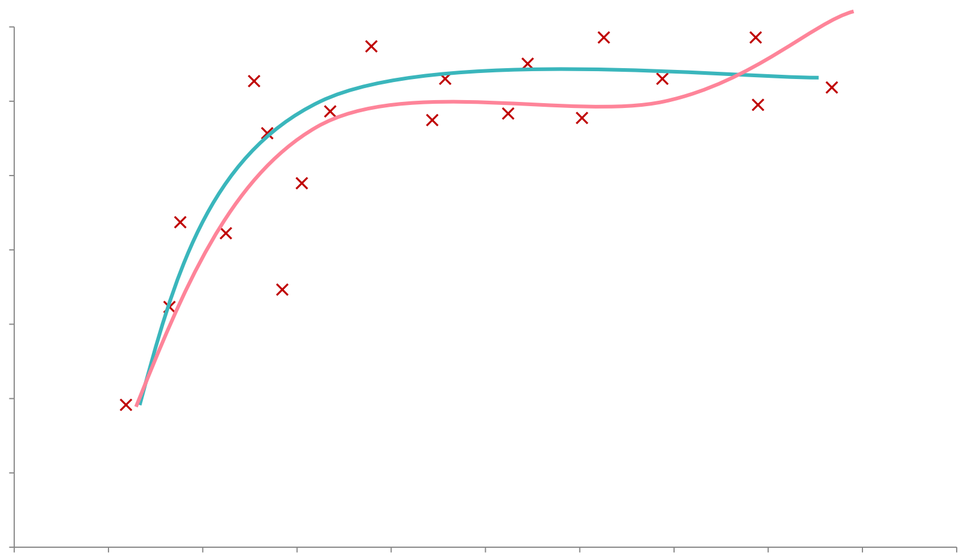
Polynomial Regression



면적 (x)	가격 (y_i)
480	100
580	125
620	220
890	230
990	290
…	…

- 만약 땅 가격이 면적의 제곱에 더 관련이 있다면?
- 일차 함수인 $h(x)$는 모델 예측에 부적합

Polynomial Regression



$$\theta_0 + \theta_1 x + \theta_2 \sqrt{x}$$
$$\theta_0 + \theta_1 x + \theta_2 x^2$$
$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$$

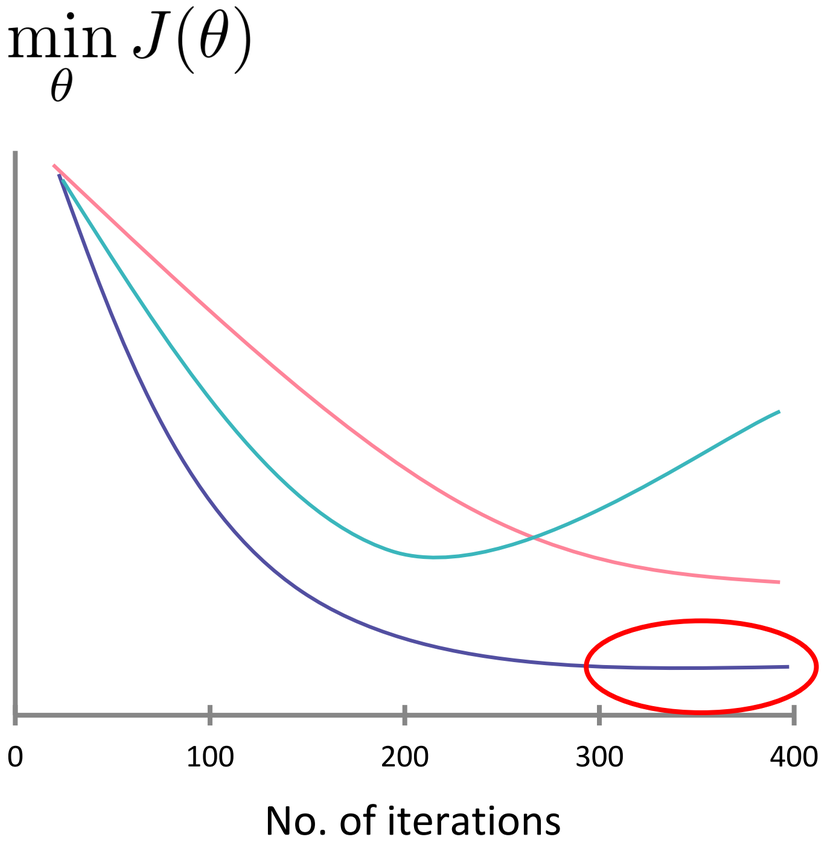
- $h(x)$함수가 제곱근, 2차, 3차 함수 등의 곡선일수도 있다
- x^n 꼴을 사용한다면 feature scaling이 더욱 중요해진다
- $x: 1\sim100, \quad x^2: 1\sim10{,}000, \quad x^3: 1\sim1{,}000{,}000$

Learning Rate

- 목표: $\underset{\theta_0,\theta_1}{\text{minimize}} J(\theta_0,\ \theta_1)$ $\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$

- Gradient descent가 어떻게 해야 잘 작동하는가?
- Learning rate α를 어떻게 선택해야 할까?

Learning Rate



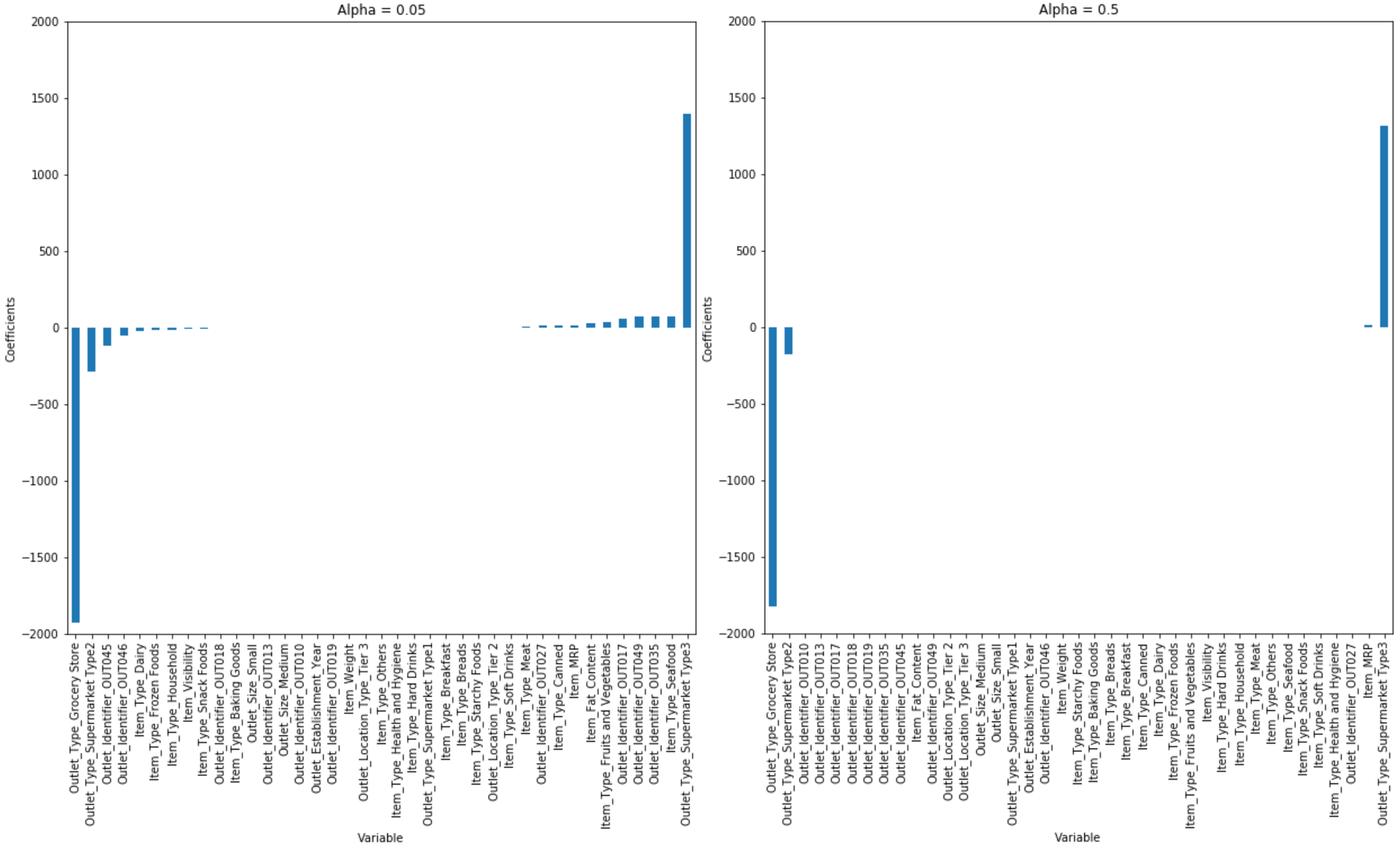
- α가 너무 작으면 수렴이 느림
- α가 너무 크면 수렴 안될 수도 있음 (overshootting)

- $J(\theta)$의 변화 값이 일정 수준이상 작아지면 수렴으로 판정

- α를 작은 값부터 천천히 늘려나가 보자: 0.001 -> 0.01 -> 0.03 -> 0.1 …

Lasso Regression

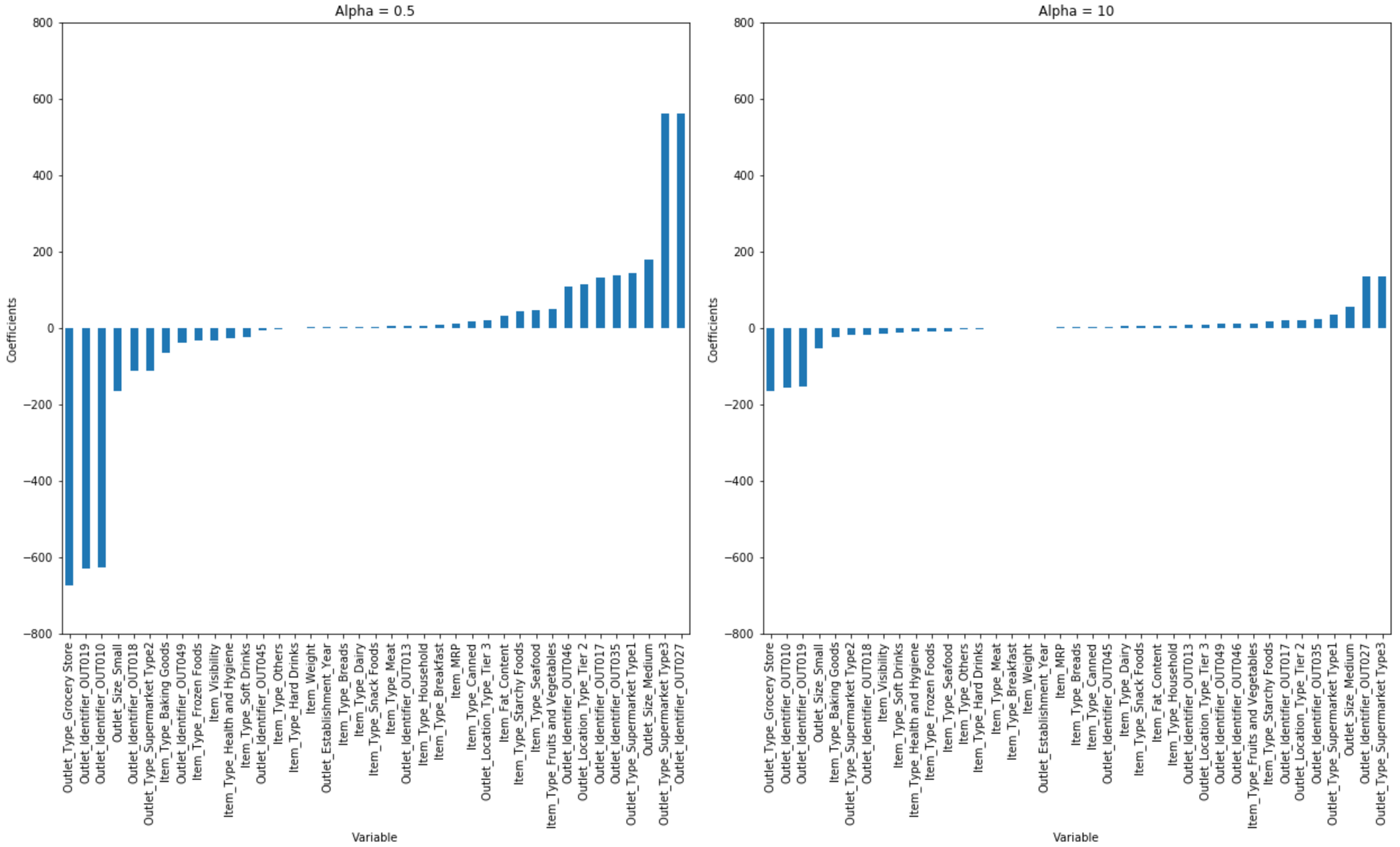
- Cost function에 L1 **regularization** 항을 추가
- **Regularization?:** overfitting 을 막기 위해 사용하는 기법, 중요하지 않은 계수 θ를 0에 가깝게 만들어 **모델의 복잡성**을 줄인다.
- $J(\theta_0,\ \theta_1) = \frac{1}{2m}(\sum_{i=1}^{m}(h(x_i) - y_i)^2) + \frac{\lambda}{2}\sum_{j=1}^{n}|\theta_j|$


Alpha = 0.05
Alpha = 0.5
Coefficients
Variable
2000
1500
1000
500
0
−500
−1000
−1500
−2000
Outlet_Type_Grocery Store
Outlet_Type_Supermarket Type2
Outlet_Identifier_OUT045
Outlet_Identifier_OUT046
Item_Type_Dairy
Item_Type_Frozen Foods
Item_Type_Household
Item_Visibility
Item_Type_Snack Foods
Outlet_Identifier_OUT018
Item_Type_Baking Goods
Outlet_Size_Small
Outlet_Identifier_OUT013
Outlet_Size_Medium
Outlet_Identifier_OUT010
Outlet_Establishment_Year
Outlet_Identifier_OUT019
Item_Weight
Outlet_Location_Type_Tier 3
Item_Type_Others
Item_Type_Health and Hygiene
Item_Type_Hard Drinks
Outlet_Type_Supermarket Type1
Item_Type_Breakfast
Item_Type_Breads
Item_Type_Starchy Foods
Outlet_Location_Type_Tier 2
Item_Type_Soft Drinks
Item_Type_Meat
Outlet_Identifier_OUT027
Item_Type_Canned
Item_MRP
Item_Fat_Content
Item_Type_Fruits and Vegetables
Outlet_Identifier_OUT017
Outlet_Identifier_OUT049
Outlet_Identifier_OUT035
Item_Type_Seafood
Outlet_Type_Supermarket Type3
Outlet_Type_Grocery Store
Outlet_Type_Supermarket Type2
Outlet_Identifier_OUT010
Outlet_Identifier_OUT013
Outlet_Identifier_OUT017
Outlet_Identifier_OUT018
Outlet_Identifier_OUT019
Outlet_Identifier_OUT035
Outlet_Identifier_OUT045
Item_Fat_Content
Outlet_Identifier_OUT049
Outlet_Location_Type_Tier 2
Outlet_Location_Type_Tier 3
Outlet_Size_Medium
Outlet_Size_Small
Outlet_Type_Supermarket Type1
Outlet_Establishment_Year
Outlet_Identifier_OUT046
Item_Weight
Item_Type_Starchy Foods
Item_Type_Baking Goods
Item_Type_Breads
Item_Type_Breakfast
Item_Type_Canned
Item_Type_Dairy
Item_Type_Frozen Foods
Item_Type_Fruits and Vegetables
Item_Visibility
Item_Type_Hard Drinks
Item_Type_Household
Item_Type_Meat
Item_Type_Others
Item_Type_Seafood
Item_Type_Snack Foods
Item_Type_Soft Drinks
Item_Type_Health and Hygiene
Outlet_Identifier_OUT027
Item_MRP
Outlet_Type_Supermarket Type3

Ridge Regression

- Cost function에 L2 regularization 항을 추가
- $J(\theta_0,\ \theta_1) = \frac{1}{2m}(\sum_{i=1}^{m}(h(x_i) - y_i)^2) + \frac{\boldsymbol{\lambda}}{2}\sum_{j=1}^{n}\boldsymbol{\theta_j^2}$

- $\theta_j := \theta_j\left(\mathbf{1} - \boldsymbol{\alpha}\frac{\boldsymbol{\lambda}}{\boldsymbol{m}}\right) - \alpha\frac{1}{m}\left(\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y_i\right)x_j^{(i)}\right)$


Alpha = 0.5
Coefficients
Variable
Outlet_Type_Grocery Store
Outlet_Identifier_OUT019
Outlet_Identifier_OUT010
Outlet_Size_Small
Outlet_Identifier_OUT018
Outlet_Type_Supermarket Type2
Item_Type_Baking Goods
Outlet_Identifier_OUT049
Item_Type_Frozen Foods
Item_Visibility
Item_Type_Health and Hygiene
Item_Type_Soft Drinks
Outlet_Identifier_OUT045
Item_Type_Others
Item_Type_Hard Drinks
Item_Weight
Outlet_Establishment_Year
Item_Type_Breads
Item_Type_Dairy
Item_Type_Snack Foods
Item_Type_Meat
Outlet_Identifier_OUT013
Item_Type_Household
Item_Type_Breakfast
Item_MRP
Item_Type_Canned
Outlet_Location_Type_Tier 3
Item_Fat_Content
Item_Type_Starchy Foods
Item_Type_Seafood
Item_Type_Fruits and Vegetables
Outlet_Identifier_OUT046
Outlet_Location_Type_Tier 2
Outlet_Identifier_OUT017
Outlet_Identifier_OUT035
Outlet_Type_Supermarket Type1
Outlet_Size_Medium
Outlet_Type_Supermarket Type3
Outlet_Identifier_OUT027
Alpha = 10
Coefficients
Variable
Outlet_Type_Grocery Store
Outlet_Identifier_OUT010
Outlet_Identifier_OUT019
Outlet_Size_Small
Item_Type_Baking Goods
Outlet_Type_Supermarket Type2
Outlet_Identifier_OUT018
Item_Visibility
Item_Type_Soft Drinks
Item_Type_Health and Hygiene
Item_Type_Frozen Foods
Item_Type_Seafood
Item_Type_Others
Item_Type_Hard Drinks
Item_Type_Meat
Item_Type_Breakfast
Item_Weight
Outlet_Establishment_Year
Item_MRP
Item_Type_Breads
Item_Type_Canned
Outlet_Identifier_OUT045
Item_Type_Dairy
Item_Type_Snack Foods
Item_Fat_Content
Item_Type_Household
Outlet_Identifier_OUT013
Outlet_Location_Type_Tier 3
Outlet_Identifier_OUT049
Outlet_Identifier_OUT046
Item_Type_Fruits and Vegetables
Item_Type_Starchy Foods
Outlet_Identifier_OUT017
Outlet_Location_Type_Tier 2
Outlet_Identifier_OUT035
Outlet_Type_Supermarket Type1
Outlet_Size_Medium
Outlet_Identifier_OUT027
Outlet_Type_Supermarket Type3

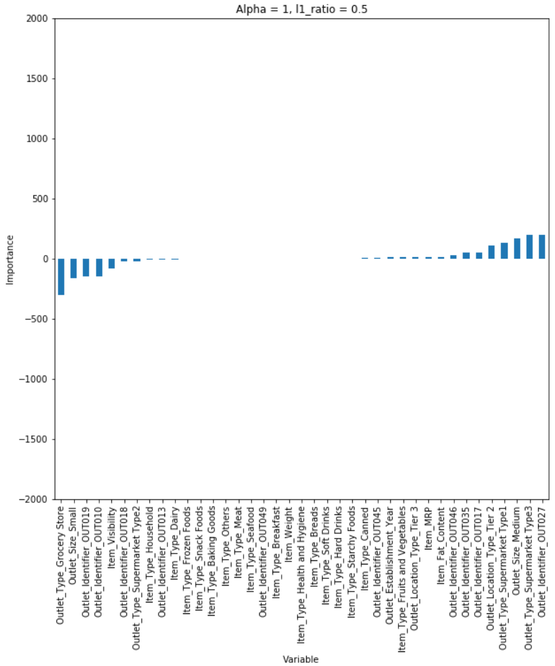
Elastic Net Regression

- Ridge 회귀는 계수값을 0에 가깝게 만들지만 0은 아니므로 모델이 여전히 복잡할 수 있음
- Lasso 회귀는 너무 많은 계수를 0으로 만들어 모델의 정확성이 떨어질 수 있음

- 그러면 두 개를 같이 써보자!

Elastic Net Regression

- $J(\theta_0,\ \theta_1) = \frac{1}{2m}(\sum_{i=1}^{m}(h(x_i) - y_i)^2) + \frac{\boldsymbol{\lambda_2}}{2}\sum_{j=1}^{n}\boldsymbol{\theta_j^2} + \frac{\boldsymbol{\lambda_1}}{2}\sum_{j=1}^{n}|\theta_j|$
- $\boldsymbol{\lambda_1}$과 $\boldsymbol{\lambda_2}$의 비율을 조정해 가며 모델을 만든다



상황에 맞게 계수를 정하는 것이 중요